

**UNIVERSIDADE FEDERAL DA PARAÍBA**  
**CENTRO DE CIÊNCIAS E TECNOLOGIA**  
**COORDENAÇÃO DE PÓS-GRADUAÇÃO EM INFORMÁTICA**

**DISSERTAÇÃO DE MESTRADO**

**SAVAD - SISTEMA DE AVALIAÇÃO DE DESEMPENHO DE  
MODELOS DE REDES DE FILAS**

**Francisco de Assis Coutinho Souto**

**Campina Grande - PB**

**Novembro/1993**

T15  
10/11/93  
21304

**DISSERTAÇÃO DE MESTRADO**

**TÍTULO: SAVAD - SISTEMA DE AVALIAÇÃO DE DESEMPENHO DE  
MODELOS DE REDES DE FILAS**

**CURSO: MESTRADO EM INFORMÁTICA DA UNIVERSIDADE FEDERAL DA  
PARAÍBA**

**ÁREA DE CONCENTRAÇÃO: CIÊNCIA DA COMPUTAÇÃO**

**MESTRANDO: Francisco de Assis Coutinho Souto**

**ORIENTADORA: Profa. Maria Izabel Cavalcanti Cabral**

**CO-ORIENTADOR: Prof. Hélio de Menezes Silva**



---

S726s Souto, Francisco de Assis Coutinho  
SAVAD : sistema de avaliacao de desempenho de modelos de  
redes de filas / Francisco de Assis Coutinho Souto. -  
Campina Grande, 1993.  
109 f. : il.

Dissertacao (Mestrado em Informatica) - Universidade  
Federal da Paraiba, Centro de Ciencias e Tecnologia.

1. Rede de Computadores 2. Redes de Filas 3. Dissertacao  
I. Cabral, Maria Izabel Cavalcanti, Dra. II. Silva, Helio  
de Menezes, Dr. III. Universidade Federal da Paraiba -  
Campina Grande (PB) IV. Título

CDU 004.7(043)

---

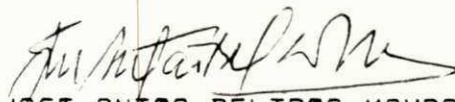
SAVAD: SISTEMA DE AVALIAÇÃO DE DESEMPENHO DE MODELOS DE REDES DE  
FILAS

FRANCISCO DE ASSIS COUTINHO SOUTO

DISSERTAÇÃO APROVADA EM 10.12.1993

  
MARIA IZABEL CAVALCANTI CABRAL, D.Sc  
Orientador

  
HELIO DE MENEZES SILVA, M.Sc  
Componente da Banca

  
JOSE ANTAO BELTRAO MOURA, Ph.D  
Componente da Banca

  
JOSE AUGUSTO SURUAGY MONTEIRO, Ph.D  
Componente da Banca

Campina Grande, 10 de dezembro de 1993

## **AGRADECIMENTOS**

Agradeço à minha esposa, minhas filhas, familiares e amigos pelo apoio e estímulo prestados.

Agradeço aos meus orientadores, Profa. Maria Izabel Cavalcanti Cabral e o Prof. Hélio de Menezes Silva, pelo apoio a mim dispensados.

## SUMÁRIO

<b>RESUMO</b> .....	1
<b>ABSTRACT</b> .....	2
<b>CAPÍTULO 1 INTRODUÇÃO</b> .....	3
1.1 Objetivos .....	5
1.2 Contribuição Científica .....	6
1.3 Organização .....	6
<b>CAPÍTULO 2 SAVAD</b> .....	8
2.1 Elementos de Modelagem .....	8
2.2 Exemplos de Modelos .....	14
2.2.1 Modelagem de uma Clínica Médica .....	14
2.2.2 Conexão de Transporte com o Protocolo de Janela Deslizante .....	17
2.2.3 Modelo da Integração de Voz e Dados em Redes Locais com Passagem de Ficha .....	20
2.3 Estrutura do SAVAD .....	25
2.3.1 Interface .....	26
2.3.2 Núcleo do SAVAD .....	29
2.3.3 Solução .....	31
2.4 Medidas de Desempenho .....	32

<b>CAPÍTULO 3 PROJETO ORIENTADO A OBJETOS DO SAVAD</b> .....	<b>33</b>
3.1 Abordagem Orientada a Objetos .....	33
3.2 Documentação do Projeto .....	35
3.2.1 Diagrama Geral de Herança de Classes .....	38
3.2.2 Diagrama Geral Cliente-Servidor .....	39
3.2.3 Protocolos de Classes .....	42
3.2.4 Diagramas de Booch das Classes .....	44
3.2.5 Diagrama Herança/Cliente-Servidor .....	46
3.3 Projeto do SAVAD .....	48
3.3.1 Identificação das Abstrações de Dados .....	48
3.3.2 Identificação dos Atributos das Abstrações de Dados .....	51
3.3.3 Identificação das Operações das Abstrações de Dados .....	51
3.3.4 Definição da Comunicação entre os Objetos .....	51
3.3.5 Utilização do Mecanismo de Herança .....	53
3.3.6 Implementação do Projeto .....	55
<b>CAPÍTULO 4 IMPLEMENTAÇÃO E VALIDAÇÃO DO SAVAD</b> .....	<b>56</b>
4.1 Dimensão do Projeto .....	56
4.2 Identificação do Ambiente Operacional e Capacidade de Processamento ...	57
4.3 Arquivos do SAVAD .....	57
4.4 Identificação dos Modelos de Redes .....	58
4.5 Ativação do Simulador Digital .....	62
4.6 Dinâmica de Funcionamento do SAVAD .....	64
4.7 Validação do SAVAD .....	69

<b>CAPÍTULO 5 CONCLUSÕES E SUGESTÕES</b> .....	76
5.1 Expansão do SAVAD .....	78
APÊNDICE A - DIAGRAMAS DE HERANÇA DAS CLASSES .....	81
APÊNDICE B - DIAGRAMAS CLIENTE-SERVIDOR .....	88
APÊNDICE C - MENSAGENS DE ERRO/AJUDA .....	101
APÊNDICE D - LIMITES UTILIZADOS NA IMPLEMENTAÇÃO DO SAVAD .....	104
APÊNDICE E - MANUAL DO USUÁRIO .....	105
REFERÊNCIAS BIBLIOGRÁFICAS .....	106

## LISTA DE FIGURAS

2.1	Elementos de Modelagem do SAVAD.....	13
2.2	Modelagem de uma Clínica Médica .....	15
2.3	Modelagem de uma Conexão de Transporte com o Protocolo de Janela Deslizante .....	18
2.4	Modelagem da Integração de Voz e Dados em Redes Locais com Passagem de Ficha.....	22
2.5	Módulos do SAVAD.....	25
2.6	Interface Básica.....	26
2.7	Tela de Ajuda Sensível ao Contexto ("help on-line.....	27
2.8	Definição da Estação de Serviço.....	28
2.9	Definição da Rota do Modelo.....	28
2.10	Resultados da Avaliação do Sistema M/M/1 .....	29
3.1	Notação para Representação de Classes.....	36
3.2	Notação para Instanciação de Objetos .....	36
3.3	Notação para Relacionamento de Classes "Amigas" .....	37
3.4	Notação para Herança entre Classes .....	38
3.5	Diagrama de Herança de Classes para as Soluções Analíticas do SAVAD.....	39
3.6	Notação para os Diagramas Cliente-Servidor.....	40
3.7	Diagrama Cliente-Servidor do Mecanismo de Controle de Eventos pelo SAVAD.....	41
3.8	Notação para os Diagramas de BOOCH das Classes.....	44
3.9	Diagrama de BOOCH da Classe zSorvedouro .....	45

3.10	Notação dos Diagramas Herança/Cliente-Servidor das Classes .....	46
3.11	Diagrama Herança/Cliente-Servidor da Classe zDecisão .....	47
3.12	Diagrama Geral Cliente-Servidor da Classe “Nome e o Tipo do Elemento” .....	52
3.13	Diagrama Geral Cliente-Servidor com a Expansão do Sub-Sistema “TelaArquivo” .....	52
3.14	Diagrama de Herança das Classes de Edição .....	53
3.15	Diagrama de Herança das Classes dos Elementos do SAVAD .....	54
4.1	Diagrama Cliente-Servidor do Acesso Global ao Modelo e Monitoração da Interface .....	65
4.2	Diagrama Cliente-Servidor da Inclusão/Alteração de Elementos do Modelo .....	67
4.3	Diagrama Cliente-Servidor da Verificação de Integridade e Solução do Modelo .....	68
4.4	Modelo de Rede M/M/1 .....	69
4.5	Modelo de Rede de Filas Abertas- Teorema de Jackson .....	70
4.6	Modelo de Rede de Filas Fechadas- Análise do Valor Médio .....	73

## **RESUMO**

Sistemas que exibem contenção de recursos são comumente modelados usando o paradigma de redes de filas. Exemplos de tais sistemas são redes de computadores, sistemas de tráfego, processos industriais, agências bancárias, etc. Esta Dissertação apresenta o SAVAD, um sistema especialista para avaliação de desempenho de modelos de redes de filas. SAVAD escolhe e executa, eficientemente, a técnica mais adequada para a solução do modelo proposto pelo usuário (analítica ou numérica) e fornece as medidas de desempenho relevantes solicitadas. SAVAD oferece uma interface poderosa, flexível, inteligente e amigável, com elementos de modelagem que permitem ao usuário especificar facilmente modelos de redes de filas. SAVAD foi desenvolvido utilizando a abordagem orientada a objetos. As características de modularidade, extensibilidade e reutilização de software inerentes a esta abordagem permitem que SAVAD seja facilmente enriquecido com novos elementos de modelagem e novas opções de soluções analíticas e numéricas.

## ABSTRACT

Systems that exhibit resource contention usually are modelled using the networks of queues paradigm. Examples of that systems are computer networks, traffic systems, manufacturing process, bank branches, etc. This dissertation introduces SAVAD, an expert system for performance evaluation of models for networks of queues. SAVAD chooses and efficiently applies the most adequate technique (analytic or numeric) in order to solve the user proposed model, issuing the requested pertinent performance measures. SAVAD provides a powerful, flexible, intelligent and friendly interface, which allows the user to easily model networks of queues using modeling elements. The object-oriented approach was used in designing and implementing SAVAD. The nice characteristics inherited from this approach (namely: software reusability, extensibility and modularity) allow SAVAD to be easily enrichable by new modeling elements and by new options for both analytical and numerical solutions.

# CAPÍTULO 1

## INTRODUÇÃO

A análise de desempenho de um sistema é essencial para verificar o seu comportamento com respeito às demandas solicitadas. Normalmente, a análise de desempenho é feita observando-se o comportamento de algumas medidas de interesse com as variações da carga do sistema. Essa análise pode ser feita com o sistema em funcionamento através de campanhas de medição ("workbench"). É importante porém, que seja feita a nível de projeto para garantir a especificação e uso adequado bem como reduzir os custos totais.

Um mecanismo de análise de um sistema é a sua simplificação e representação em forma de um modelo. Basicamente, o modelo procura representar os aspectos mais relevantes do problema descartando os demais [Li & Unger, 1987]. O modelo não deve ser excessivamente simplificado pois pode não refletir o comportamento real do problema em análise, invalidando os resultados obtidos nem deve ser muito complexo pois vai dificultar a análise. Desta forma, o modelo deve ser detalhado o suficiente, de forma a refletir o comportamento do problema em análise e permitir simulações de seu funcionamento em condições controladas. A modelagem deve ser executada por um especialista com o conhecimento geral do problema e em modelagem.

Sistemas que exibem contenção de recursos podem ser modelados usando o paradigma de redes de filas [Lages & Nogueira, 1986]. Exemplos de tais sistemas são redes de computadores, sistemas de tráfego, processos industriais, agências bancárias, etc.

Uma rede de filas é um sistema com múltiplas filas interconectadas de acordo com uma topologia, operando assíncrona e concorrentemente. Em uma rede de filas, clientes (ou fregueses) ficam em uma fila de algum nó até que sejam atendidos de acordo com uma disciplina especificada, passando em seguida para o nó seguinte da rede. A população da rede corresponde ao número de clientes que transitam em um dado instante pelos diversos nós da rede.

Redes de filas podem ser classificadas em abertas, fechadas e mistas [Kleinrock, 1975]. Redes abertas se caracterizam pela topologia aberta com a entrada de novos clientes na rede, passagem pelos diversos nós até serem excluídos ou retirados da rede. As redes abertas podem ter uma população estável ou não. Nas redes abertas com população estável, o número de clientes se mantém constante com o tempo. À medida que um cliente é retirado da rede em uma extremidade, um outro cliente é inserido na outra. Quando a população não é estável, o número de clientes pode flutuar com o tempo.

Redes fechadas caracterizam-se pela topologia fechada, em que o número de clientes circulando na rede é constante. Neste caso não existe a inclusão ou retirada de clientes da rede.

Redes mistas combinam características das duas outras, com trechos abertos e trechos fechados.

Na solução de modelos de redes de filas podem ser utilizadas técnicas analíticas ou numéricas. As técnicas analíticas são baseadas na Teoria das Filas [Kleinrock, 1975] [Sauer & Chandy, 1981] sendo mais adequadas por fornecerem resultados rápidos e precisos. Nem sempre é possível a utilização destas técnicas devido à inexistência de soluções para o modelo desejado ou devido à sua própria complexidade. Neste caso deve-se procurar solucionar o modelo através de técnicas numéricas. Dentre estas técnicas, geralmente, emprega-se a Simulação Digital [Cabral, 1990].

A avaliação de desempenho de modelos de redes de filas pode não ser simples. Normalmente, somente especialistas em avaliação de desempenho são capazes de executar tal tarefa.

Tradicionalmente para a avaliação de desempenho de modelos de redes de filas, foram utilizadas técnicas analíticas ou linguagens de simulação de propósito geral [Kreutzer, 1986], tais como GPSS [Schriber, 1974], SIMULA [Birtwistle, 1973], etc. Atualmente a ênfase nesta área é na construção de ambientes de modelagem integrados que empreguem técnicas de Inteligência Artificial (sistemas especialistas).

Um sistema especialista procura reproduzir em um computador o conhecimento existente em determinada área. O sistema especialista dispõe de um motor de inferência que analisa as informações disponíveis procurando reconhecer padrões e tomar as decisões adequadas [Harman & King, 1988]. Para isto ele dispõe de uma base de conhecimento que contém as Regras e os Fatos que armazenam o conhecimento do sistema. À medida que a base de conhecimento é enriquecida, o sistema fica mais apto a reconhecer novos modelos e se tornar mais abrangente.

Um ambiente de modelagem integrado deve permitir aos usuários descrever e modificar representações dos sistemas reais ou hipotéticos de forma simples. Deve permitir a construção de modelos sem obrigar o usuário a aprender uma nova linguagem de modelagem. Deve também dispor de uma interface que permita o fornecimento das informações descritivas do modelo sem exigir que o usuário conheça o formato dos dados. O usuário deve poder fazer modificações simples no modelo, sem que sejam necessárias modificações no projeto do sistema, bem como possibilitar a este selecionar os tipos e as formas de saída desejadas.

Existem alguns ambientes de modelagem integrados, tais como RESQ [Sauer e MacNair, 1985], INT3 [Lehmann, 1986], QNAP2 [Poitier, 1986], NATSIM [Khoshnevis & Austin, 1986] e o sistema descrito em [Lehmann et al., 1986]. Esses ambientes necessitam de um alto grau de sofisticação em termos de hardware e software, e do conhecimento prévio do usuário sobre a técnica que deve ser usada para solucionar o modelo.

Ambientes de modelagem integrados sugerem sistemas de software complexos, devendo-se utilizar metodologias estruturadas e seguras. Dentre essas destaca-se a abordagem orientada a objetos.

A abordagem orientada a objetos permite o desenvolvimento de sistemas complexos de maneira natural e bem estruturada. Os elementos do problema são representados por suas propriedades, comportamentos e comunicação entre si, permitindo alto grau de abstração, encapsulamento e modularidade [Carvalho, 1990]. Tais características são da maior importância no desenvolvimento de sistemas complexos.

## 1.1 Objetivos

Esta dissertação tem como objetivo o desenvolvimento de um ambiente de modelagem integrada denominado Sistema de Avaliação de Desempenho de Modelos de Redes de Filas - SAVAD. SAVAD constitui-se em sistema especialista onde, a partir de um modelo de redes de filas proposto pelo usuário, escolhe e aplica o método mais adequado para solucionar este modelo, fornecendo as medidas de desempenho relevantes. Para isto SAVAD utiliza técnicas analíticas, baseadas na Teoria das Filas, e a técnica Simulação Digital.

O SAVAD oferece as seguintes facilidades:

- Uma interface amigável, inteligente e integrada para a especificação do modelo de rede de filas.
- Um conjunto de elementos poderosos e flexíveis, que permitem facilmente especificar o modelo desejado de forma simples, genérica e abrangente.
- Resolução do modelo através da técnica mais eficiente, analítica ou por Simulação Digital. Opcionalmente, o usuário pode solicitar que o modelo seja solucionado por Simulação Digital, independente da existência de solução analítica.
- As medidas de desempenho relevantes para os modelos.
- Possibilidade de ampliação dos elementos de modelagem bem como das opções de soluções dos modelos (analíticas e numéricas). Esta ampliação é facilitada pelas características de modularidade, extensibilidade e reutilização de software da abordagem orientada a objetos utilizada na construção do SAVAD.
- Possibilidade de utilizar sua interface para a especificação de modelos de redes de filas, acoplando outros módulos de solução mais específicos.

SAVAD integra um Simulador Digital de modelos de redes de filas, desenvolvido como Dissertação de Mestrado pelo aluno Haroldo Conceição Filho, que funcionará de forma integrada ao sistema conforme descrito em [Conceição Filho et al., 1992].

Na implementação do SAVAD foi utilizada a linguagem de programação C++. A linguagem C++ incorpora os conceitos básicos da programação orientada a objetos mantendo a eficiência e facilidade da linguagem C [Stroustrup, 1986].

O SAVAD faz parte de um projeto mais abrangente denominado "Um Ambiente de Simulação Inteligente para Avaliar o Desempenho de Sistemas Distribuídos", em desenvolvimento pelas Áreas de Redes de Computadores e Inteligência Artificial da Universidade Federal da Paraíba, e aprovado e financiado pelo CNPQ [Cabral, 1992].

## 1.2 Contribuição Científica

O SAVAD é um ambiente de modelagem integrada com uma interface amigável.

Os elementos empregados para a descrição dos modelos de redes de filas são aqueles propostos em [Cabral, 1989] e apresentam grande flexibilidade, possibilitando a modelagem de uma grande variedade de sistemas que apresentem contenção de recursos.

São empregadas técnicas de Inteligência Artificial no projeto do sistema para o reconhecimento dos modelos de redes de filas submetidos ao SAVAD e para a escolha mais adequada de sua solução, seja analítica ou numérica. Desta forma os usuários do SAVAD não necessitam obrigatoriamente conhecer com profundidade técnicas de modelagem e de avaliação de desempenho de sistemas de redes de filas.

Esta Dissertação contribui para o desenvolvimento da abordagem orientada a objetos, especialmente em projetos de porte significativo. Devido às características do projeto, são explorados diversos aspectos da abordagem orientada a objetos com sugestões para a extensão/adaptação da notação existente. Sob alguns aspectos a presente dissertação se propõe a servir como um modelo de documentação de projetos orientados a objetos.

O funcionamento do SAVAD em equipamentos de pequeno porte (microcomputadores compatíveis com IBM-PC), em ambiente MS-DOS, torna possível sua utilização em larga escala.

A modularidade, extensibilidade e reutilização de software possibilitadas pela abordagem orientada a objetos facilita a ampliação e otimização do SAVAD em versões posteriores, bem como sua portabilidade para outros ambientes como UNIX e WINDOWS.

## 1.3 Organização

A presente dissertação está organizada em 5 (cinco) capítulos e 5 (cinco) apêndices. Os demais capítulos desta dissertação estão organizados conforme segue:

**Capítulo 2:** Descreve o SAVAD e seus elementos de modelagem.

**Capítulo 3:** Apresenta o projeto orientado a objetos do SAVAD bem como sugestões acerca da notação utilizada.

**Capítulo 4:** É feita uma descrição acerca da implementação e são apresentados exemplos de modelo de redes de filas e analisados seus resultados.

**Capítulo 5:** Apresenta as conclusões e são feitas sugestões sobre o aperfeiçoamento do projeto.

Os apêndices apresentam o Manual do Usuário, diagramas de herança das classes, diagramas cliente-servidor das classes, mensagens de erro/ajuda e limites usados na implementação do SAVAD.

Informações mais detalhadas sobre a documentação do projeto orientado a objetos do SAVAD tais como os protocolos das classes do SAVAD, diagramas de herança/cliente-servidor e diagramas de Booch estão disponíveis sob a forma de Relatório Técnico [Souto, 1993].

## CAPÍTULO 2

### SAVAD

O presente capítulo fornece uma visão geral do SAVAD. São apresentados os seus elementos de modelagem; exemplos de modelagem; seus módulos básicos e, finalmente, as medidas de desempenho fornecidas pelo SAVAD [Souto et al., 1992] [Cabral et al., 1993].

A solução de um modelo de redes de filas para a determinação das medidas de desempenho com o auxílio do SAVAD é obtida de acordo com as seguintes etapas:

1. O usuário representa o problema existente em um modelo de redes de filas utilizando os elementos de modelagem do SAVAD.
2. Os elementos do modelo são descritos individualmente para o SAVAD.
3. O SAVAD faz a verificação da integridade do modelo especificado.
4. O modelo de rede é solucionado. O SAVAD procura a técnica de solução mais adequada. Caso não exista solução analítica, o SAVAD submete o modelo à Simulação Digital. Esta opção também pode ser solicitada pelo usuário, independente da existência de solução analítica.
5. Apresentação para o usuário das medidas de desempenho do modelo solucionado.
6. Montagem de configurações alternativas para o modelo, analisando-se o seu comportamento em diversas situações.

Apresentamos a seguir os elementos utilizados na modelagem de redes de filas e a estrutura geral do SAVAD.

#### 2.1 Elementos de Modelagem

Para a modelagem de sistemas de redes de filas são utilizados elementos que descrevem o comportamento do problema a ser analisado [Cabral et al., 1989].

Os elementos do SAVAD possuem flexibilidade para permitir a modelagem de forma sofisticada.

O SAVAD oferece os seguintes elementos de modelagem:

#### a) Clientes

Clientes são entidades temporárias que circulam através dos elementos (nós) do modelo solicitando serviços.

O caminho a ser seguido por um cliente é denominado *Rota*. Associado a um cliente existe uma *classe*. Clientes de uma mesma classe possuem os mesmos atributos. As classes permitem que clientes com comportamentos diversos possam ser caracterizados e analisados de formas distintas.

Clientes são componentes virtuais do modelo na medida em que não são especificados explicitamente pelo usuário. Informações sobre clientes são obtidas durante o processamento através dos demais elementos do modelo.

Podemos caracterizar como clientes mensagens que trafegam em uma rede de computadores, peças em uma linha de montagem ou pessoas que procuram atendimento em uma agência bancária.

#### b) Estações de serviço

Estações de serviço, representam os recursos de um modelo de redes de filas. Podemos citar como estações de serviço canais de transmissão em uma rede de computadores, caixas atendendo clientes em uma agência bancária, etc.

Uma estação de serviço pode ter um ou mais servidores e apenas um cliente é atendido de cada vez por um servidor. Os clientes chegam a uma estação de serviço de acordo com um processo de chegada. Ao chegar à estação, se algum servidor estiver livre, o cliente será atendido, caso contrário entrará em uma fila de espera.

Os intervalos entre chegadas de clientes sucessivos em uma estação de serviço são determinados por uma Função de Distribuição de Probabilidade (FDP) do tipo:

- Exponencial
- Uniforme
- Normal
- Determinística
- Geral (definida pelo usuário através de coordenadas de uma dada Função de Distribuição de Probabilidade).

São permitidos os tipos de Estações:

- Servidor simples (um único servidor)
- Servidor múltiplo (diversos servidores)
- Servidor infinito (número ilimitado de servidores).

A fila dos servidores simples e dos servidores múltiplos pode ter comprimento limitado ou não. Servidores infinitos não têm filas, isto é, sempre que um cliente chega à estação de serviço, há um servidor disponível para atendê-lo.

O atendimento de clientes é determinado por uma disciplina de escalonamento que pode ser do tipo:

- FCFS ("First Come, First Served")
- LCFS ("Last Come, First Served")
- "Header of Line"
- Tempo
- Aleatória

Uma estação de serviço pode ter uma só entrada ou várias. Este último caso pode ocorrer quando a estação for compartilhada por várias rotas ou no caso de uma rota com realimentação.

A estação de serviço pode dispor de uma única saída ou de várias. Caso existam várias saídas, as probabilidades de envio de clientes para cada uma delas podem ser iguais ou diferentes. Se as probabilidades de passagem de clientes forem diferentes, devem ser especificadas pelo usuário, respeitando-se o valor 1 (um) para sua soma. Caso sejam iguais não necessitam ser especificadas.

### c) Fontes

Fontes geram clientes que circulam em modelos de redes abertas. Os clientes são gerados segundo uma Função de Distribuição de Probabilidade similar àquelas disponíveis para as estações de serviço (exponencial, uniforme, determinística, normal e geral).

### d) Sorvedouros

Sorvedouros são elementos que eliminam clientes em modelos de redes abertas.

### e) Pontos de controle

Pontos de controle são elementos que permitem controlar o fluxo de clientes nos modelos de redes. Esses elementos foram projetados para fornecer maior flexibilidade e facilidade na construção do modelo. Os pontos de controle oferecidos pelo SAVAD são os seguintes:

- 1. Ponto de multiplicação:** Multiplica um único cliente que chega a este nó em vários. Pode possuir uma entrada e "n" saídas. Pode ser usado para modelar pacotes de dados com tipos diferentes de informação que devem receber tratamentos distintos no receptor tais como "pacotes de dados" e "reconhecimento". Também pode ser usado para representar um demultiplexador.

**2. Ponto de fusão:** Faz a fusão de dois ou mais clientes em um único cliente. Pode possuir " $n$ " entradas e uma saída. Pode ser usado para modelar uma multiplexação de informações. No receptor a mensagem será reconstituída e entregue aos seus destinatários. Os clientes que serão fundidos devem aguardar nas diversas entradas para que a fusão ocorra de forma simultânea para todos os clientes, gerando um novo cliente na saída.

**3. Ponto de Sincronização:** Destina-se a bloquear clientes até que uma condição seja satisfeita. Para isto o número de clientes em cada uma das suas filas de entrada deve atingir um limite especificado pelo usuário.

Um ponto de sincronização pode possuir as seguintes configurações:

- a) Duas entradas e duas saídas correspondentes.
- b) Duas entradas e uma saída associada a uma das entradas.
- c) Uma entrada e uma saída correspondente.

No primeiro caso a liberação de clientes para as saídas ocorrerá desde que a quantidade de clientes nas duas filas de entrada atinja os limites especificados para essas filas, simultaneamente.

No segundo caso a liberação de clientes para a saída ocorrerá apenas a partir de uma das filas de entrada e desde que a quantidade de clientes nas duas filas de entrada atinja os limites especificados nas duas filas, simultaneamente.

No último caso, a liberação de clientes para a saída desejada dependerá do número de clientes e do limite especificado para a fila de entrada.

Podem existir as seguintes situações, de acordo com os limites especificados para as filas de entrada (" $n$ "):

- Liberação de " $n$ " clientes. A especificação do número de clientes é feita usando um *inteiro positivo*.
- Liberação de *até* " $n$ " clientes. A especificação do número de clientes é feita usando um *inteiro negativo*.
- Passagem *livre* de clientes na outra saída. A especificação é feita usando " $0$ " (*zero*).

**4. Ponto Escalonador:** Permite o escalonamento de clientes de acordo com uma disciplina estabelecida pelo usuário [Brasileiro et al., 1992]. Um ponto escalonador pode possuir uma ou mais filas. Os clientes de uma mesma fila devem ser de mesma classe. As filas podem ter comprimento limitado ou não. A liberação dos clientes de uma fila obedece a uma disciplina especificada. Se uma fila selecionada para liberar clientes estiver vazia o direito de liberação passa para a próxima fila.

O escalonamento de filas ocorre conforme uma das disciplinas:

- **Cíclica:** As filas são escalonadas conforme uma sequência contendo seus números associados. Após a seleção da última fila da sequência o escalonamento volta para a primeira fila. As filas podem ser referenciadas diversas vezes dentro da sequência, aumentando dessa forma sua prioridade ou obedecendo a um controle desejado.

A disciplina de liberação de clientes nas filas pode ser feito de forma:

- **Não exaustiva:** (somente um cliente é liberado em cada fila).
- **Limitada:** (até "n" clientes são liberados em cada fila).
- **Exaustiva:** (Todos os clientes da fila são liberados).
- **Aleatória:** As filas são escalonadas de forma aleatória, podendo liberar um cliente de cada vez.
- **Livre:** Neste caso qualquer fila que não estiver vazia pode tentar liberar um cliente. Se mais de uma fila tentar liberar um cliente ao mesmo tempo ocorre uma colisão e estas filas ficam bloqueadas por um período de tempo determinado por uma Função de Distribuição de Probabilidade especificada pelo usuário.

#### f) Classe

As classes são usadas para determinar tipos diferentes de clientes em circulação no modelo. Associada a cada classe existe uma prioridade (inteiro não negativo). Como exemplos de classes diversas em um mesmo sistema podemos citar em uma rede de computadores, pacotes de dados e pacotes de reconhecimento.

#### g) Rota

As rotas estabelecem os diversos caminhos existentes na rede por onde os clientes circulam. O modelo de rede é definido pelo estabelecimento de rotas com classes especificadas unindo os diversos elementos. O roteamento no SAVAD é fixo.

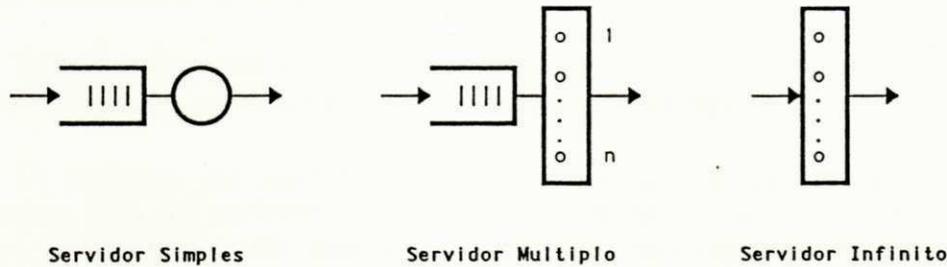
Associadas às rotas existem classes de clientes de modo que possam ser dados tratamentos diferenciados para os diversos tipos de clientes.

Rotas podem ser abertas ou fechadas. Rotas abertas necessitam de elementos que gerem e eventualmente eliminem clientes. Rotas Fechadas possuem um número limitado de clientes em circulação. Rotas podem possuir ou não realimentação.

Clientes de rotas distintas podem compartilhar elementos.

No SAVAD uma rota é especificada através da definição da sequência dos seus elementos constituintes, na ordem desejada.

Na figura 2.1 temos os símbolos utilizados para a representação gráfica dos elementos do SAVAD.



a) Tipos de Estacoes de Servico

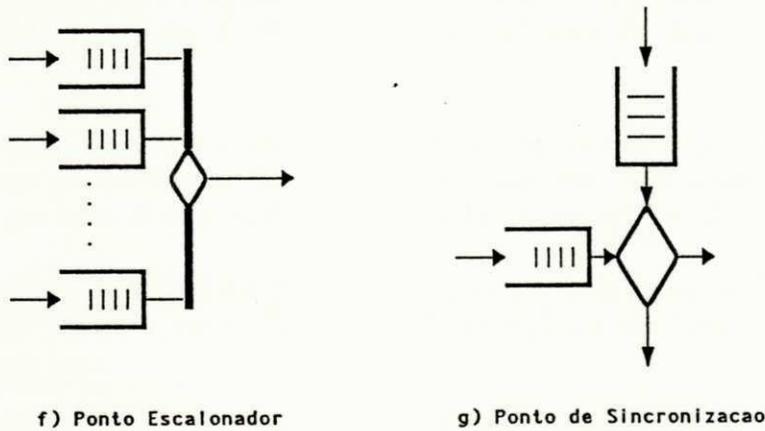
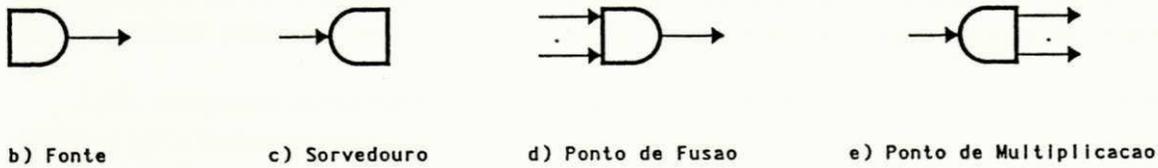


Fig 2.1 - Elementos de Modelagem do SAVAD

## 2.2 Exemplos de Modelos

Apresentamos três exemplos de modelos construídos com os elementos do SAVAD. O primeiro exemplo apresenta a modelagem de uma clínica médica. O segundo exemplo apresenta a modelagem de uma conexão de transporte com controle de fluxo usando o protocolo de janela deslizante, conforme descrito em [Cabral et al., 1989]. O terceiro exemplo apresenta a modelagem de integração de voz e dados em redes locais com passagem de ficha, conforme descrito em [Brasileiro et al., 1991].

### 2.2.1 Modelagem de uma Clínica Médica

Temos a seguir um exemplo de modelagem utilizando os elementos do SAVAD. O modelo procura descrever um fluxo de pacientes em uma clínica médica.

Os pacientes são recebidos e registrados na recepção da clínica. A partir do seu estado aparente, 80% dos pacientes recebidos são encaminhados para um dos 8 médicos (clínicos) para fazer uma consulta. Os demais pacientes são imediatamente enviados para um setor de emergência onde são atendidos por uma equipe médica.

Os médicos, após efetuarem a consulta dos seus pacientes, fazem o seguinte encaminhamento: 70% são atendidos e retornam para casa, 10% são enviados para o setor de emergência e os 20% restantes são encaminhados para o setor de exames médicos, devendo depois retornar para uma consulta complementar, sem necessitar passar novamente pela recepção.

Após receberem atendimento no setor de emergência, 40% dos pacientes ficam internados na clínica e 60% retornam para suas casas.

Os intervalos entre chegadas de clientes sucessivos na clínica e os tempos de atendimento em seus diversos setores são distribuídos exponencialmente e a disciplina de atendimento é do tipo FCFS.

A recepção da clínica, composta por 2 funcionárias, não impõe limite de atendimento para os pacientes que podem formar filas. As recepcionistas têm capacidade de atendimento médio de 30 pacientes por hora. A taxa média de chegada de pacientes é de 20 por hora.

Os médicos demoram em média 15 minutos em cada consulta. O setor de exames da clínica faz 5 exames por hora, em média. O setor de emergência tem capacidade de atender em média 10 pacientes por hora.

A figura 2.2 mostra o modelo de redes de filas usando elementos do SAVAD para o presente exemplo. No modelo uma fonte modela o processo de geração de pacientes para a clínica. As recepcionistas e os médicos são representados por estações de serviço com múltiplos servidores. O setor de emergência e o setor de exames são representados por estações de serviço com servidor único, abstraindo-se suas complexidades.

As medidas de desempenho relevantes são as seguintes: vazão média (quantidade de pacientes atendidos por hora), tempo médio de espera em fila pelos pacientes, comprimento médio das filas, fator de utilização das estações de serviço e o tempo médio de atendimento dos pacientes. A unidade de tempo considerada é um minuto.

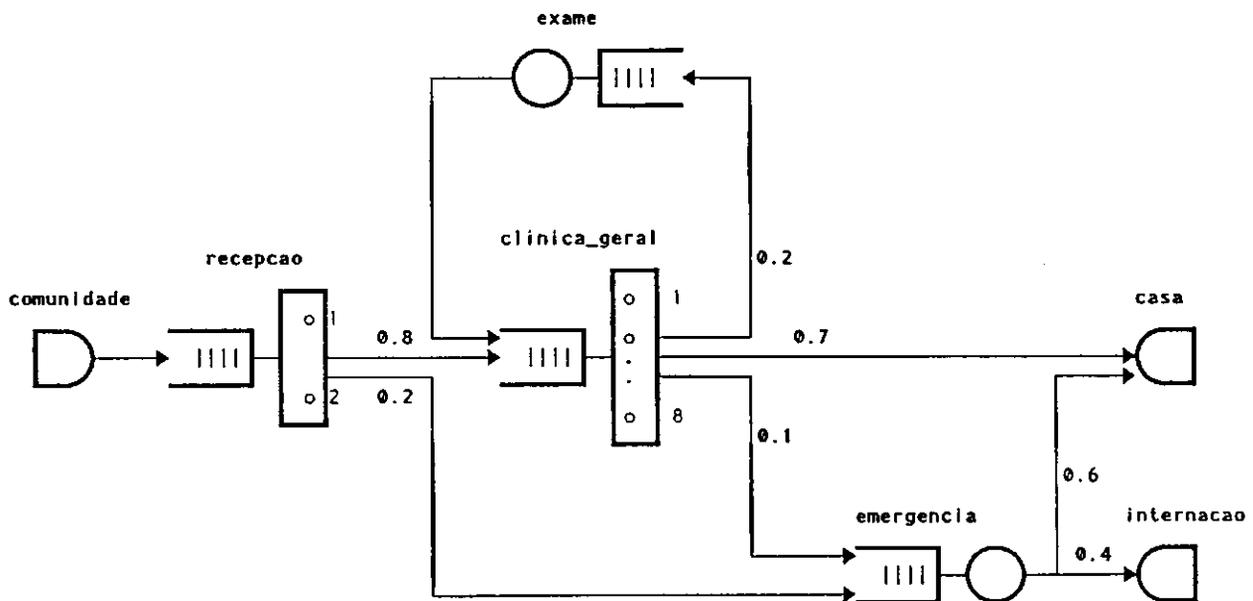


Fig. 2.2 - Modelagem de uma Clínica Médica

Os elementos do modelo com os respectivos parâmetros são mostrados a seguir:

**Clientes:** Pacientes a serem atendidos pela clínica.

**Fonte:**

Nome: comunidade

Distribuição do intervalo entre chegadas de clientes sucessivos: Exponencial - Média: 3 min.  
(20 pacientes/h)

**Servidores:**

a) Nome: recepção

Número de servidores: 2

Disciplina de escalonamento: FCFS

Comprimento da fila: Sem limite

Distribuição: Exponencial - Média: 4 min. (30 pacientes/h para as 2 recepcionistas)

b) Nome: clínica\_geral

Número de servidores: 8

Disciplina de escalonamento: FCFS

Comprimento da fila: Sem limite

Distribuição: Exponencial - Média: 15 min.

c) Nome: emergência

Número de servidores: Único

Disciplina de escalonamento: FCFS

Comprimento da fila: Sem limite

Distribuição: Exponencial - Média: 6 min. (10 pacientes/h)

d) Nome: exame

Número de servidores: Único

Disciplina de escalonamento: FCFS

Comprimento da fila: Sem limite

Distribuição: Exponencial - Média: 12 min. (5 exames/h)

**Sorvedouros:**

a) Nome: casa

b) Nome: internação

**Classe:**

Nome: clas\_paciente

Prioridade: Sem prioridade (0)

**Rota:***comunidade* → *recepção* → (0.8 *clínica\_geral*, 0.2 *emergência*)*clínica\_geral* → (0.7 *casa*, 0.1 *emergência*, 0.2 *exame*)*emergência* → (0.6 *casa*, 0.4 *internação*)*exame* → *clínica\_geral*

### 2.2.2 Conexão de Transporte com o Protocolo de Janela Deslizante

O modelo consiste de uma conexão de transporte tendo como mecanismo de controle de fluxo, o protocolo de janela deslizante. Esse modelo usa um sistema de espera, isto é, quando a janela fecha, mensagens que chegam a uma entidade de transporte ficam bloqueadas em fila, esperando que a janela abra para serem transmitidas pela conexão de transporte de referência à conexão de transporte parceira. A esta conexão de transporte atribui-se uma conexão de rede que é modelada por um conjunto de canais de transmissão. Esses canais são responsáveis pelo encaminhamento das mensagens na sub-rede de comunicação. Para cada mensagem entregue à entidade de destino, uma mensagem de reconhecimento é gerada e enviada à entidade de transporte de referência.

A entidade de transporte apresenta a distribuição dos tempos de interchegadas de mensagens do tipo exponencial com taxa média de  $\lambda$  mensagens por segundo. A sua capacidade de armazenamento é de  $k$  mensagens e a disciplina de escalonamento para as mensagens é do tipo FCFS. O comprimento da janela para a conexão de transporte é  $W$  mensagens. Cada canal da sub-rede de comunicação possui: fila sem restrição de comprimento, distribuição do tempo de transmissão exponencial com média de  $m$  mensagens por segundo e disciplina de escalonamento do tipo FCFS. Finalmente, assume-se que o tempo de geração e o tempo de transmissão de uma mensagem de reconhecimento sejam desprezíveis.

A figura 2.3 apresenta o modelo de redes de filas usando os elementos do SAVAD. Nesse modelo, a geração de mensagens na conexão de transporte é representada por uma fonte denominada *emissor*. A entidade de transporte é representada pelo ponto de sincronização denominado *pt\_controle* que possui duas entradas e uma saída. As filas associadas a essas entradas denominadas *pt\_controle.1* e *pt\_controle.2*, armazenam respectivamente, as mensagens que devem ser transmitidas pela conexão de transporte e as permissões para a transmissão de mensagens (comprimento da janela). A fila *pt\_controle.1* possui uma fila com capacidade de  $k$  mensagens. A fila *pt\_controle.2* possui uma fila com capacidade igual ao comprimento da janela ( $W$  mensagens). As filas do ponto de sincronização têm disciplina de escalonamento do tipo FCFS.

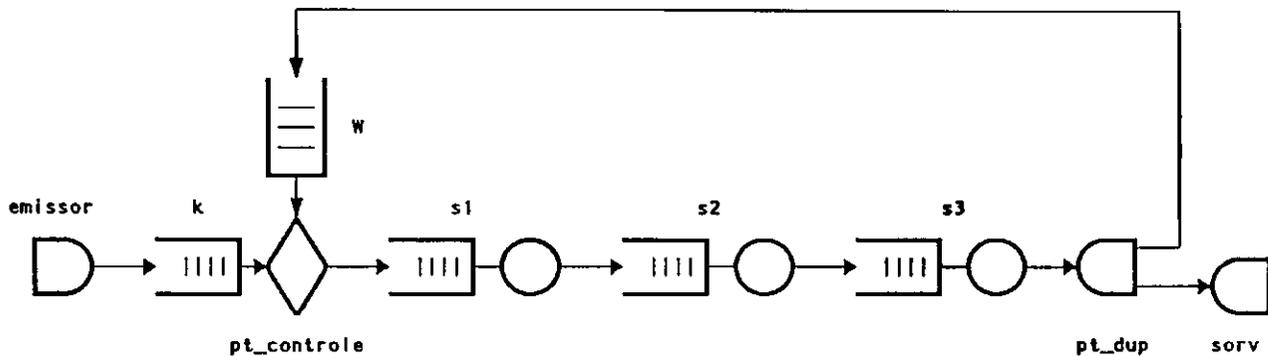


Fig. 2.3 - Modelagem de uma Conexão de Transporte com o Protocolo da Janela Deslizante

Uma mensagem liberada por *pt\_controle* percorre os canais de comunicação até chegar à estação de destino. Os canais de comunicação são modelados por servidores simples. Para efeito de simplificação do modelo não foi representado o atraso do envio de reconhecimento.

A entidade de transporte parceira é modelada por um ponto de duplicação, denominado *pt\_dup* e um sorvedouro denominado *sorv*. Cada mensagem que chega a esta entidade é duplicada pelo nó *pt\_dup*. Uma dessas mensagens segue para o sorvedouro, representando a entrega da mensagem à entidade de transporte par. A outra mensagem é devolvida à entidade de transporte, representando o envio de uma mensagem de reconhecimento ou permissão de transmissão.

As medidas de desempenho relevantes são as seguintes: vazão média (total de mensagens transmitidas por segundo), atraso médio de admissão (tempo médio em que as mensagens ficam bloqueadas numa entidade de transporte, esperando que a janela abra) e tempo médio de resposta fim-a-fim (atraso médio de admissão adicionado ao atraso médio de transmissão das mensagens nos canais de comunicação).

Os elementos do modelo com os respectivos parâmetros são mostrados a seguir:

**Clientes:** Mensagens de dados e mensagens de reconhecimento.

**Fonte:**

Nome: emissor

Distribuição: Exponencial - Média:  $\lambda$  mensagens/segundo

**Estações de serviço:**

a) Nome: s1

Número de servidores: Simples

Disciplina de escalonamento: FCFS

Comprimento da fila: Sem limite

Distribuição: Exponencial - Média: "m" mensagens/segundo

b) Nome: s2

Número de servidores: Simples

Disciplina de escalonamento: FCFS

Comprimento da fila: Sem limite

Distribuição: Exponencial - Média: "m" mensagens/segundo

c) Nome: s3

Número de servidores: Simples

Disciplina de escalonamento: FCFS

Comprimento da fila: Sem limite

Distribuição: Exponencial - Média: "m" mensagens/segundo

**Sorvedouros:**

Nome: sorv

**Ponto de duplicação:**

Nome: pt\_dup

**Ponto de sincronização:**

Nome: pt\_controle

Fila 1:

Disciplina de escalonamento: FCFS

Comprimento da fila: k

Fila 2:

Disciplina de escalonamento: FCFS

Comprimento da fila: W

**Classes:**

- a) Nome: dados  
Prioridade: Normal
- b) Nome: reconhecimento  
Prioridade: Normal

**Rotas:**

- a) Nome: rota1  
Classe: dados

*emissor* → *pt\_controle.1* → *s1* → *s2* → *s3* → *pt\_dup* → *sorv*

- b) Nome: rota2  
Classe: reconhecimento

*pt\_controle.2* → *s1* → *s2* → *s3* → *pt\_dup* → *pt\_controle.2*

**2.2.3 Modelo da Integração de Voz e Dados em Redes Locais com Passagem de Ficha**

As redes com passagem de ficha podem ter uma topologia em anel ou barra. Nestas redes, há uma ficha que passa de interface em interface para o estabelecimento da ordem de acesso ao meio. De acordo com o protocolo de comunicação para as redes com passagem de ficha, quando uma interface recebe uma ficha livre, esta tem o direito à transmissão de pacotes.

Considere-se um exemplo de uma rede local com passagem de ficha com  $N = 10$  interfaces, sendo  $V = 5$  com tráfego de voz e  $D = 5$  com tráfego de dados. A avaliação de desempenho de redes locais com passagem de ficha, é semelhante àquela de filas com serviço cíclico [Kuehn, 1979]. Devido à intratabilidade matemática da maioria dos problemas de filas com serviço cíclico, algumas suposições simplificadoras serão consideradas. Seguem essas suposições:

- As interfaces têm capacidade ilimitada de armazenamento de pacotes e disciplina de escalonamento do tipo FCFS.
- Os processos estocásticos de chegada de tráfego são estacionários e independentes; da mesma forma o são os processos de serviço (tempo de transmissão). Também supõe-se a independência estatística dos processos de chegada de tráfego e serviço.
- O processo de geração de pacotes de um interlocutor (fonte de voz) durante uma conversação é uma sequência de pacotes de comprimento fixo, espaçados de  $T_p$  segundos (gerados durante um surto de conversação), seguida de um intervalo de quietude equivalente ao comprimento do intervalo de silêncio. Esta sequência de pacotes e intervalos de quietude, alternam no tempo. A distribuição dos intervalos entre chegadas de

de pacotes de voz é do tipo geral, definida pelo usuário, conforme [Brasileiro, 1987]. A taxa média de geração de pacotes é especificada por  $\lambda_V$  pacotes por segundo.

- O tráfego de dados é gerado conforme um Processo de Poisson com parâmetro  $\mu_D$  pacotes por segundo.
- Para cada interface, a distribuição do tempo de transmissão de um pacote na sub-rede de comunicação é definida pelo usuário, de acordo com a natureza do tráfego. Portanto para o tráfego de voz (pacotes de comprimento fixo), o tempo de transmissão é fixo, definido por  $T_V$  segundos, enquanto para o tráfego de dados a distribuição do tempo de transmissão é exponencial com média  $m_D$  pacotes por segundo.
- Neste modelo, as medidas de desempenho de interesse são uma estimativa da probabilidade de perda de pacotes de voz (cauda da distribuição do atraso fim-a-fim) e atraso médio fim-a-fim (tempo médio de espera na fila adicionado ao tempo médio de transmissão na sub-rede de comunicação) para ambos os tipos de tráfego.

A figura 2.4 apresenta o modelo de redes de filas na integração dos tráfegos de voz e dados, em redes locais com passagem de ficha, usando os elementos disponíveis pelo SAVAD.

Na figura,  $f_{Vi}(f_{Di}) = 1,2,3,4,5$  representam as fontes de geração de pacotes de voz(dados), conforme o processo de geração de pacotes já mencionado. Os "buffers" em cada interface e protocolo de acesso de uma rede local com passagem de ficha são conjuntamente modelados por um ponto escalonador com 10 filas e disciplina de escalonamento cíclica não exaustiva. Os pacotes que chegam a cada fila do ponto escalonador, oriundos da fonte  $f_{Vi}(f_{Di})$  pertencem à classe voz  $cl_V$  (classe dados  $cl_D$ ). Nas redes locais com passagem de ficha, quando uma ficha livre chega a uma interface, esta tem o direito de transmitir um pacote (disciplina não exaustiva) na sub-rede de comunicação. Estas ações correspondem ao ponto escalonador retirar um pacote da respectiva fila (interface receptora da ficha livre) e transferi-lo para o servidor simples (nó REDE) com fila de tamanho igual a 1 ( $k = 1$ ). Neste servidor o pacote é servido imediatamente segundo uma distribuição geral do tempo de serviço especificado para cada classe. Observe-se que o tempo de serviço corresponde ao tempo na respectiva fila do ponto escalonador adicionado ao tempo de transmissão do pacote na sub-rede de comunicação. A obtenção das distribuições do tempo de serviço, para ambos os tráfegos, é mostrada em [Brasileiro, 1987]. Após o serviço desse pacote, o ponto escalonador direciona-se para uma outra fila segundo a sua disciplina de escalonamento. Os pacotes de voz (dados) servidos pelo nó REDE entram no nó sorvedouro  $sorv_V$  ( $sorv_D$ ), onde são eliminados do modelo, significando que estes pacotes foram transmitidos pela sub\_rede de comunicação e entregues aos seus respectivos destinos.

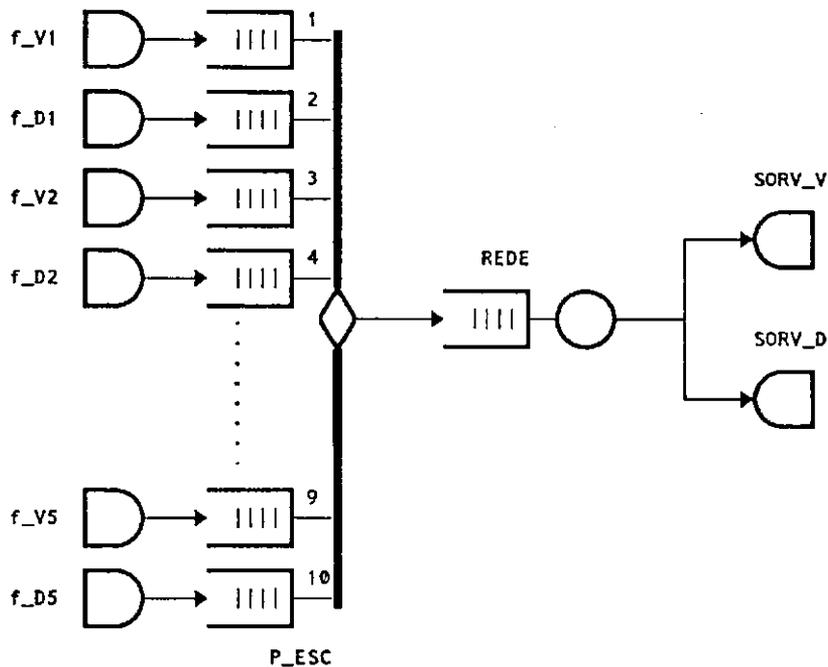


Fig. 2.4 - Modelagem da Integração de Voz e Dados em Redes Locais com Passagem de Ficha

Os elementos do modelo de integração de voz e dados em redes locais com passagem de ficha com seus respectivos parâmetros são sumarizados a seguir:

**Clientes:** Correspondem aos pacotes. Há pacotes de voz e pacotes de dados.

**Estação de serviço:**

- a) Nome: REDE
  - Número de servidores: Simples
  - Disciplina de Escalonamento: FCFS
  - Comprimento de fila (k): 1
  - Distribuição: Geral

**Ponto Escalonador:**

- a) Nome: P\_ESC
  - Número de filas (n): 10
  - Disciplina de escalonamento: cíclica (1-10)
  - Comprimento das filas: sem limite

**Fontes:**

a) Nomes:  $f_{V1}$ ,  $f_{V2}$ ,  $f_{V3}$ ,  $f_{V4}$ ,  $f_{V5}$

Distribuição: Geral

Média:  $\lambda v$  pacotes por segundo

b) Nomes:  $f_{D1}$ ,  $f_{D2}$ ,  $f_{D3}$ ,  $f_{D4}$ ,  $f_{D5}$

Distribuição: Geral

Média:  $\lambda d$  pacotes por segundo

### Sorvedouros:

a) Nomes:  $sorv_V$ ,  $sorv_D$

### Classes:

a) Nome:  $cl_V$

Prioridade: Não há (0)

b) Nome:  $cl_D$

Prioridade: Não há (0)

### Rotas:

a) Nome:  $r1$

Classe:  $cl_V$

$f_{V1} \rightarrow P_{ESC.1} \rightarrow REDE \rightarrow Sorv_V$

b) Nome:  $r2$

Classe:  $cl_V$

$f_{V2} \rightarrow P_{ESC.3} \rightarrow REDE \rightarrow Sorv_V$

c) Nome:  $r3$

Classe:  $cl_V$

$f_{V3} \rightarrow P_{ESC.5} \rightarrow REDE \rightarrow Sorv_V$

d) Nome:  $r4$

Classe:  $cl_V$

$f_{V4} \rightarrow P_{ESC.7} \rightarrow REDE \rightarrow Sorv_V$

e) Nome: r5  
Classe: cl\_V

$f_{V5} \rightarrow P\_ESC.9 \rightarrow REDE \rightarrow sorv\_V$

f) Nome: r6  
Classe: cl\_D

$f_{D1} \rightarrow P\_ESC.2 \rightarrow REDE \rightarrow Sorv\_D$

g) Nome: r7  
Classe: cl\_D

$f_{D2} \rightarrow P\_ESC.4 \rightarrow REDE \rightarrow Sorv\_D$

h) Nome: r8  
Classe: cl\_D

$f_{D3} \rightarrow P\_ESC.6 \rightarrow REDE \rightarrow Sorv\_D$

i) Nome: r9  
Classe: cl\_D

$f_{D4} \rightarrow P\_ESC.8 \rightarrow REDE \rightarrow Sorv\_D$

j) Nome: r10  
Classe: cl\_D

$f_{D5} \rightarrow P\_ESC.10 \rightarrow REDE \rightarrow Sorv\_D$

### 2.3 Estrutura do SAVAD

O SAVAD está estruturado de forma modular conforme apresentado na figura 2.5.

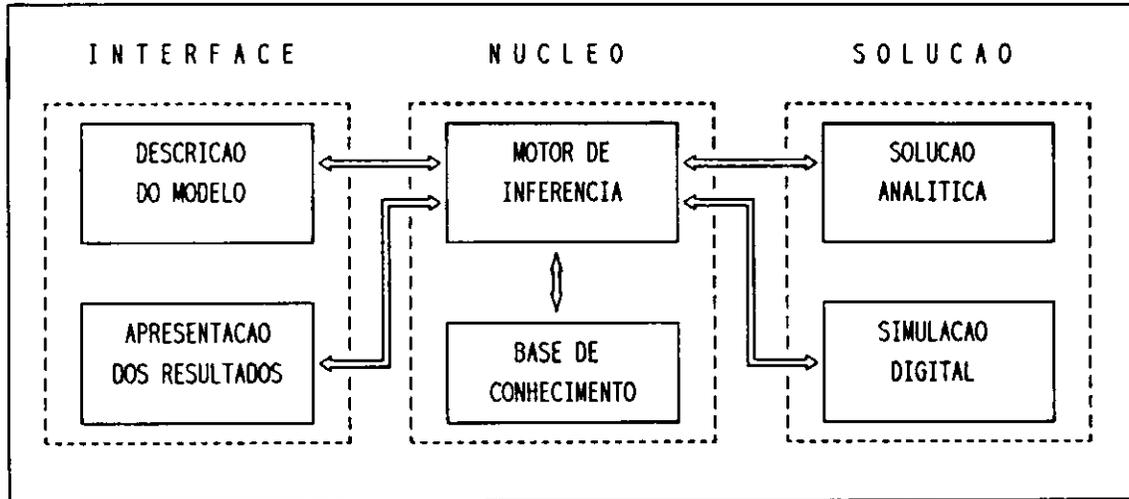


Fig. 2.5 - Módulos do SAVAD

- a) **Interface:** O usuário interage com o SAVAD através de sua interface. Ela permite ao usuário a especificação do modelo de redes de filas, a solicitação da solução do modelo (opcionalmente) e a apresentação dos resultados.
- b) **Núcleo do SAVAD:** O núcleo do sistema é composto de um motor de inferência e uma base de conhecimento.
- **Motor de Inferência:** Procura reconhecer o tipo de rede especificado pelo usuário a partir das informações disponíveis na base de conhecimento e qual a solução mais apropriada. O motor de inferência procura identificar a existência de solução analítica com um nível de complexidade computacional aceitável. Caso o modelo não possua solução analítica conhecida (registrada na base de conhecimento) o modelo será submetido a Simulação Digital. O usuário também poderá solicitar a Simulação Digital do modelo, independente deste poder ser solucionado analiticamente.
  - **Base de conhecimento:** Contém as regras e os fatos para o reconhecimento dos diversos tipos de redes que podem ser solucionadas pelo SAVAD.

c) **Solução:** O módulo solução é composto de uma biblioteca numérica e de um simulador digital.

Apresentamos a seguir os módulos do SAVAD. Detalhes acerca da implementação desses módulos encontram-se no capítulo 4.

### 2.3.1 Interface

O SAVAD oferece ao usuário uma interface inteligente e amigável com as seguintes facilidades: uma janela principal onde são feitas todas as operações, um menu principal e um menu auxiliar com as teclas de PF's ativas no rodapé da tela (figura 2.6). O usuário, de forma interativa, pode descrever o seu modelo, escolher a técnica desejada para solucioná-lo e receber as medidas de desempenho referentes ao modelo. Durante a descrição do modelo, a interface, automaticamente, valida cada um dos elementos especificados, e, antes de solucionar o modelo, valida-o globalmente [Cabral et al., 1992]. A interface também oferece facilidades tais como manipulação de arquivos e acesso direto aos comandos do sistema operacional (MS-DOS).

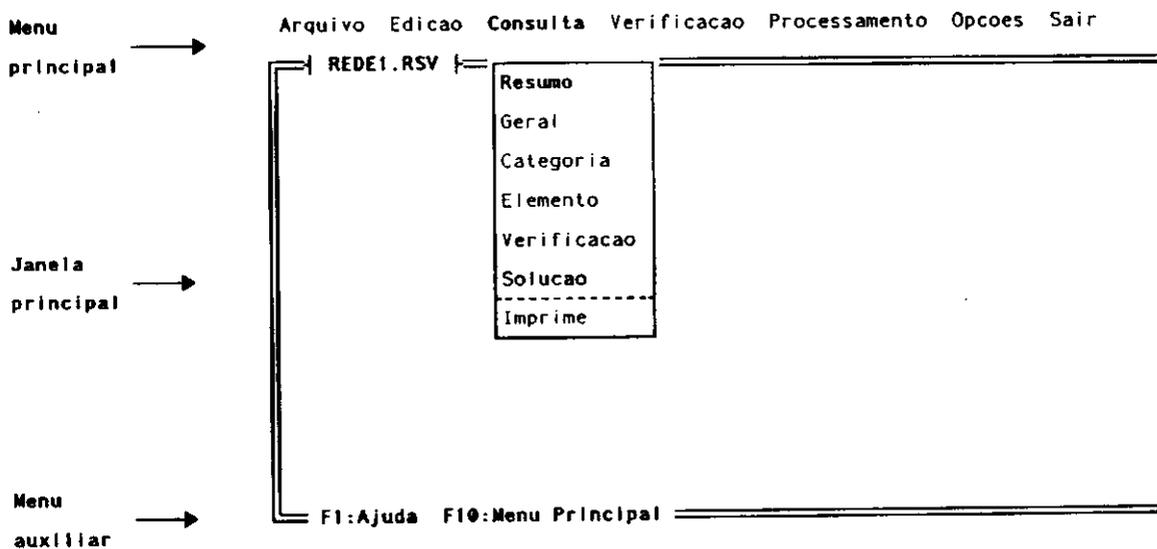


Fig. 2.6 - Interface Basica

Acionando-se a tecla <F1> em qualquer ponto da interface, o usuário disporá de um auxílio sensível ao contexto ("Help on-line"). A figura 2.7 apresenta um exemplo de auxílio sensível ao contexto. A tecla <F10> cancela a atividade corrente e faz o SAVAD retornar ao menu principal.

```

Arquivo  Edicao  Consulta  Verificacao  Processamento  Opcoes  Sair
-----
| REDE1.RSV - INSERCAO DE ELEMENTOS |
|
| TIPO.....: Estacao de Servico
| NOME.....: estacaol
| TIPO DE SERVIDOR.....: Multiplo
| COMPRIMENTO DA FILA.....: 5
| QUANT. DE SERVIDORES.....: —
| DISCIPLINA.....: Fcfs
| DISTRIBUICAO:
|   TIPO.....: Exponencial
|   MEDIA.....: 1.0000
|
| F1:Ajuda  F10:Menu Principal
  
```

Informe o valor correspondente  
Numero inteiro maior que 1

Fig. 2.7 - Tela de Ajuda Sensível ao Contexto ("help on-line")

O menu principal fornece as funções básicas do SAVAD, estando sempre visível para o usuário.

- **Arquivo:** Permite a manipulação dos arquivos descritos pelo modelo. Esta opção oferece as seguintes possibilidades: Criar um novo arquivo; ler, salvar, cancelar, copiar ou renomear um arquivo já existente; e apresentar um diretório desejado.
- **Edição:** Permite a especificação e alteração dos elementos do modelo, através das sub-opções: inserir, alterar, retirar, renomear e copiar elementos.
- **Consulta:** Permite consultar o SAVAD das seguintes formas: configuração do modelo, integridade do modelo, medidas de desempenho do modelo e sumário do modelo (número de elementos de um determinado tipo e informações acerca da integridade do modelo). A consulta pode ser feita por elemento, categoria de elementos ou do modelo global.
- **Verificação:** Testa a integridade do modelo especificado. Esta verificação também é feita automaticamente antes de solucionar qualquer modelo especificado ou alterado e ainda não verificado com sucesso, evitando a submissão para solução de modelos inconsistentes ou incompletos.

- **Processa:** Soluciona o modelo de rede, determinando as medidas de desempenho.
- **Opções/DOS:** Permite ao usuário executar diretamente comandos do sistema operacional (MS-DOS).
- **Sair:** Salva alterações eventuais efetuadas no modelo e encerra o SAVAD (retornando para o nível do sistema operacional).

As figuras a seguir representam algumas telas da interface mostrando a inserção de elementos do tipo estação de serviço (figura 2.8) e rota (figura 2.9). Para o sistema M/M/1, a rota é aberta, por isto os seus elementos são especificados em sua ordem natural de ocorrência. A figura 2.10 apresenta os resultados da avaliação de desempenho desse sistema.

```

Arquivo  Edicao  Consulta  Verificacao  Processamento  Opcoes  Sair
| REDE1.RSV - INSERCAO DE ELEMENTOS |
TIPO.....: Estacao de Servico
NOME.....: canal_com
TIPO DE SERVIDOR.....: Simples
COMPRIMENTO DA FILA.....: Sem limite
QUANT. DE SERVIDORES.....: Unico
DISCIPLINA.....: Fcfs
DISTRIBUICAO:
  TIPO.....: Exponencial
  MEDIA.....: 2.0000
FATOR DE DESENCORAJAMENTO: 0.0000
F1:Ajuda  F10:Menu Principal

```

Fig. 2.8 - Definicao da Estacao de Servico

```

Arquivo  Edicao  Consulta  Verificacao  Processamento  Opcoes  Sair
| REDE1.RSV - INSERCAO DE ELEMENTOS |
TIPO.....: Rota
NOME.....: rota_dados
CLASSE.....: class_dados
| ROTA |
transmissor → canal_com → receptor
TIPO.....: Aberta
POPULACAO.....: 0
F1:Ajuda  F9:Fim edicao  F10:Menu Principal

```

Fig. 2.9 - Definicao da Rota do Modelo

Arquivo Edicao Consulta Verificacao Processamento Opcoes Sair

```
REDEXEMP.RSV - SOLUCAO
TIPO DE MODELO DE REDE...: M/M/1
SOLUCAO.....: Analitica
MEDIDAS DE DESEMPENHO:
  Tempo medio de espera em fila.....: 0.500
  Tempo medio de resposta do sistema...: 1.000
  Numero medio de fregueses no sistema: 1.000
  Fator de utilizacao.....: 0.500
Enter:Continua
```

Fig. 2.10 - Resultados da Avaliacao do Sistema M/M/1

A interface do SAVAD bem como sua operacionalização estão apresentadas de maneira bastante detalhada e abrangente no apêndice E, que foi organizado em forma de Manual do Usuário. No manual é feita uma descrição de todas as funções disponíveis, encadeamento das tarefas e dos formatos dos itens de cada elemento do SAVAD. Além disso, o Manual do Usuário fornece uma descrição conceitual dos elementos do SAVAD, bem como um exemplo completo de uma especificação de um modelo de redes de filas.

O Manual do Usuário informa a configuração de hardware e software necessários e a forma de instalação e ativação do SAVAD. Esse Manual foi estruturado de forma a ser auto-suficiente, isto é, o usuário normalmente não necessitará de outra fonte de informação para a operação do SAVAD.

### 2.3.2 Núcleo do SAVAD

O núcleo do SAVAD procura determinar qual o método de solução mais adequado para o modelo de rede de filas especificado pelo usuário, seja analítico ou numérico. Para isto dispõe de um motor de inferência e uma base de conhecimento.

Se a técnica de Simulação Digital não tiver sido solicitada pelo usuário, o motor de inferência procura solucionar o modelo analiticamente. Para isto ele tenta identificar o modelo com base nas informações disponíveis em sua base de conhecimento. Não sendo possível, o SAVAD submete o modelo à Simulação Digital.

Quando a Simulação Digital é a técnica escolhida para solucionar um modelo de redes de filas, parâmetros de controle da simulação são solicitados, conforme apresentados no Manual do Usuário (Apêndice E).

Após a identificação do modelo de rede, o motor de inferência acionará a biblioteca numérica do módulo solução para o cálculo correspondente.

Após o modelo ser solucionado, o motor de inferência repassa os resultados obtidos para a Interface para que sejam apresentados ao usuário.

O SAVAD foi projetado e implementado como um Sistema Baseado em Conhecimento (SBC) de maneira que a base de conhecimento e a biblioteca numérica possam ser ampliadas de forma bastante simples.

Os modelos que presentemente fazem parte da base de conhecimento do SAVAD e que podem ser solucionados analiticamente são as redes Markovianas abaixo:

1. M/M/1
2. M/M/1 com chegadas desencorajadas
3. M/M/  $\infty$
4. M/M/m
5. M/M/1/K
6. M/M/m/m
7. M/M/1//M
8. M/M/ $\infty$ //M
9. M/M/m/K/M
10. Redes abertas (solucionadas pelo Teorema de Jackson)
11. Redes fechadas (solucionadas pela Análise do Valor Médio - AVM)

A identificação dos sistemas de filas segue a notação de Kendall:

A/B/m/K/M

Onde,

A, B - FDP (Função de Distribuição de Probabilidade) dos intervalos entre chegadas de clientes sucessivos e de serviço.

O SAVAD identifica as seguintes FDP's:

- M - Exponencial
- G - Geral
- D - Processos determinísticos

m - Número de servidores

K - Comprimento da fila. Se não for especificado, consider sem limite.

M - Tamanho da população de clientes do sistema. Se não for especificado, considerar sem limite.

Os modelos de rede 1 a 9 serão designadas neste trabalho como modelos "M/M/.". A identificação e solução desses modelos foi efetuada conforme [Kleinrock, 1975].

Os modelos de rede 10 e 11 foram identificados e solucionados conforme descrito em [Moura et al., 1986]. No caso do AVM as estações de serviço devem ter disciplina de escalonamento FCFS e todas as cadeias visitando as estações de serviço devem ter a Função de Distribuição de Probabilidade exponencial com a mesma média.

Nos modelos de redes de filas que podem ser solucionados pela Análise do Valor Médio, o motor de inferência inicialmente analisa a complexidade computacional da implementação dessa Análise. Para isso o SAVAD verifica qual a quantidade de memória principal disponível no microcomputador. Caso a solução do modelo usando a AVM exija a utilização de memória em quantidade superior à disponível, o modelo será submetido à Simulação Digital.

Caso o modelo não possa ser identificado dentre os tipos relacionados, este será submetido à Simulação Digital. Neste caso o SAVAD solicita ao usuário o fornecimento dos parâmetros para a simulação.

### 2.3.3 Solução

O módulo solução é composto por:

- **Biblioteca Numérica**

A biblioteca numérica possui um conjunto de soluções analíticas para modelos de redes de filas markovianas.

- **Simulador Digital**

O simulador digital está montado externamente ao SAVAD, sendo por este acionado, caso não exista solução analítica para o modelo de redes. O usuário pode, opcionalmente submeter o modelo à Simulação Digital, mesmo que exista solução analítica para o modelo.

Após a resolução do modelo de redes de filas, os resultados com as medidas de desempenho são enviados para a interface para serem apresentados ao usuário.

## 2.4 Medidas de Desempenho

As medidas de desempenho, conforme os modelos de redes de filas submetidos ao SAVAD são:

### a) Modelos de rede do tipo M/M/1:

- Tempo médio de espera em fila
- Tempo médio de resposta do sistema (tempo médio de espera em fila mais tempo médio de serviço)
- Número médio de clientes no sistema
- Fator de utilização do(s) servidor(es)

### b) Modelos solucionados usando o Teorema de Jackson:

1) Para cada estação de serviço:

- Tempo médio de espera
- Comprimento médio da fila
- Fator de utilização do(s) servidor(es)

2) Tempo médio de resposta do sistema (tempo médio de espera em fila mais tempo médio de serviço)

### c) Modelos solucionados usando a Análise do Valor Médio:

1) Para cada cadeia (rota) da rede:

- Vazão média
- Tempo médio de circulação (somatório dos tempos médios de resposta das estações de serviço)

2) Para cada estação de serviço da rede:

- Comprimento médio da fila

### d) Modelos solucionados usando Simulação Digital:

O programa de Simulação Digital fornece medidas de desempenho conforme o modelo de redes de filas submetido ao SAVAD. Maiores detalhes encontram-se em [Cabral et al., 1993].

## CAPÍTULO 3

### PROJETO ORIENTADO A OBJETOS DO SAVAD

#### 3.1 Abordagem Orientada a Objetos

A abordagem orientada a objetos permite o desenvolvimento de sistemas de maneira natural e modular. Os elementos do problema são representados por suas propriedades, comportamentos e comunicação entre si.

No projeto orientado a objetos, é feita uma representação do problema existente para o domínio da solução.

Esta abordagem permite que o problema seja representado por um conjunto de objetos que caracterizam o seu comportamento no mundo real. Os objetos constituem-se em cápsulas com as estruturas de dados e as rotinas a elas associadas [Korson e McGregor, 1990]. Cada objeto representa uma abstração de uma parte do problema.

As classes correspondem à descrição de um conjunto de propriedades e atributos que se deseja representar. Um mesmo problema pode ser representado por classes diversas, de acordo com o enfoque desejado. Classes não têm existência real; elas são categorias de objetos com o mesmo comportamento e com as mesmas propriedades. Objetos são instâncias das classes, possuindo existência real [Gibs et al., 1990].

O processamento consiste na troca de mensagens entre os objetos. Os objetos agem uns sobre os outros através do envio de mensagens, solicitando informações, alterando valores ou executando operações. Ao receber uma mensagem, o objeto ativa uma das funções (métodos) que descreve o comportamento deste objeto associado a esta mensagem [Wirfs-Brock et al., 1990].

Quando se emprega a abordagem orientada a objetos, procura-se identificar:

- a) Objetos (e suas classes)
- b) Mensagens
- c) Sequências mensagens-objetos

As propriedades básicas que uma linguagem deve possuir para ser orientada a objetos, são:

- a) Abstração
- b) Encapsulamento
- c) Herança
- d) Polimorfismo

A **abstração de dados** permite que apenas as características principais sejam apresentadas ao usuário. Elementos secundários para o propósito desejado e níveis de complexidade desnecessários são desconsiderados. Um mesmo problema pode se apresentar de formas diversas do acordo com as necessidades.

O **encapsulamento** possibilita isolar a solução do problema de sua implementação. Apenas a interface do objeto fica disponível para usuários externos e os detalhes de implementação não são visíveis. Deste modo nenhum elemento externo tem acesso aos dados ou funções existentes no interior do objeto.

O **mecanismo de herança** permite que as características de uma classe possam ser usadas por uma outra. Neste caso a classe herdeira (sub-classe) pode dispor do código da classe originária (superclasse). A classe herdeira pode incluir novos métodos ou alterar métodos já existentes. Desta forma, a partir de classes mais genéricas, são construídas classes mais específicas.

O **polimorfismo** permite que operações (ou funções) com implementações diversas possam ser definidas com o mesmo nome, sendo reconhecidas pelo contexto. O polimorfismo torna os procedimentos mais simples e naturais pois operações similares têm denominações comuns como na vida real embora sejam implementadas de maneiras distintas (ex.: operações aritméticas com tipos numéricos diversos).

A **amarração dinâmica** possibilita a ativação do procedimento adequado apenas quando da execução do programa. A amarração dinâmica normalmente é utilizada para implementar o polimorfismo.

Classes desenvolvidas e já testadas podem ser reutilizadas. Novas classes podem ser derivadas destas classes com a implementação de novos métodos ou a alteração dos já existentes. Um conjunto de classes elementares pode ser desenvolvido e testado, podendo ser utilizado em projetos posteriores de forma similar à construção civil usando módulos pré-fabricados. O desenvolvimento de projetos se torna mais seguro e produtivo desde que os erros de programas podem ser mais facilmente isolados, bastando analisar os códigos das classes derivadas [Jordan, 1990].

As linguagens orientadas a objetos mais comuns são SMALLTALK [Jonathan, 1988], C++ [Stroustrup, 1986], Object Pascal [Szmit, 1993], etc. A linguagem C++ corresponde a uma extensão da linguagem C. Foi desenvolvida visando fornecer as facilidades da abordagem orientada a objetos mantendo a compatibilidade e eficiência de código executável da linguagem C bem como aprimorá-la em alguns aspectos.

### 3.2 Documentação do Projeto

O projeto orientado a objetos do SAVAD corresponde a uma representação do problema para sua implementação.

O presente projeto é composto por um número considerável de classes que se relacionam entre si. A representação de projetos de sistemas desenvolvidos utilizando a programação orientada a objetos através das metodologias estruturadas convencionais (YORDON, GANE-SARSON, etc.) não consegue captar e representar aspectos essenciais do projeto. As características básicas da abordagem orientada a objetos (polimorfismo, herança entre classes, encapsulamento e abstração de dados), bem como a instanciação de objetos e as mensagens trocadas entre eles não conseguem ser representadas adequadamente.

Reconhecendo este problema, diversos autores apresentam propostas de representação para a abordagem orientada a objetos, entre eles [Booch, 1991], [Pressman, 1987], [Wasserman et al., 1990], [Wasserman et al., 1991], [Korson & McGregor, 1990], [Henderson-Sellers & Edwards, 1990], [Wirfs-Brock et al., 1990], [Gibbs et al., 1990], [Coad & Yourdon, 1992], etc. Por ter tido uma disseminação mais recente, as técnicas de representação do projeto orientado a objetos ainda não foram consolidadas, com os autores apresentando notações ainda divergentes entre si e que eventualmente não representam com a concisão e presteza necessárias, todos os aspectos do projeto ou da linguagem de programação utilizada.

Na representação do presente projeto foram aproveitadas algumas sugestões de notações descritas pelos autores já relacionados. Em alguns casos foi desenvolvida uma notação própria, baseada em nossa experiência, para melhor descrever aspectos específicos do projeto e da linguagem de programação C++.

A figura 3.1 apresenta a notação empregada para representação de classes/objetos. As classes virtuais são utilizadas pelo C++ para representar classes que são herdadas de forma indireta, através de mais de uma classe, simultaneamente, utilizando o mecanismo de herança múltipla. Desta forma é feita uma única instanciação para o objeto da classe virtual, mesmo que em uma etapa intermediária da cadeia de heranças mais de uma classe o referencie como classe primitiva. Este mecanismo faz com que os métodos/estruturas de dados da classe primitiva obtidas pela classe derivada sejam as mesmas, sem ambiguidades, qualquer que seja a rota de herança escolhida.

A figura 3.2 apresenta a instanciação de objetos conforme [Booch, 1991] e [Wasserman et al., 1990].

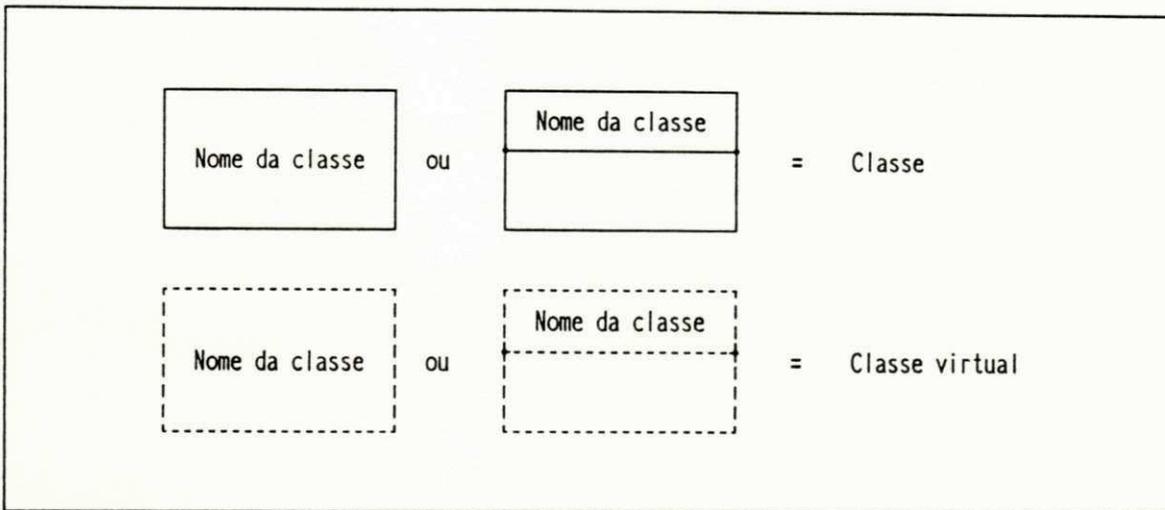


Fig. 3.1 - Notacao para Representacao de Classes

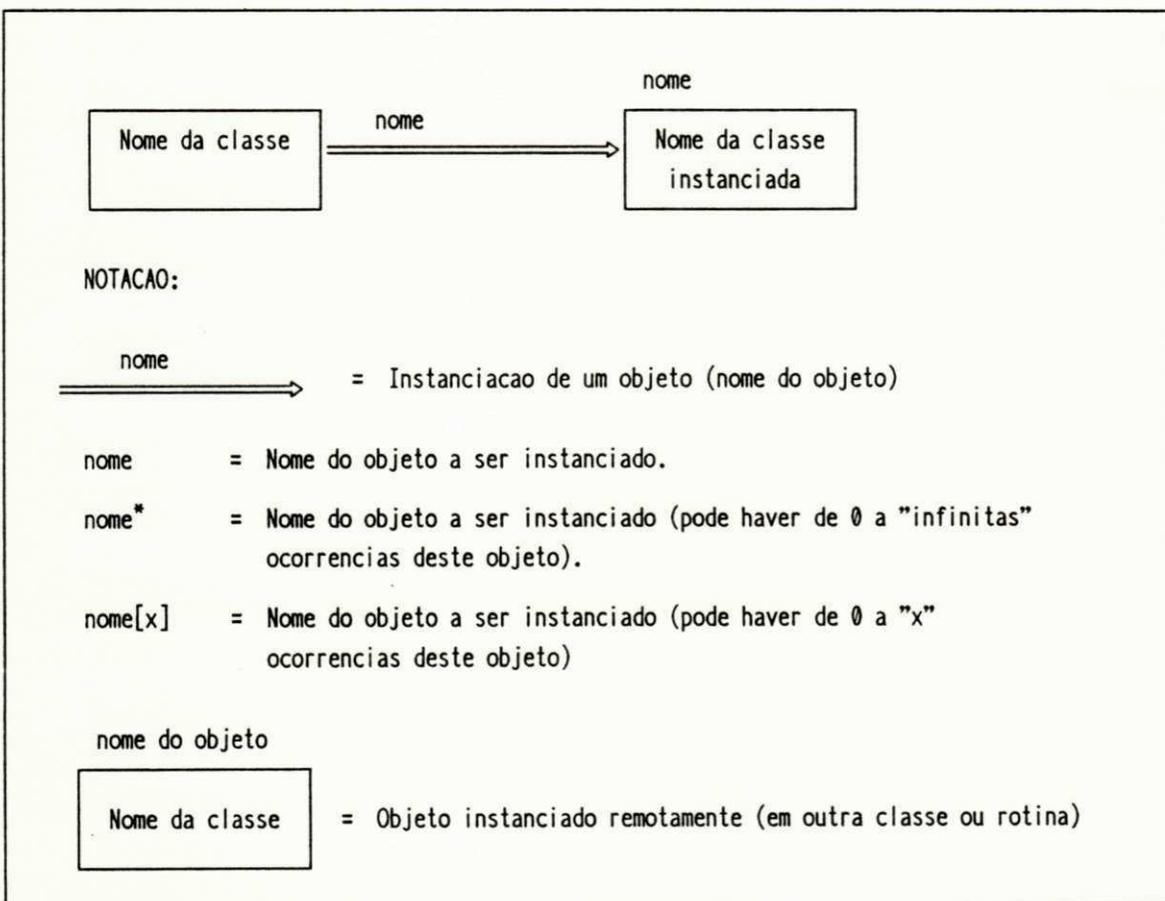


Fig. 3.2 - Instanciacao de Objetos

O relacionamento entre classes "amigas" (friends), existente em C++ é mostrado na figura 3.3, conforme [Wasserman et al., 1991]. O conceito de classe "friend" na linguagem C++ não segue os princípios da orientação a objetos na medida em que permite que uma classe designada como "friend" em uma outra classe tenha acesso a todas as estruturas de dados e métodos da classe, sem qualquer restrição. Desta forma as estruturas de dados e os métodos da classe podem ser acionados "remotamente", sem o envio de mensagens, pela classe "amiga". Por outro lado isto aumenta a eficiência de código do programa ao evitar o tráfego muito intenso de mensagens e que volumes muito grandes de dados sejam movimentados e duplicados. O mecanismo de classes "friend" deve ser utilizado com muito cuidado pois diminui a legibilidade do programa e aumenta a possibilidade de erros.

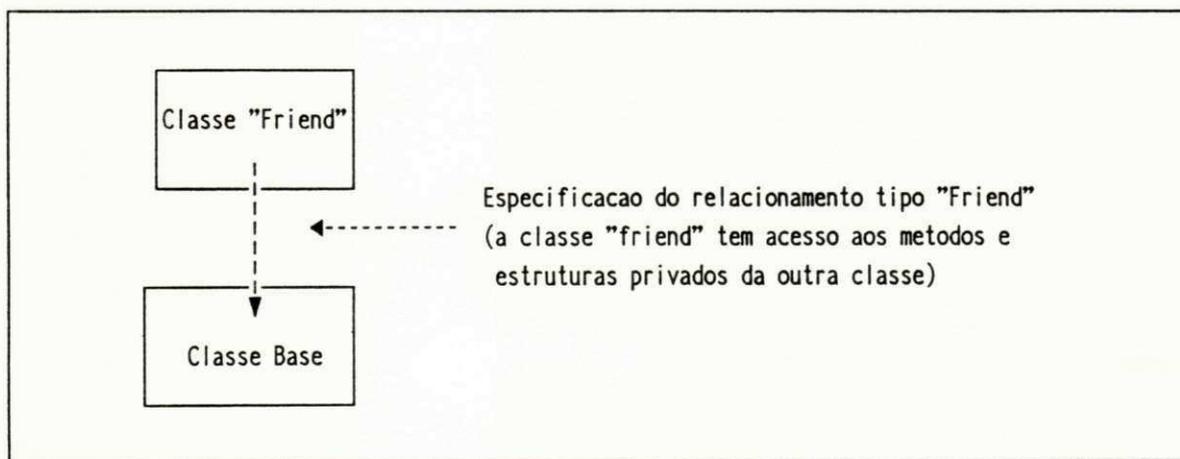


Fig. 3.3 - Notacao de Relacionamento entre Classes "Amigas" ("Friend")

A descrição do projeto orientado a objetos do SAVAD foi feita através das seguintes representações:

- 1) Diagrama geral de herança de classes.
- 2) Diagrama geral cliente-servidor.
- 3) Protocolo das classes.
- 4) Diagrama de Booch das classes.
- 5) Diagrama herança/cliente-servidor.

As representações 1 e 2 fornecem uma visão mais abrangente do projeto. Os diagramas gerais de classes do SAVAD encontram-se nos apêndices A e B. As representações 3, 4 e 5 formam um "dicionário de classes" que descreve detalhadamente as classes que compõem o projeto. Tal dicionário permite identificar para cada classe sua interface, seus relacionamentos com as demais classes, e os impactos decorrentes de alterações no projeto. O "dicionário de classes" do SAVAD encontra-se em [Souto et al., 1993].

### 3.2.1 Diagrama Geral de Herança de Classes

Com o mecanismo de herança, pode-se criar classes mais específicas a partir de classes mais genéricas. Desta forma as classes podem ser definidas de forma mais simples e segura.

As classes genéricas, também chamadas de superclasses, são definidas com as funcionalidades mínimas e comuns. A partir destas classes são derivadas outras classes-herdeiras que lhes acrescentam funcionalidade, sucessivamente.

O diagrama procura mostrar de forma global todos os relacionamentos em termos de herança existentes entre as classes do sistema. Para sua representação foi empregada a notação definida por [Booch, 1991] e [Wasserman et al., 1991] (figura 3.4).

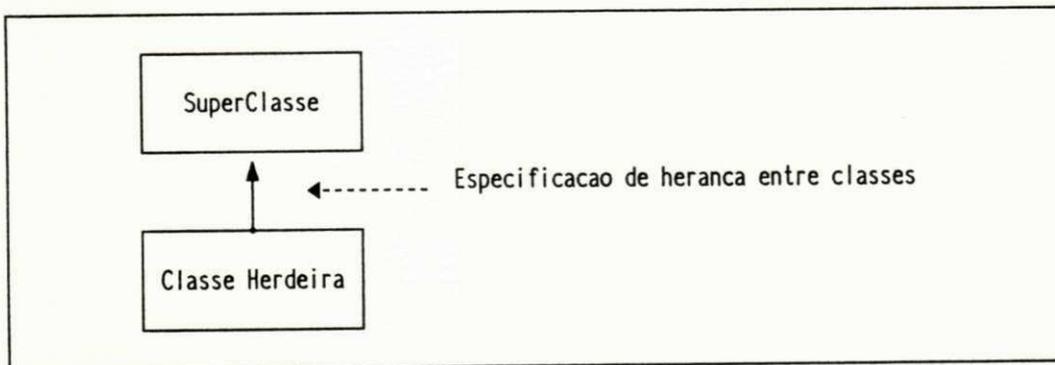


Fig. 3.4 - Notacao para Heranca entre Classes

A figura 3.5 apresenta um exemplo de diagrama de herança das classes para as soluções analíticas do SAVAD.

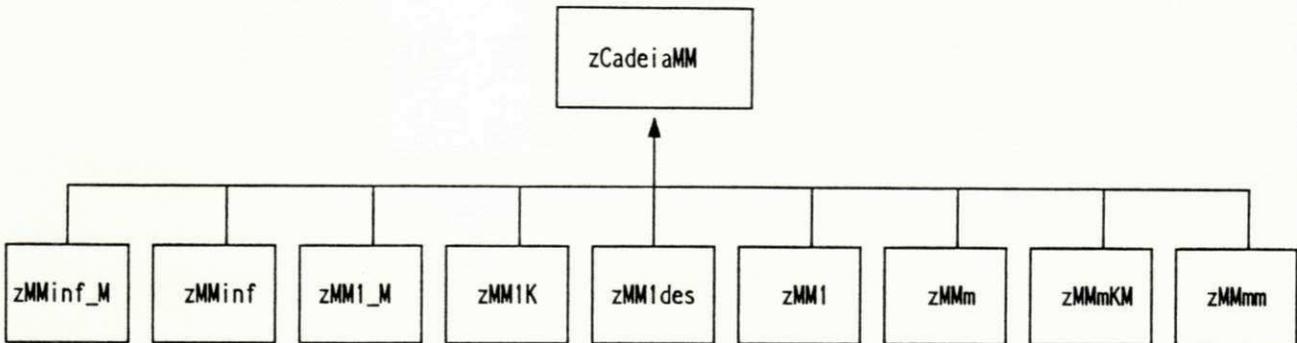


Fig. 3.5 - Diagrama de Herança de Classes para as Soluções Analíticas do SAVAD

### 3.2.2 Diagrama Geral Cliente-Servidor

O diagrama mostra de forma geral e sucinta as mensagens enviadas entre as classes, mostrando a interdependência entre elas. Tem-se o relacionamento cliente-servidor entre as classes.

**Classe cliente:** Solicita os serviços da classe servidora através do envio de mensagens.

**Classe servidora:** Presta serviços para a classe cliente, ao receber uma mensagem.

Uma mesma classe pode ser cliente em algumas situações e servidora em outras.

Na representação dos diagramas cliente-servidor procura-se fornecer uma visão global do sistema. Eventualmente, devido ao número elevado de classes do sistema e o seu grande inter-relacionamento, pode-se simplificar o(s) diagrama(s) de diversas formas:

- a) Utilização de diversos diagramas parciais.
- b) Classes com comportamentos similares podem ser representadas em forma de subsistemas. Estes subsistemas podem ser expandidos posteriormente, em outros diagramas.
- c) Uma mesma classe com alto nível de acoplamento com as demais pode ser representada mais de uma vez no diagrama para reduzir o entrelaçamento de classes.

A figura 3.6 mostra a notação empregada para a representação conforme definido por [Presman, 1987]. Foram feitas extensões na notação para simplificar a representação e permitir uma visualização das mensagens que são enviadas entre os objetos. O apêndice B apresenta os diagramas gerais cliente-servidor do SAVAD.

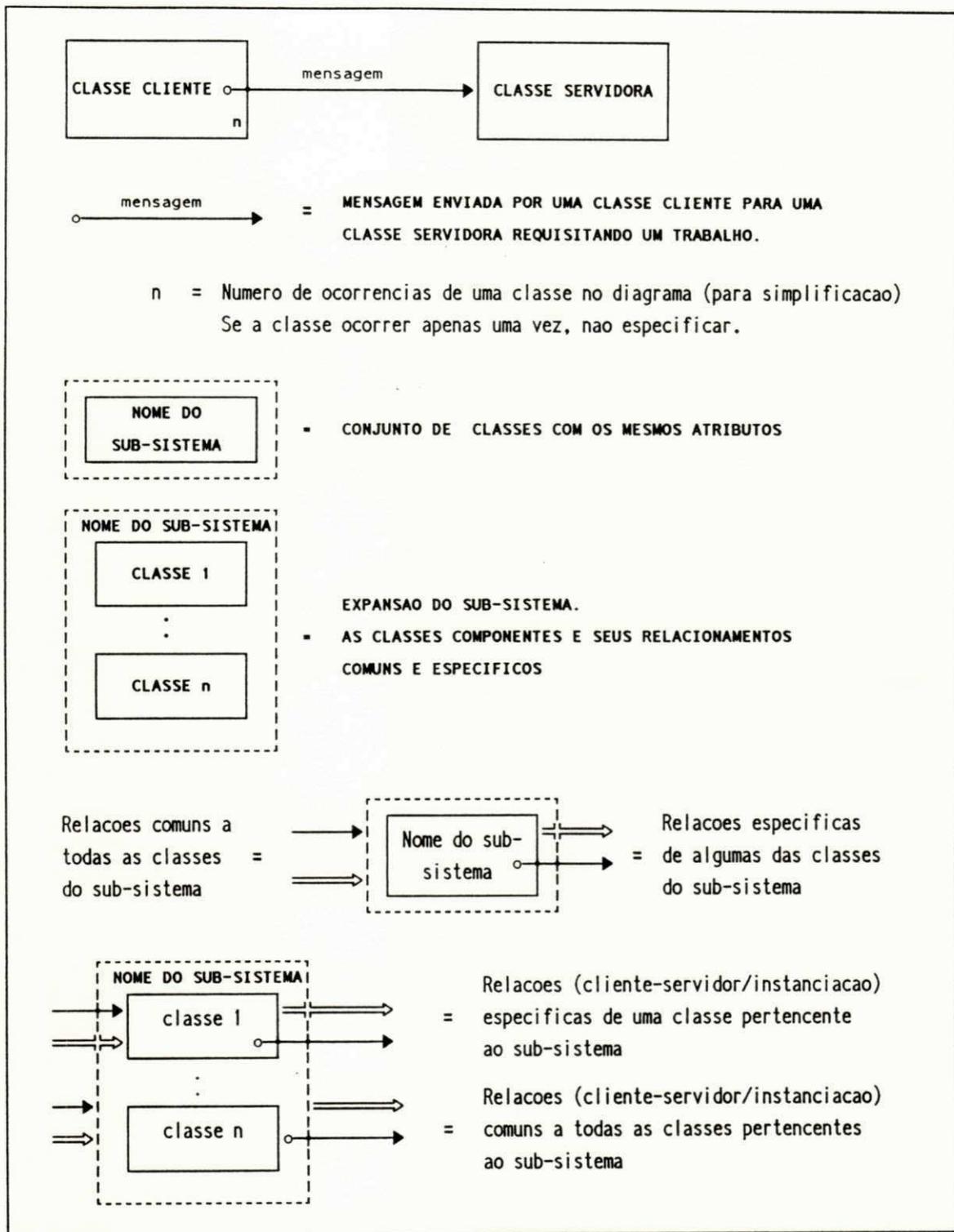


Fig. 3.6 - Notacao para os Diagramas Cliente-Servidor

A figura 3.7 apresenta um exemplo de diagrama cliente-servidor utilizado no SAVAD.

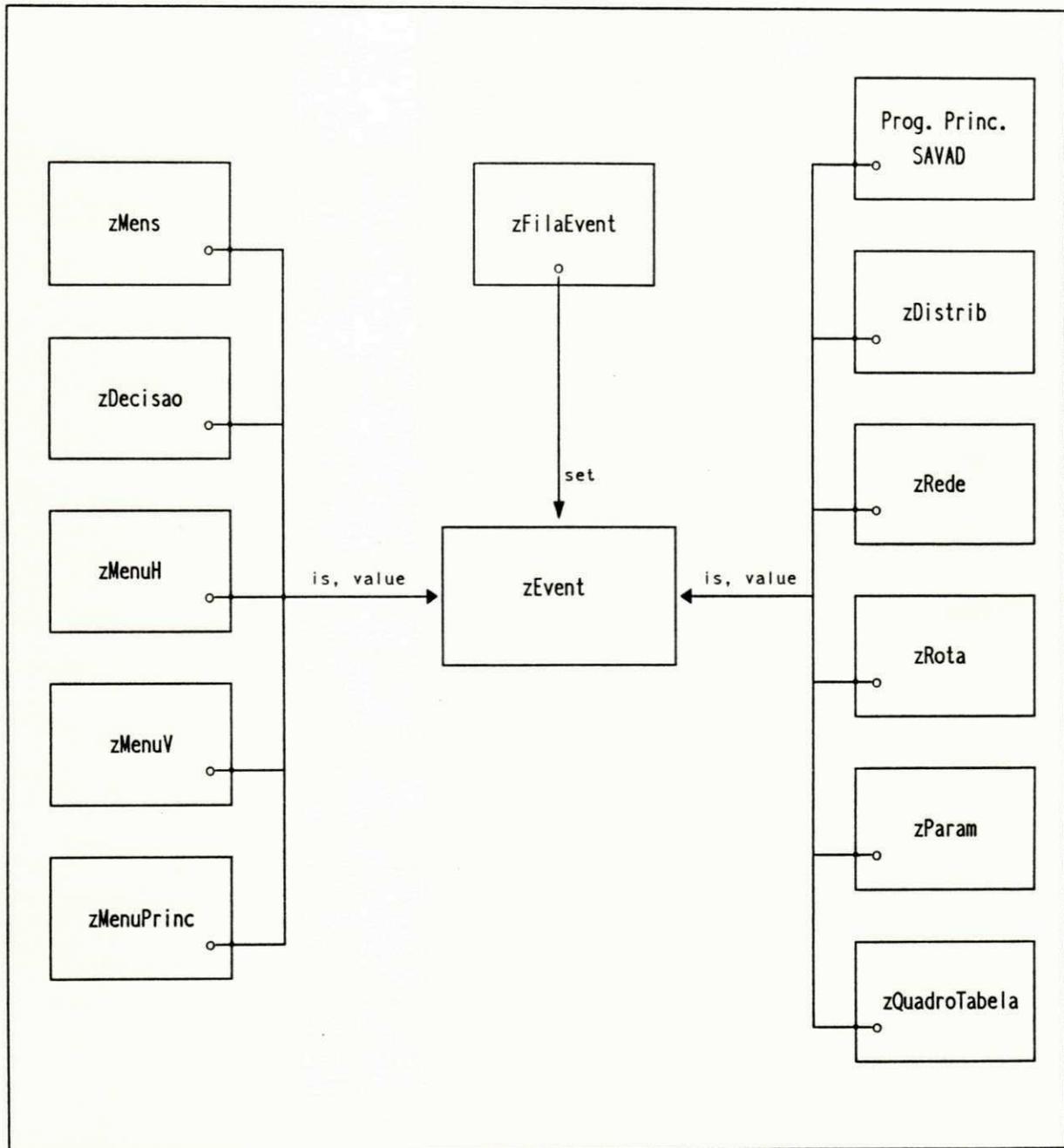


Fig. 3.7 - Diagrama Cliente-Servidor do Mecanismo de Controle de Eventos (teclado, mouse, temporizador, etc.) pelo SAVAD

### 3.2.3 Protocolos de Classes

É feita uma descrição sucinta e formal de cada uma das classes do sistema, de acordo com a notação descrita por [Presman, 1987].

Formato:

**1. Classe:** Nome da classe

**2. Descrição:** Descrição da classe.

**3. Superclasses:** Classes das quais esta é herdeira direta.

**4. Variáveis de classe:** Relação e descrição das variáveis. As variáveis de classes são criadas uma única vez quando o primeiro objeto desta classe é instanciado, podendo ser utilizadas por todos os demais objetos instanciados.

Podem ser usadas para representar constantes comuns ou variáveis que devam ser acessadas por todos os objetos.

**5. Variáveis de instância:** Relação e descrição das variáveis.

São variáveis específicas de cada objeto instanciado.

**6. Métodos:**

**6.1 - Públicos:** Relação e descrição dos métodos acessíveis a todas as demais classes.

**6.2 - Protegidos:** Relação e descrição dos métodos acessíveis somente às classes herdeiras.

**6.3 - Privados:** Relação e descrição dos métodos internos da classe.

São feitas extensões para uma descrição mais apurada da classe com respeito às características da linguagem de programação utilizada (C++). São feitas as seguintes especificações adicionais:

- Classes virtuais.
- Métodos virtuais e protegidos.
- Herança de classe pública, protegida e privada.
- Estruturas de dados protegidas.
- Classes "friend" que acessam os métodos e variáveis privados de uma outra classe.

Apresentamos a seguir um exemplo de protocolo da classe zMens usada no SAVAD.

1. **CLASSE:** *zMens*

2. **DESCRIÇÃO:** *A classe se propõe a exibir uma janela com uma mensagem para o usuário. A mensagem será exibida até que o usuário acione uma tecla qualquer.*

3. **SUPERCLASSES:** *zAreaTela (Privada)*  
*zObjetoInterativo (Pública)*

4. **VARIÁVEIS DE CLASSE:**

*maxw:* *Largura máxima de uma linha da resposta.*

5. **VARIÁVEIS DE INSTÂNCIA:** *Não há.*

6. **MÉTODOS**

6.1 - **PÚBLICOS:**

*Construtor:* *Obtém os dados básicos de operação. Salva o conteúdo anterior da tela. Apresenta janela com a mensagem para o usuário e fica em estado de espera.*

*wait:* *Aguarda o acionamento de uma tecla pelo usuário.*

*error:* *Informa da ocorrência de erro quando da criação da classe zAreaTela.*

*process\_event:* *Verifica se houve acionamento de alguma tecla. Despreza os demais eventos.*

6.2 - **PRIVADOS**

*howhigh:* *Fornece a altura da janela a ser exibida.*

*howwide:* *Fornece a largura da janela a ser exibida.*

### 3.2.4 Diagrama de BOOCH das Classes

Tais diagramas procuram apresentar o funcionamento geral de cada classe e sua interface.

Os diagramas apresentam para cada classe os métodos, as estruturas de dados, bem como os parâmetros de entrada/saída dos métodos públicos e protegidos.

Os métodos e estruturas de dados são apresentados como públicos, protegidos e privados.

Foi usada a notação conforme descrita em [Booch, 1991] e adaptada por [Wasserman, 1991]. Foram feitas extensões nesta notação para descrever os métodos e estruturas de dados protegidas e virtuais bem como representar imediatamente as super-classes (herança) e classes "amigas" (friend) conforme mostra a figura 3.8.

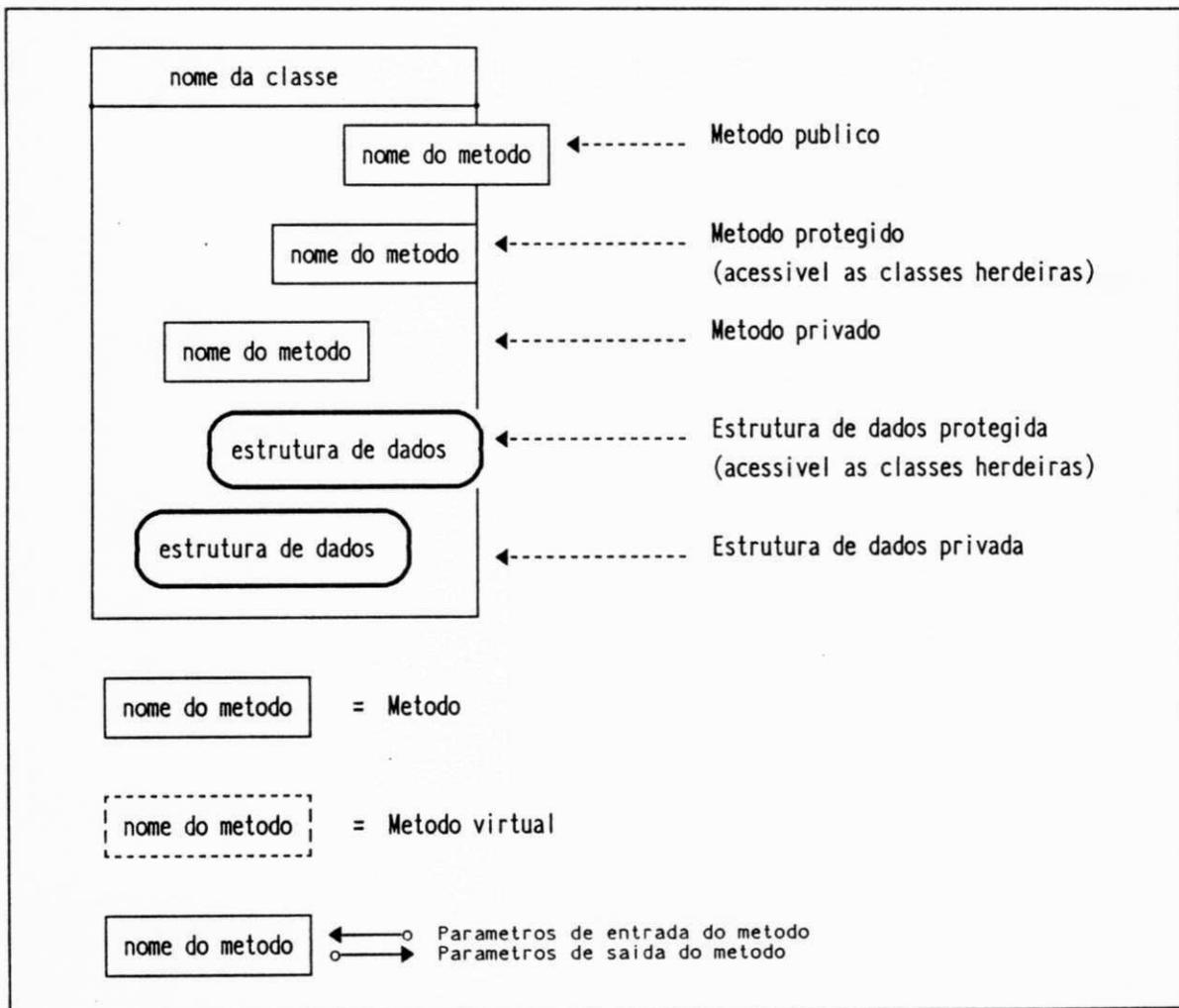


Fig. 3.8 - Notação para os Diagramas de BOOCH das Classes

A figura 3.9 apresenta o diagrama de Booch da classe zSorvedouros do SAVAD.

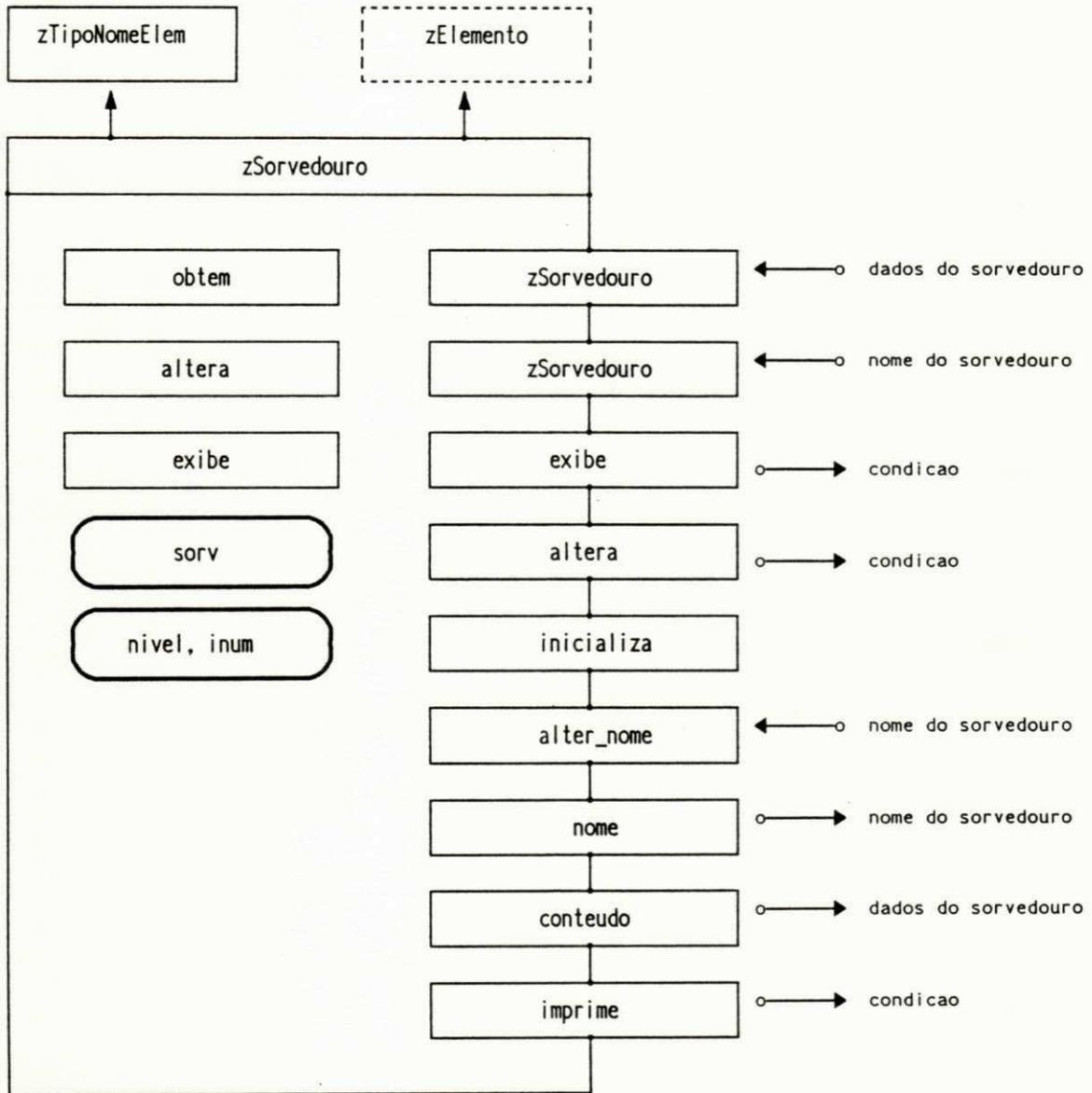


Fig. 3.9 - Diagrama de BOOCH da Classe zSorvedouros

### 3.2.5 Diagrama Herança/Cliente-Servidor

O diagrama mostra, para cada classe, quais as demais classes que se relacionam com ela, isto é, mostra todas as classes servidoras desta classe (fornecem serviços para a classe), com as mensagens enviadas bem como as instanciações de objetos efetuadas. Este diagrama também apresenta as classes clientes (usam os serviços da classe) com as respectivas mensagens e instanciações.

São representadas as superclasses, classes herdeiras e as classes "amigas" da classe em análise.

A partir dos diagramas podem-se ver todos os relacionamentos para as diversas classes do sistema, verificando-se facilmente quais as demais classes afetadas por uma alteração em uma das classes.

Os diagramas comportam-se como listas duplamente encadeadas, desde que as classes ora são representadas como classe base (em seu próprio diagrama), ora como classe cliente/servidora (nos diagramas das outras classes). Esta representação, por outro lado, simplifica muito a manipulação e a consulta dos diagramas.

Para a representação dos diagramas herança/cliente-servidor foi usada uma combinação da notação utilizada nos diagramas cliente-servidor e de herança conforme a figura 3.10.

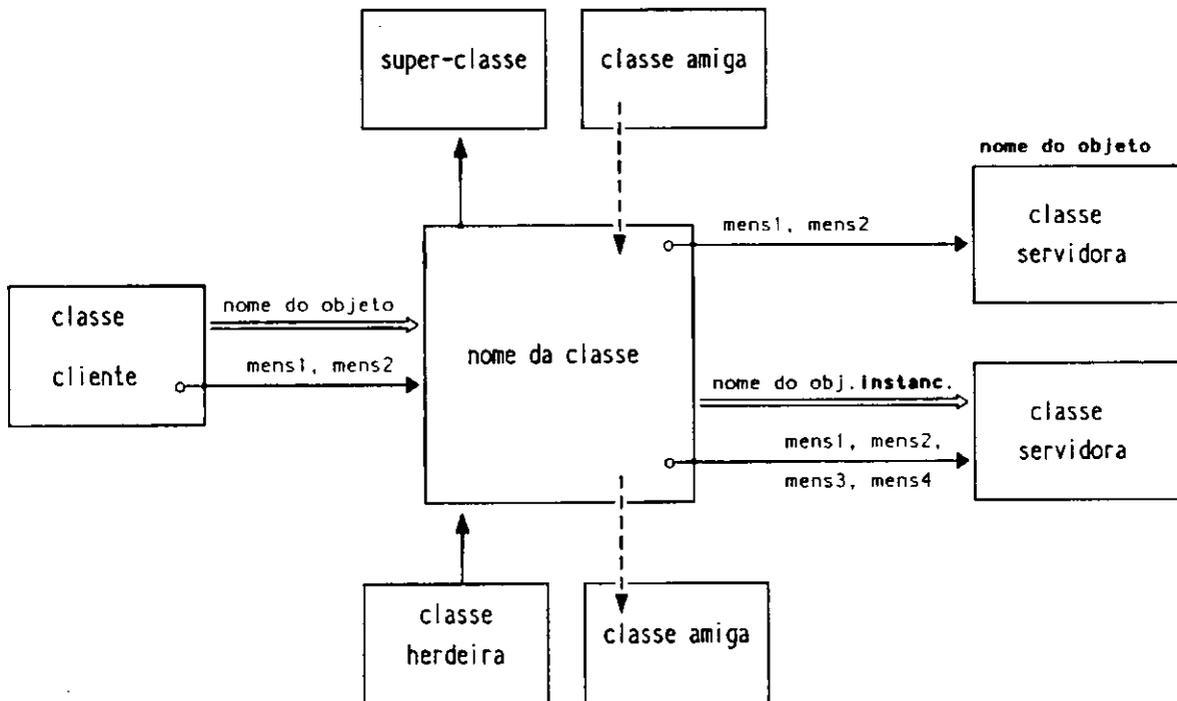


Fig. 3.10 - Notação dos Diagramas Herança/Cliente-Servidor das Classes

A fig. 3.11 apresenta o diagrama herança/cliente-servidor da classe *zDecisao* do SAVAD.

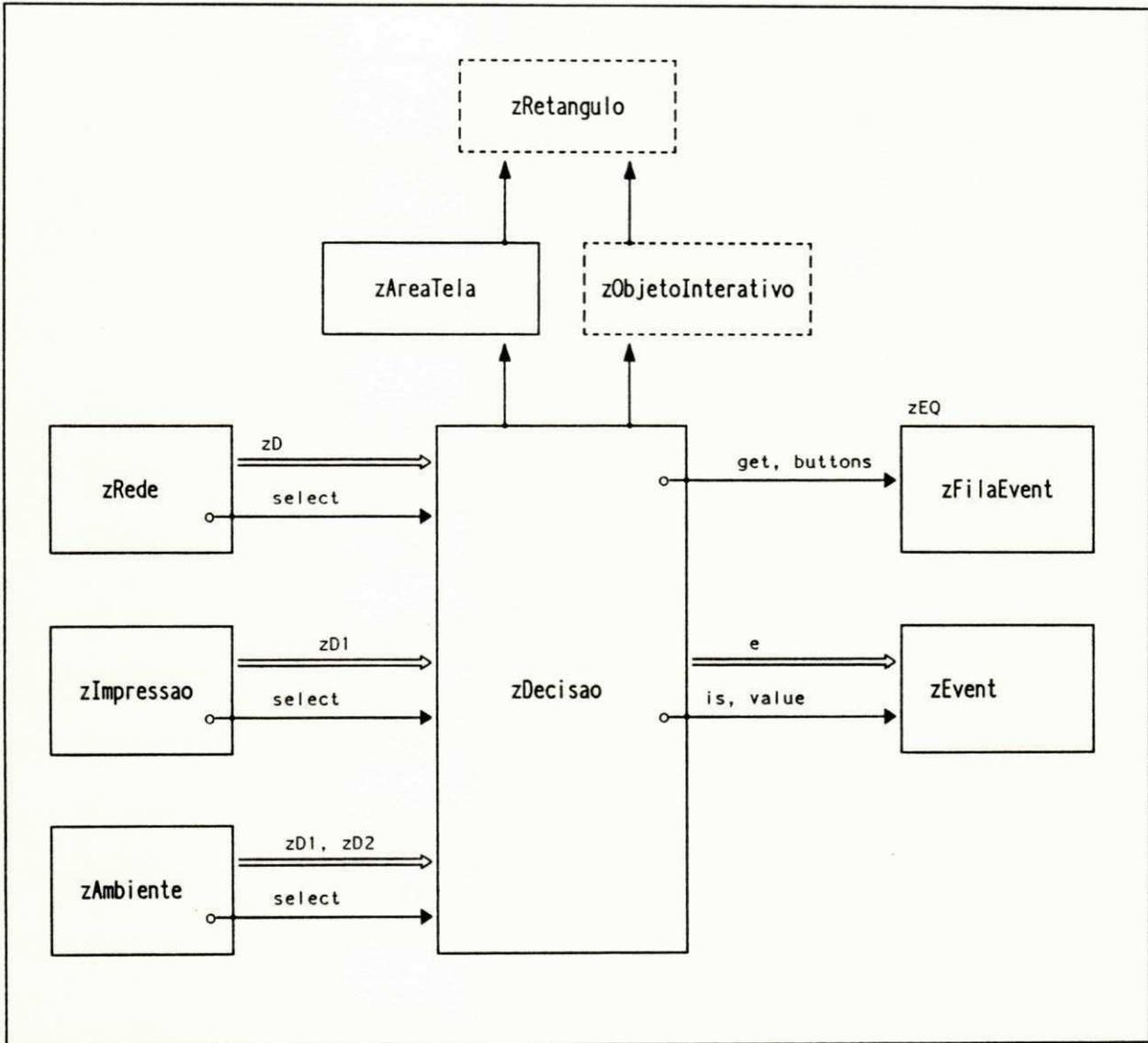


Fig. 3.11 - Diagrama Heranca/Cliente-Servidor da Classe *zDecisao*

### 3.3 Projeto do SAVAD

No projeto orientado a objetos do SAVAD procurou-se seguir as etapas descritas em [Presman, 1987], que são:

1. Identificação das abstrações de dados do sistema.
2. Identificação das operações de cada abstração de dados.
3. Identificação dos atributos de cada abstração de dados.
4. Definição da comunicação entre os objetos.
5. Utilização do mecanismo de herança.
6. Implementação e teste do projeto.

O projeto foi desenvolvido utilizando uma metodologia incremental em que as etapas 1 a 6 foram executadas sucessivamente, incorporando novas funções e facilidades ao projeto ao mesmo tempo que outras funções já desenvolvidas eram testadas.

Vamos descrever cada uma das etapas:

#### 3.3.1 Identificação das Abstrações de Dados

Nesta etapa são identificadas as classes de objetos. No SAVAD foram definidas as seguintes classes:

##### a) Gerenciamento geral do SAVAD

- zAmbiente

##### b) Elementos do SAVAD

- zFonte
- zServidor
- zSorvedouro
- zRota
- zClasse
- zEscalonador
- zMultiplicação
- zFusão
- zRede
- zSincronização
- zDistrib
- zTextRota
- zEditorRota
- zTipoNomeElem
- zElemento

## b) Soluções analíticas

- zAvm
- zCadeiaMM
- zMM1
- zMM1des
- zMM1K
- zMM1\_M
- zMMinf
- zMMinf\_M
- zMMm
- zMMmKM
- zMMmm
- zJackson

## c) Implementação da interface

- zAjuda
- zAreaTela
- zDecisão
- zDirectory
- zEditorData
- zEditorDecimal
- zEditorInt
- zEditorString
- zEvent
- zFilaEvent
- zMens
- zMenuH
- zMenuV
- zMenuPrinc
- zNomeArquivo
- zObjetoInterativo
- zQuadro
- zQuadroArquivo
- zQuadroTabela
- zQuadroTabInt
- zQuadroTabSel
- zRetangulo
- zStringEditor
- zTelaNomeArquivo
- zTelaNomeArqPadrao
- zTelaTrabalho

- zText
- zParam

d) Núcleo do sistema especialista

- Base de conhecimento
  - zBaseConh
  - zVerifMem
- Motor de inferência
  - zMotorInf
- Análise de integridade do modelo
  - zVerifica

e) Impressão dos resultados

- zImpressão
- zImpressãoSav

f) Apoio operacional

- Manipulação de vetores
  - zBitVec
  - zBitVec16
- Manipulação de listas encadeadas, pilhas e arrays
  - zDLCursor
  - zDList
  - zDLNode
  - zGDList
  - zGDLCursor
  - zSList
  - zSListIterator
  - zGStack
  - zSLink
  - zDynamicArray
  - zGDynamicArray

O SAVAD foi implementado utilizando 75 classes. A definição das classes e sua descrição estão no protocolo das classes do sistema, os diagramas de BOOCH e os diagramas de herança/cliente-servidor destas classes estão apresentados em [Souto et al., 1993].

### 3.3.2 Identificação dos Atributos das Abstrações de Dados

Os atributos correspondem às variáveis de classe e de instância. Estes atributos (variáveis) estão descritos nos protocolos das classes do sistema e representados nos diagramas de BOOCH e diagramas de herança/cliente-servidor das classes [Souto et al., 1993].

### 3.3.3 Identificação das Operações das Abstrações de Dados

As operações correspondem aos métodos de classe e de instância. Os métodos são descritos nos protocolos das classes do sistema e representados nos diagramas de BOOCH e diagramas de herança/cliente-servidor das classes [Souto et al., 1993].

### 3.3.4 Definição da Comunicação entre os Objetos

Nesta etapa são identificadas as mensagens que os objetos enviam entre si. Deve-se definir a correspondência entre os métodos e os objetos que invocam estes métodos (mensagens). Nesta etapa é determinado o grau de interdependência entre os diversos objetos. Deve-se buscar um equilíbrio entre a utilização de outros objetos através do envio de mensagens (modularidade) e excessivo acoplamento de objetos que torne o projeto muito complexo e de difícil manutenção.

As mensagens entre os objetos do SAVAD estão representadas nos diagramas gerais cliente-servidor, mostrados no Apêndice B e nos diagramas de herança/cliente-servidor das classes apresentados em [Souto et al., 1993].

Algumas classes de baixo nível operacional (*zEvent*, *zFilaEvent* e *zBitVec16*) mas que possuem alto nível de acoplamento com as demais classes foram retiradas dos diagramas mais gerais para simplificá-los, sendo representadas individualmente.

As figuras 3.12 e 3.13 apresentam exemplos de diagramas cliente-servidor utilizados no SAVAD.

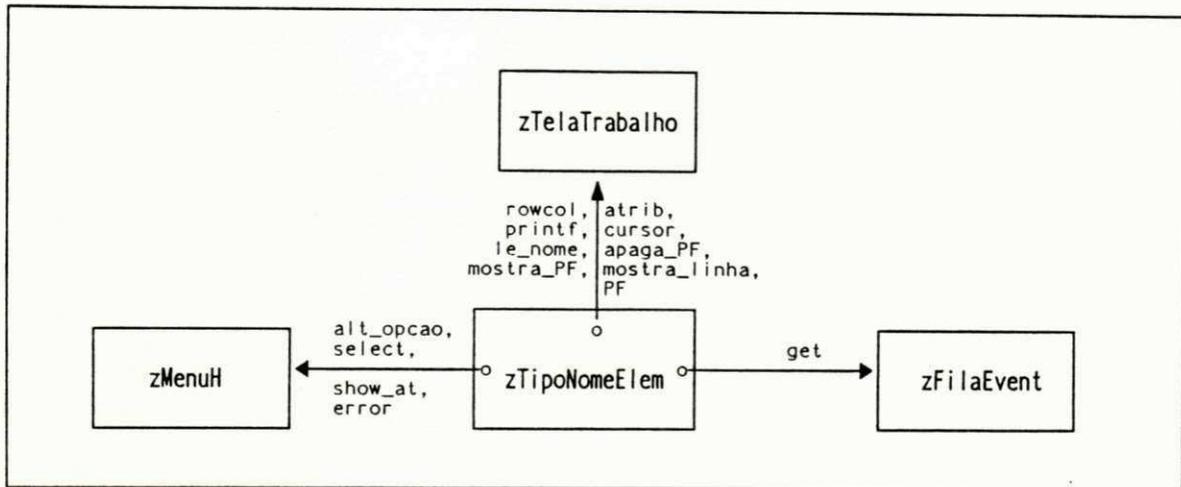


Fig. 3.12 - Diagrama Geral Cliente-Servidor da Classe "Nome e o Tipo do Elemento"

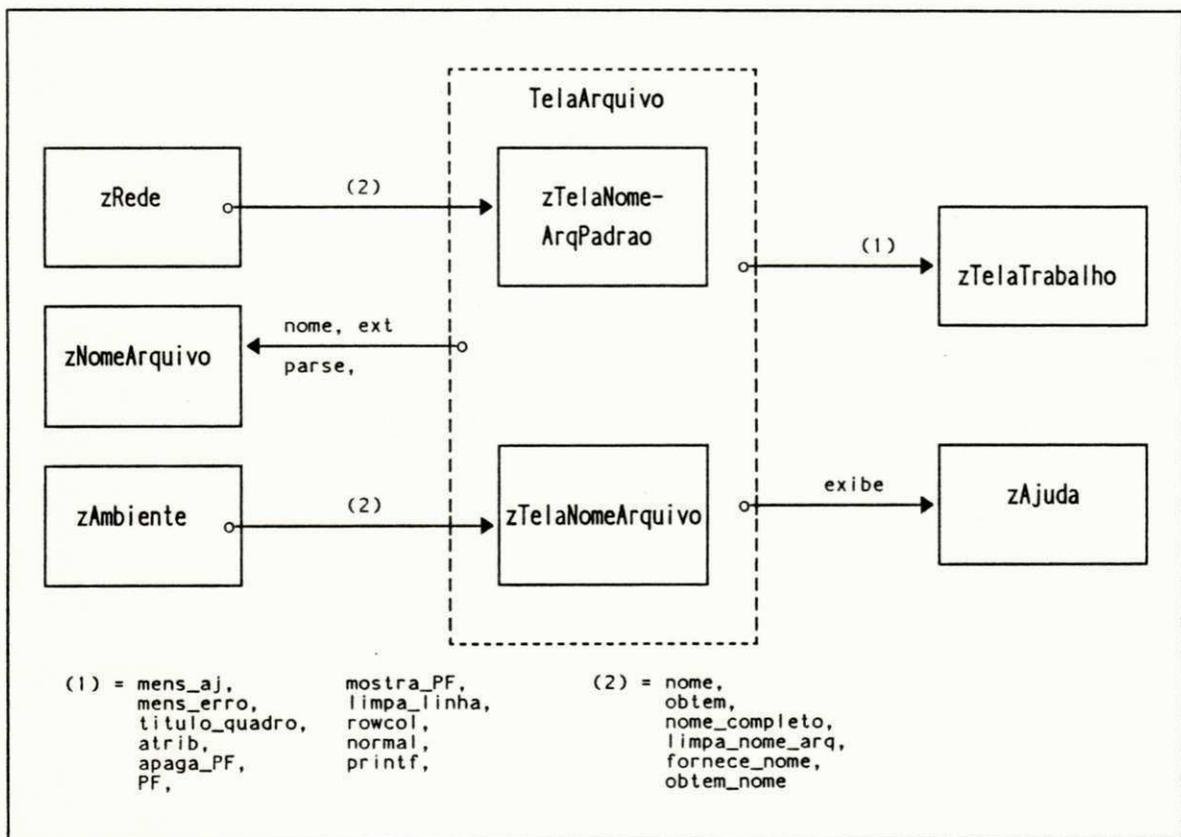


Fig. 3.13 - Diagrama Geral Cliente-Servidor com a Expansão do Sub-Sistema "TelaArquivo"

### 3.3.5 Utilização do Mecanismo de Herança

Procura-se identificar quais as classes e procedimentos mais elementares e genéricas do sistema. O objetivo é a utilização de classes de objetos que possam ser reutilizados através do sistema.

O mecanismo de herança permite a reutilização das classes já existentes. A definição de um conjunto de classes básicas e genéricas constitui os alicerces de um projeto, simplificando o seu desenvolvimento.

O mecanismo de herança de classes do SAVAD está apresentado nos diagramas gerais de herança do sistema mostrado no Apêndice A e diagramas de herança/cliente-servidor das classes em [Souto et al., 1993].

As figuras 3.14 e 3.15 apresentam exemplos da hierarquia de classes do SAVAD.

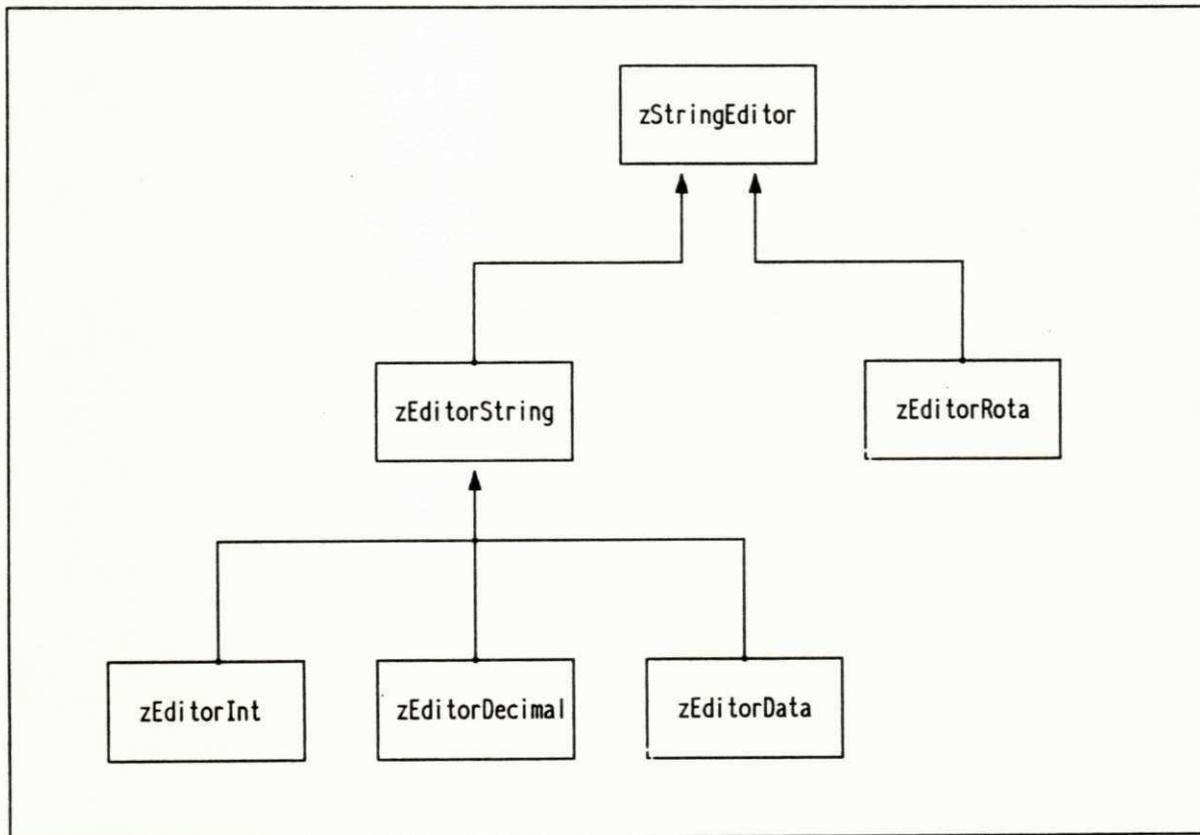


Fig. 3.14 - Diagrama de Herança das Classes de Edição (strings, números, rotas, etc.)

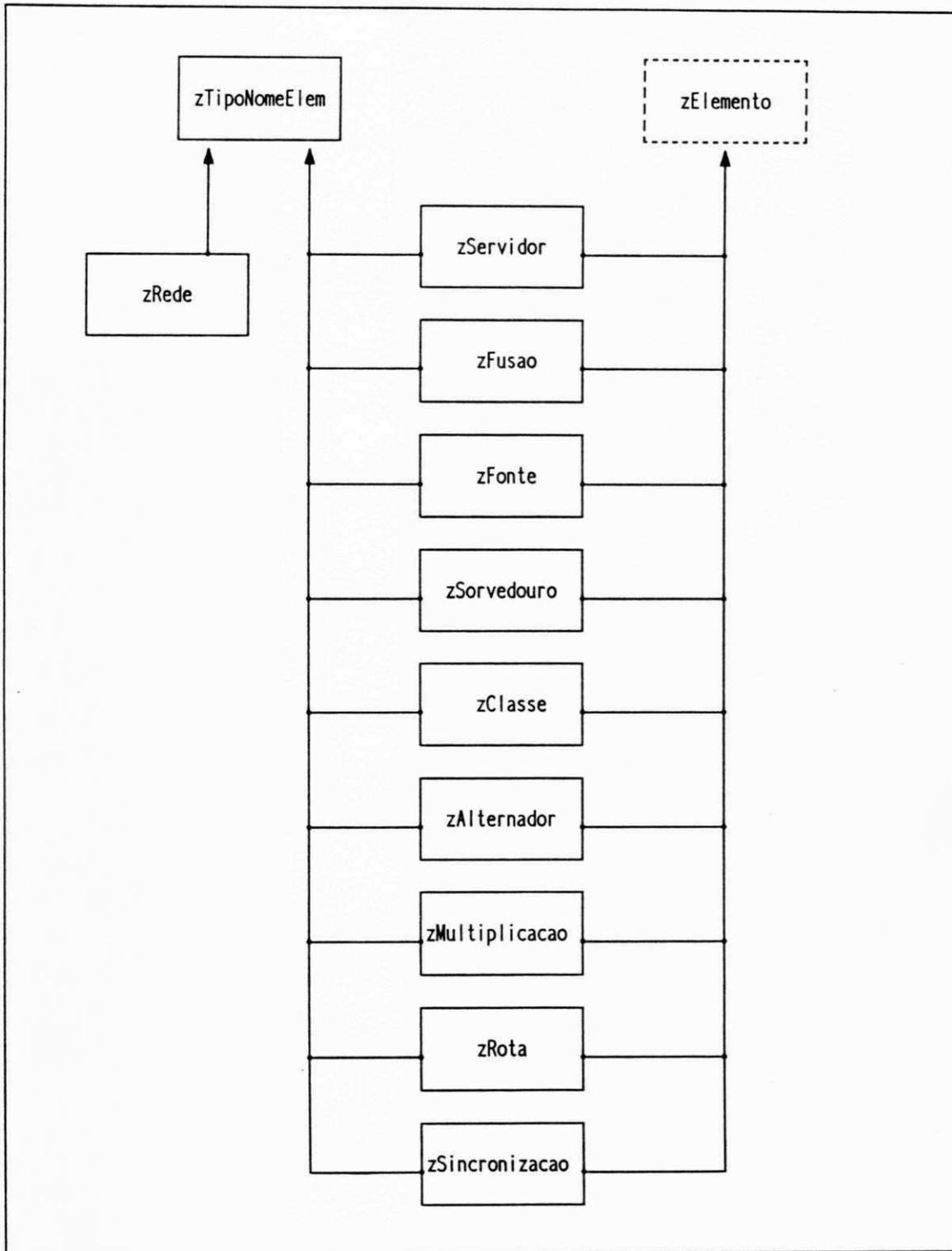


Fig. 3.15 - Diagrama de Heranca das Classes dos Elementos do SAVAD

### 3.3.6 Implementação do Projeto

O projeto foi executado utilizando uma metodologia de desenvolvimento incremental. Após o desenvolvimento de parte das classes, passou-se a refiná-lo voltando-se às etapas anteriores para:

- Acrescentar novas classes.
- Alterar classes já existentes, otimizando e aumentando sua funcionalidade.
- Reorganizar a estrutura de classes.

As etapas do projeto foram executadas sucessivamente até a sua conclusão.

Após a implementação do projeto foram feitos testes sistemáticos para a verificação do seu funcionamento, evitando-se a ocorrência de situações imprevistas e indesejáveis em sua operação.

O capítulo seguinte apresenta mais detalhes sobre a implementação do SAVAD.

## CAPÍTULO 4

### IMPLEMENTAÇÃO E VALIDAÇÃO DO SAVAD

O SAVAD foi implementado em um microcomputador compatível com sistemas IBM-PC com sistema operacional MS-DOS, utilizando a linguagem C++, versão 2.1 da ZORTECH [Zortech, 1990].

As propriedades de uma linguagem orientada a objetos tais como, polimorfismo, herança, abstração de dados e encapsulamento são plenamente implementadas por C++. A linguagem C++ também foi escolhida devido à sua eficiência na geração de código executável compacto e rápido [Eckel, 1989].

Uma outra razão da escolha dessa linguagem foi facilitar o transporte do SAVAD para outros ambientes operacionais. Para isto foram minimizadas as referências explícitas ao sistema Operacional e elementos de hardware do equipamento. Neste caso, a quantidade de código a ser reescrito será bastante reduzida.

#### 4.1 Dimensão do projeto

Na implementação do SAVAD foram utilizadas:

- 75 classes de objetos
  - 22 classes existentes na biblioteca fonte da ZORTECH.
  - 53 classes novas implementadas.
- 23.800 linhas de código fonte
  - 4.200 linhas de código fonte das classes existentes na biblioteca fonte da ZORTECH.
  - 19.600 linhas de código fonte das classes novas implementadas.

Na implementação da interface do SAVAD foram utilizadas:

- 48 classes de objetos.
- 16.500 linhas de código fonte.

## 4.2 Identificação do Ambiente Operacional e Capacidade de Processamento.

O SAVAD, durante sua inicialização, identifica o tipo de monitor do equipamento (cores ou preto e branco) e seleciona os atributos de vídeo apropriados.

Durante a execução do SAVAD, os objetos são criados e mantidos o tempo estritamente necessário. Ao término de sua utilização o objeto é eliminado, aumentando a disponibilidade de memória.

No SAVAD o limite para especificação de um modelo de redes de filas, normalmente é a quantidade de memória (principal) disponível no equipamento. As tabelas, os elementos do modelo de redes e outros itens do SAVAD são alocados dinamicamente, tendo apenas a disponibilidade de memória como fator limitante. Na presente versão, o SAVAD utiliza a memória convencional disponível em PC-XT's sob ambiente MS-DOS (640 KBytes).

O SAVAD faz uma avaliação dinâmica da quantidade de memória disponível para decidir se é possível o processamento dos modelos de redes utilizando a Análise do Valor Médio.

Nesta versão inicial, foram estabelecidos alguns limites conforme relacionados no apêndice D. Tais limites visam evitar o bloqueio excessivo de memória, e possibilitar a especificação de modelos de redes de filas de maior porte. Destinam-se também a simplificar a troca de informações entre o SAVAD e o simulador digital.

O número máximo de simulações digitais sucessivas foi especificado em 40. Para cada uma destas simulações o SAVAD gera uma semente (número pseudo-aleatório, primo, inferior a 32767) que pode ser alterada pelo usuário.

Na solução utilizando a Análise do Valor Médio a matriz de  $T_s(c)$  (tempo médio de serviço de um freguês da cadeia  $c$  no servidor  $s$ ) é obtida a partir dos tempos de serviço destas estações de serviço. Desta forma, para servidores FCFS, todas as cadeias visitando o servidor terão a mesma distribuição, exponencial, naquele servidor ( $T_s(c) = T_s$ ).

## 4.3 Arquivos do SAVAD

O SAVAD opera basicamente com três tipos de arquivos:

- Arquivos com o modelo da rede: arquivo do tipo binário e com extensão tipo ".RSV"
- Arquivos com o resultado da verificação: arquivos tipo texto com extensão ".VSV"
- Arquivos com a solução do modelo, obtida analiticamente ou por Simulação Digital: arquivos tipo texto com extensão ".SSV".

O SAVAD faz uma associação automática entre os arquivos. Os arquivos de verificação e solução têm o mesmo nome do arquivo com o modelo de rede, alterando apenas sua extensão.



- d) Estações de serviço: 0 - Não tem  
1 - Uma  
2 - Mais de uma
- e) Sorvedouros: 0 - Não tem  
1 - Um  
2 - Mais de um
- f) Outros elementos: 0 - Não tem  
1 - Tem

Identificação preliminar:

	<b>abcdef</b>
Modelo M/M/:	"111110"
Modelo Jackson:	"X1XXX0"
Modelo AVM:	"X20X00"

Como exemplo um modelo com uma rota aberta, uma fonte, uma estação de serviço, 1 sorvedouro, sem outros elementos será identificado como do tipo M/M/.

4.4.2. Se a identificação preliminar não satisfizer as condições acima, o modelo será submetido à Simulação Digital.

4.4.3. Se a identificação preliminar indicar o modelo M/M/, o motor de inferência procura fazer sua identificação definitiva [Kleinrock, 1975]. Se o modelo não for reconhecido, será submetido à Simulação Digital.

Para a identificação do modelo são utilizadas as seguintes regras:

- a) Número de rotas: 1 - Uma rota  
2 - Mais de uma rota
- b) Tipo de rota: 1 - Aberta  
2 - Fechada
- c) Conjunto de elementos: 1 - Contém fonte, estação de serviço e sorvedouro  
2 - Contém outros elementos
- d) FDP da fonte: 1 - Exponencial  
2 - Outra
- e) FDP da estação: 1 - Exponencial  
2 - Outra

- f) Disciplina da estação: 1 - FCFS  
2 - Outra
- g) Quantidade de servidores: 1 - Um  
2 - Múltiplos  
3 - Infinitos
- h) Fator de desencorajamento( $\alpha$ ): 0 - Não tem  
1 - Tem
- i) Comprimento de fila: 0 - Sem limite  
1 - Limitado
- j) População: 0 - Sem limite  
1 - Limitada
- k) Compr. fila = num. serv.: 1 - Sim  
2 - Não

## Identificação:

	<b>abcdefghijkl</b>
M/M/1:	"1111111000X"
M/M/1//M:	"1111111001X"
M/M/1/K:	"1111111010X"
M/M/1/ $\alpha$ :	"1111111100X"
M/M/m:	"1111112000X"
M/M/m/m:	"11111120102"
M/M/m/K/M:	"1111112011X"
M/M/ $\infty$ :	"1111113000X"
M/M/ $\infty$ //M:	"1111113001X"

4.4.4. Se a identificação preliminar indicar o modelo com solução utilizando o Teorema de Jackson, o motor de inferência tenta fazer sua identificação definitiva [Moura et al., 1986]. Se o modelo não for reconhecido, será submetido à Simulação Digital.

Para isto serão utilizadas as seguintes regras:

- a) Tipo de rota: 1 - Aberta  
2 - Fechada
- b) Conjunto de elementos: 1 - Contém fonte, estação de de serviço e sorvedouro  
2 - Contém outros elementos

- c) FDP da fonte: 1 - Exponencial  
2 - Outra
- d) FDP da estação: 1 - Exponencial  
2 - Outra
- e) Disciplina da estação: 1 - FCFS  
2 - Outra
- f) Quantidade de servidores: 1 - Um  
2 - Múltiplos  
3 - Infinito
- g) Fator de desencorajamento( $\alpha$ ): 0 - Não tem  
1 - Tem
- h) Comprimento de fila: 0 - Sem limite  
1 - Limitado
- i) População: 0 - Sem limite  
1 - Limitada
- j) Classes: 1 - Um só tipo  
2 - Vários tipos

Identificação:

	abcdefghijkl
Jackson:	"1111110001"
Jackson:	"1111120001"

4.4.5. Se a identificação preliminar indicar para o modelo uma solução utilizando a Análise do Valor Médio, o motor de inferência tenta fazer sua identificação definitiva [Reiser, 1979]. Se o modelo não for reconhecido, será submetido para Simulação Digital.

Para isto serão utilizadas as seguintes regras:

- a) Tipo de rota: 1 - Aberta  
2 - Fechada
- b) Conjunto de elementos: 1 - Contém apenas estações de serviço  
2 - outros

- c) FDP da estação: 1 - Exponencial  
2 - Outra
- d) Disciplina da estação: 1 - FCFS  
2 - Outra
- e) Quantidade de servidores: 1 - Um  
2 - Múltiplos  
3 - Infinito
- f) Fator de desencorajamento( $\alpha$ ): 0 - Não tem  
1 - Tem
- g) Comprimento de fila: 0 - Sem limite  
1 - Limitado
- h) População: 0 - Sem limite  
1 - Limitada
- i) Classes: 1 - Um só tipo  
2 - Vários tipos
- j) Limite de complexidade: 1 - Permite o cálculo de AVM  
2 - Não permite o cálculo

Identificação:

a b c d e f g h i j

Modelo AVM: "211110X111"

Modelo AVM: "211120X111"

4.4.6. Após a identificação do modelo, o motor de inferência aciona o Módulo Solução para o processamento do modelo apropriado.

#### 4.5 Ativação do Simulador Digital

A memória disponível em um micro-computador PC-XT com ambiente MS-DOS poderia ser insuficiente para executar o simulador digital caso o SAVAD permanecesse residente na memória. Para solucionar este problema foi adotado um processamento cooperativo, descrito a seguir:

- O simulador digital foi implementado em um programa executável à parte, funcionando de forma integrada ao sistema conforme descrito em [Conceição Filho et al., 1992].

- Antes do acionamento do simulador, o SAVAD grava o arquivo com o modelo de rede em análise, encerra suas atividades, e aciona o simulador digital especificando o arquivo com o modelo de rede a ser calculado. Esta operação ocorre de forma transparente ao usuário. O usuário recebe apenas um aviso de que a Simulação Digital irá ser iniciada. Desta forma toda a memória do computador fica disponível para o simulador digital.
- O simulador digital executa o processamento para a determinação da solução do modelo.
- Ao término da Simulação Digital, o programa simulador encerra suas atividades, e aciona o SAVAD informando além do nome do arquivo do modelo, o parâmetro "/s". Esta operação é executada de forma transparente para o usuário. Desta forma toda a memória do computador volta a ficar disponível para o SAVAD.

Ex: *SAVAD ARQUIVO.RSV /S* (ativado pelo programa simulador)

- O SAVAD inicia suas atividades e, ao reconhecer o parâmetro "/s", lê o arquivo de solução do modelo de rede e apresenta o resultado da simulação para o usuário.
- A comunicação entre o SAVAD e o programa simulador é feita através de arquivos com o modelo da rede de filas enviado pelo SAVAD para o programa simulador e o arquivo com o resultado da simulação enviado para o SAVAD.

#### 4.6 Dinâmica de Funcionamento do SAVAD

O SAVAD possui um alto grau de interatividade com o usuário, constituindo-se em um sistema dirigido por eventos gerados pelo usuário ("event-driven").

As classes principais do savad são a rede (*zRede*), o ambiente de controle do programa (*zAmbiente*), os elementos de modelagem (*zServidor*, *zRota*, etc.), o motor de inferência (*zMotorInf*), base de conhecimento (*zBaseConh*) e as soluções analíticas (*zMMI*, *zMMIK*, etc.).

Em várias etapas do processamento são ativadas e utilizadas classes de apoio operacional (listas encadeadas, arrays, etc.) bem como classes de implementação da interface (mensagens, editores de campo, menus, eventos, tela de trabalho, etc.).

Vamos descrever as etapas de funcionamento do SAVAD:

##### a) Inicialização

Nesta etapa os objetos das classes básicas do SAVAD são criados (rede, tela de trabalho, menu principal, ajuda, etc.). A tela básica de operação do sistema é apresentada ao usuário.

##### b) Estado de Monitoração do Ambiente

Nesta etapa o objeto da classe *zAmbiente* passa a controlar a execução do programa e responder às solicitações do usuário feitas através do menu principal.

A monitoração estará ocorrendo no sistema sempre que o menu principal estiver ativo. Quando uma atividade for concluída ou interrompida pelo usuário, o sistema retornará ao estado de monitoração, aguardando uma outra ação do usuário. A partir desta classe são ativadas todas as funções do SAVAD.

As funções de leitura, criação, gravação, cópia e renomeação de arquivos são implementadas pelo acionamento, a partir de *zAmbiente*, de métodos do objeto da classe *zRede*. A classe *zRede* permite a manipulação do modelo de redes de filas de forma global e de seus elementos, individualmente. A figura 4.1 apresenta um diagrama cliente-servidor com estas operações básicas.

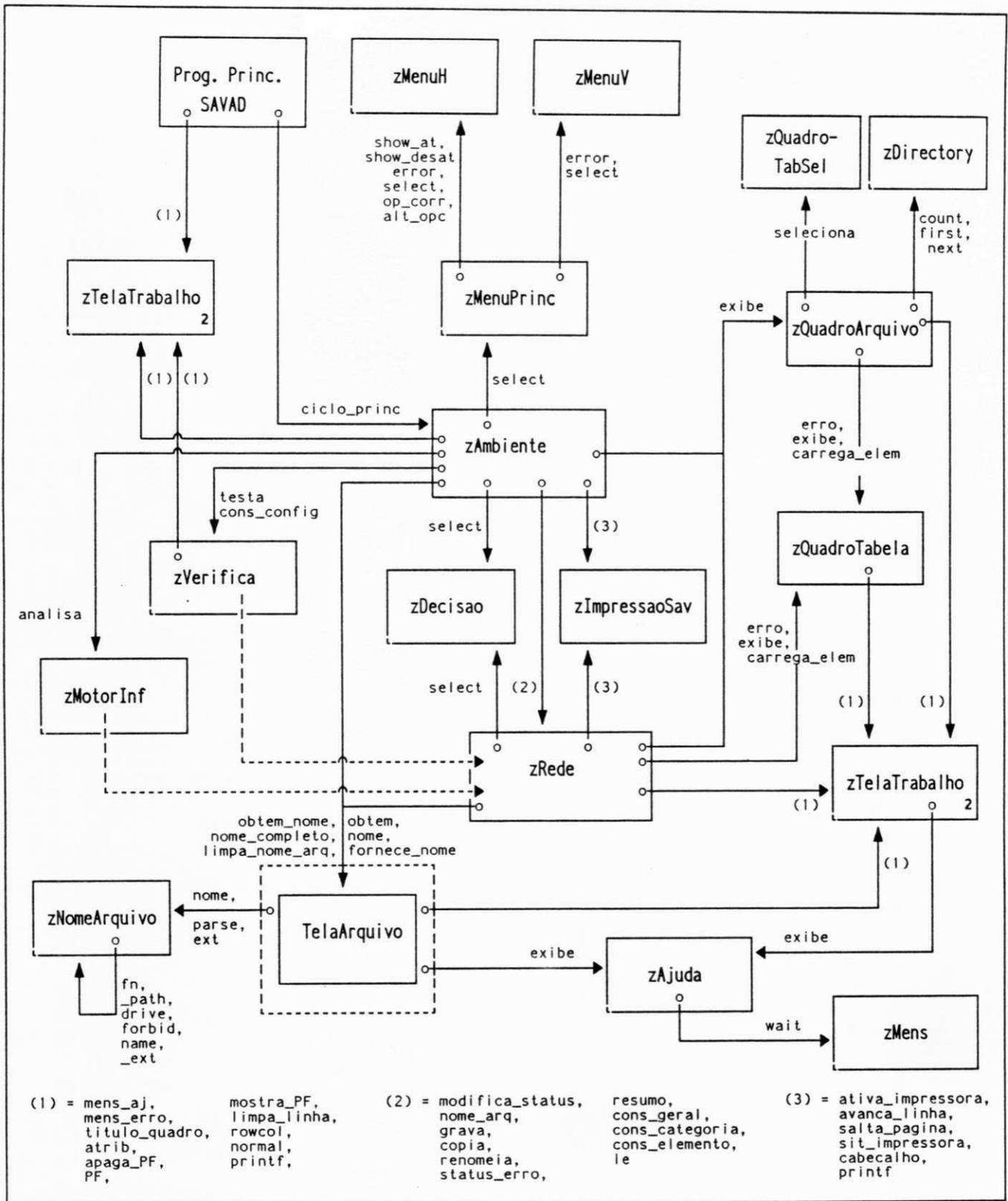


Fig. 4.1 - Diagrama Cliente-Servidor do Acesso Global ao Modelo e Monitoração da Interface

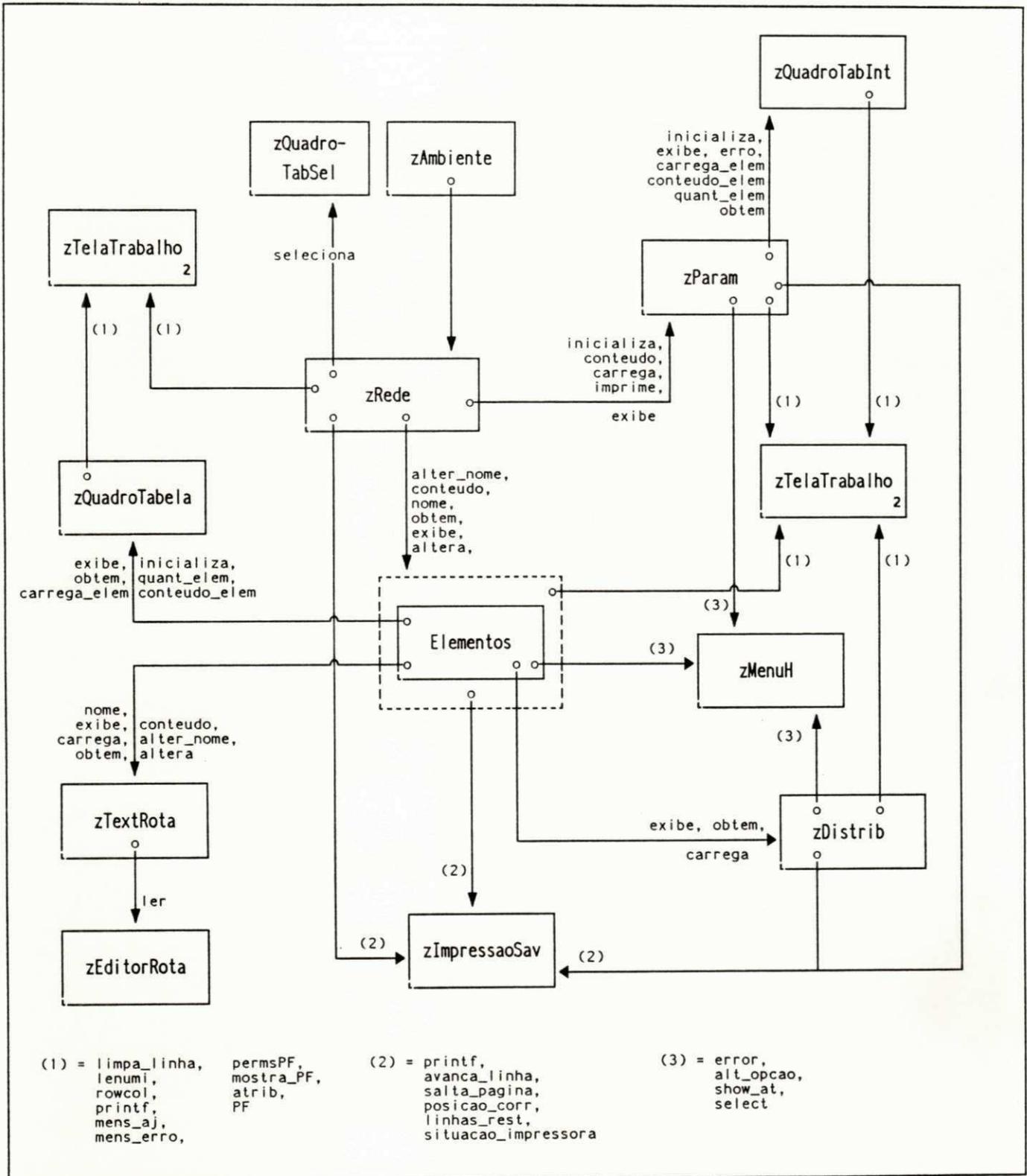


Fig. 4.2 - Diagrama Cliente-Servidor da Inclusão/Atualização de Elementos do Modelo

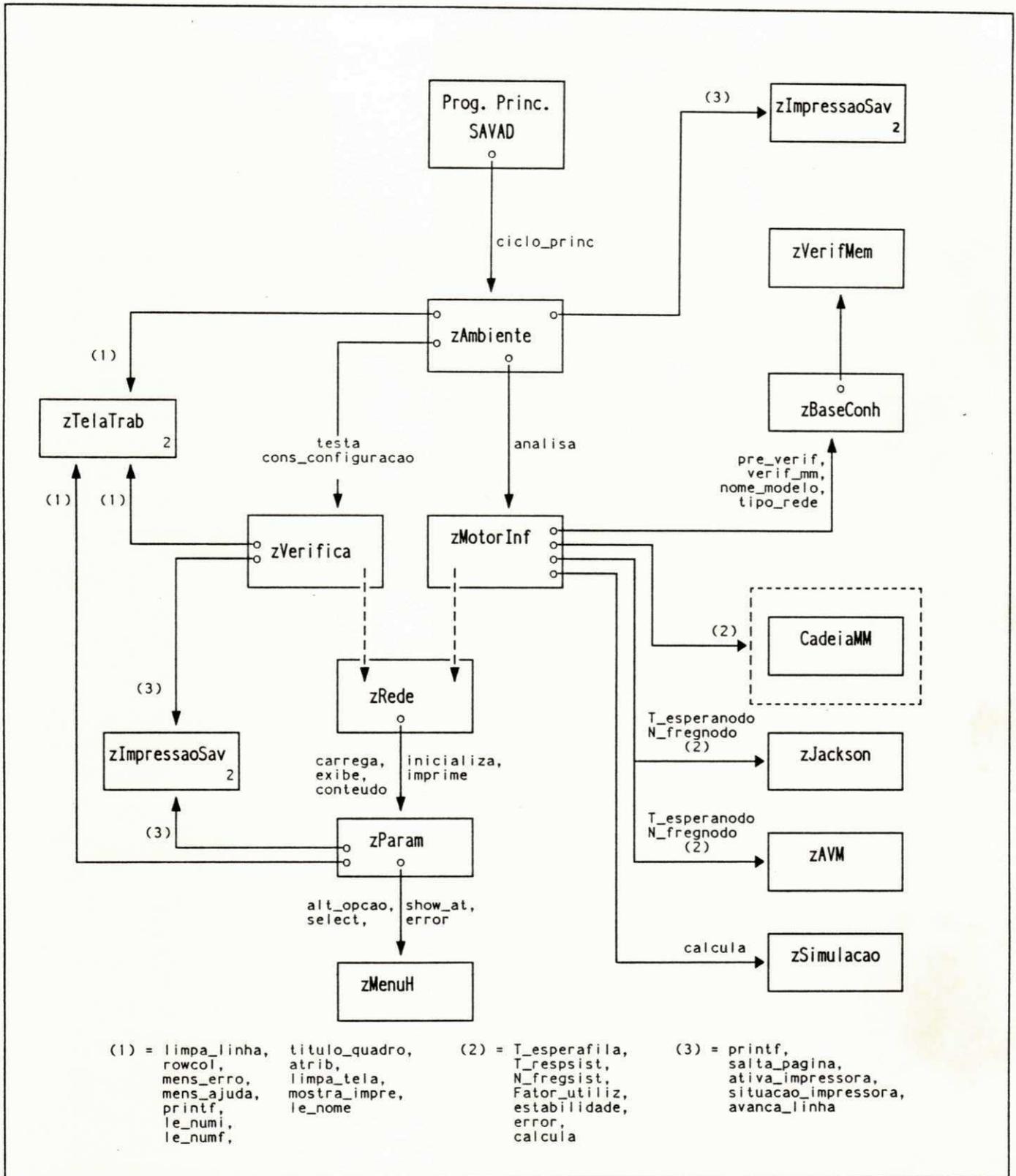


Fig. 4.3 - Diagrama Cliente-Servidor da Verificação de Integridade e Solução do Modelo

## 4.7 Validação do SAVAD

Após a implementação do SAVAD foram feitos testes sistemáticos para a validação do funcionamento dos módulos do SAVAD.

Apresentamos a seguir exemplos de modelos de redes de filas, definidas informalmente, solucionados usando técnicas analíticas implementadas no módulo solução do SAVAD. As soluções de redes de filas usando simulação fazem parte da Dissertação de Mestrado do aluno Haroldo Conceição Filho, já referenciada.

### a) Sistema M/M/1

O exemplo consiste de uma conexão com uma entidade que envia mensagens para outra através de um canal de comunicação. Neste mecanismo não existe reconhecimento para as mensagens recebidas.

A figura 4.4 apresenta o modelo da rede:

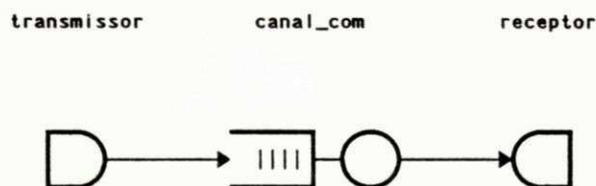


Fig. 4.4 - Modelo de Rede M/M/1

- A geração de mensagens é feita por uma fonte.
- O canal de comunicação é representado por uma estação de serviço.
- A entidade receptora é representada por um sorvedouro.

Características da entidade emissora:

- Distribuição dos tempos de interchegada de mensagens: Exponencial
- Taxa média de chegada: 1 mensagem por segundo.

Características da estação de serviço (canal de comunicação):

- Fila sem restrição de comprimento.
- Distribuição do tempo de serviço: Exponencial
- Taxa média de serviço: 2 mensagens por segundo
- Disciplina de escalonamento: FCFS

A solução fornecida pelo SAVAD para o modelo é a seguinte:

TIPO DE SOLUÇÃO: ANALÍTICA

MODELO DE REDE: M/M/1

MEDIDAS DE DESEMPENHO

Tempo médio de espera em fila:	0.50
Tempo médio de resposta do sistema:	1.00
Número médio de clientes no sistema:	1.00
Fator de utilização:	0.50

**b) Modelo de Rede Aberta com Realimentação**

O exemplo consiste de uma rede aberta com realimentação, conforme mostra a figura 4.5. Esse modelo pode ser solucionado usando o Teorema de Jackson.

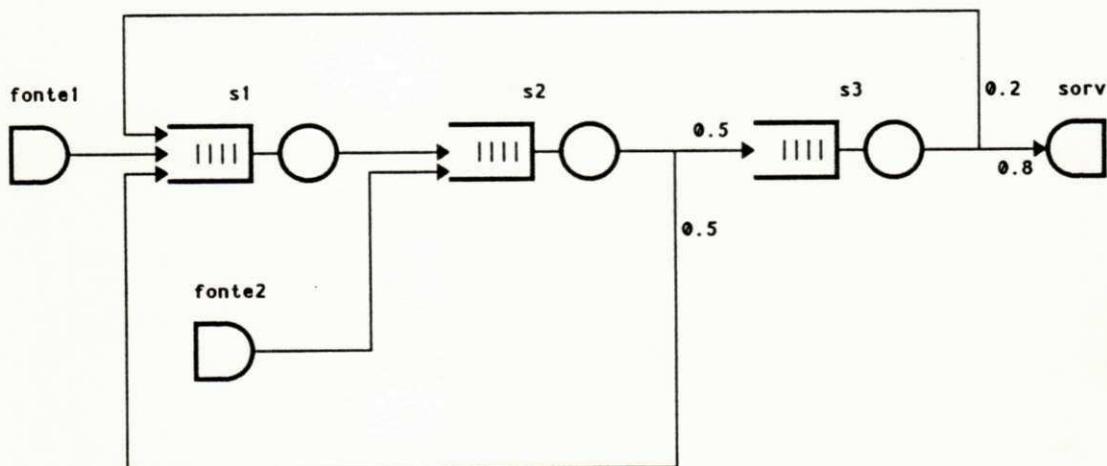


Fig. 4.5 - Modelo de Rede de Filas Aberta com Realimentação - Teorema de Jackson

## Características das fontes:

- Distribuição dos tempos de interchegada de mensagens: Exponencial
- Taxa média de geração:

Fonte f1: 1 mensagem por segundo

Fonte f2: 1,5 mensagens por segundo

## Características das estações de serviço:

- Filas sem restrições de comprimento.
- Distribuição dos tempos de serviço: Exponencial
- Disciplina de escalonamento: FCFS
- Número de servidores: 1
- Taxa média de serviço:

Estação s1: 5 mensagens por segundo

Estação s2: 7 mensagens por segundo

Estação s3: 4 mensagens por segundo

- Rotas:

a) Nome: r1

Classe: Sem prioridade

$fonte1 \rightarrow s1 \rightarrow s2 \rightarrow (0.5 s1, 0.5 s3)$

$s3 \rightarrow (0.2 s1, 0.8 sorv)$

b) Nome: r2

Classe: Sem prioridade

$fonte2 \rightarrow 2 \rightarrow (0.5 s1, 0.5 s3)$

$s1 \rightarrow s2$

$s3 \rightarrow (0.2 s1, 0.8 sorv)$

Temos a seguir a solução fornecida pelo SAVAD para este modelo:

TIPO DE SOLUÇÃO: ANALÍTICA

MODELO DE REDE: ABERTA COM REALIMENTAÇÃO (TEOREMA DE JACKSON)

MEDIDAS DE DESEMPENHO

Estação: s1

Tempo médio de espera no servidor:	4.000000
Comprimento de fila:	19.000000
Fator de utilização:	0.950000

Estação: s2

Tempo médio de espera no servidor:	1.333333
Comprimento de fila:	8.333333
Fator de utilização:	0.892857

Estação: s3

Tempo médio de espera no servidor:	1.142857
Comprimento de fila:	3.571428
Fator de utilização:	0.781250

Tempo de permanência no sistema:	6.476190
----------------------------------	----------

### c) Modelo de Rede Fechada

O exemplo consiste de uma rede fechada com múltiplas rotas, conforme mostra a figura 4.6. Esse modelo pode ser solucionado usando a Análise do Valor Médio.

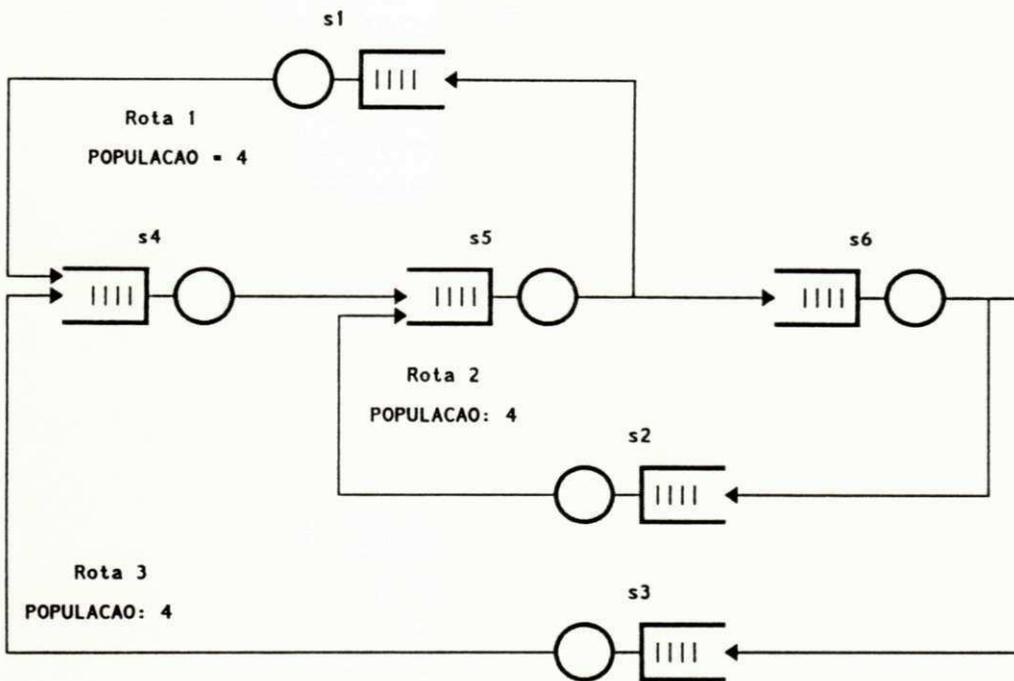


Fig. 4.6 - Modelo de Rede de Filas Fechada - Análise do Valor Médio

Características das estações de serviço:

- Filas sem restrições de comprimento.
- Distribuição dos tempos de serviço: Exponencial
- Disciplina de escalonamento: FCFS
- Número de servidores: 1
- Taxa média de transmissão:

Estação s1: 1 mensagem por segundo

Estação s2: 2 mensagens por segundo

Estação s3: 3 mensagens por segundo

Estação s4: 3 mensagens por segundo

Estação s5: 3 mensagens por segundo

Estação s6: 1 mensagem por segundo

- Rotas:

a) Nome: r1

População: 4

Classe: Sem prioridade

$s1 \rightarrow s4 \rightarrow s5 \rightarrow s1$

b) Nome: r2

População: 4

Classe: Sem prioridade

$s5 \rightarrow s6 \rightarrow s2 \rightarrow s5$

c) Nome: r3

População: 4

Classe: Sem prioridade

$s4 \rightarrow s5 \rightarrow s6 \rightarrow s3 \rightarrow s4$

Temos a seguir a solução fornecida pelo SAVAD para este modelo:

TIPO DE SOLUÇÃO: ANALÍTICA

MODELO DE REDE: ABERTA (ANÁLISE DO VALOR MÉDIO)

MEDIDAS DE DESEMPENHO

Cadeia: r1

Vazão média: 0.767315

Tempo médio de circulação: 5.212983

Cadeia: r2

Vazão média: 0.443647

Tempo médio de circulação: 9.016168

Cadeia: r3

Vazão média: 0.429003

Tempo médio de circulação: 9.323940

Servidor: s1

Comprimento médio de fila: 2.952103

Servidor: s2

Comprimento médio de fila: 0.287358

Servidor: s3  
Comprimento médio de fila: 0.167383

Servidor: s4  
Comprimento médio de fila: 0.686316

Servidor: s5  
Comprimento médio de fila: 1.273829

Servidor: s6  
Comprimento médio de fila: 6.633010

## CAPÍTULO 5

### CONCLUSÕES E SUGESTÕES

O Sistema de Avaliação de Desempenho de Modelos de Redes de Filas (SAVAD) foi desenvolvido visando oferecer uma ferramenta de uso fácil e poderoso para a modelagem e solução de sistemas de redes de filas. Em seu desenvolvimento foi empregada a abordagem orientada a objetos e utilizada a linguagem C++.

O SAVAD fornece um ambiente integrado com uma interface amigável, inteligente e poderosa, permitindo ao usuário facilmente especificar o seu modelo de redes de filas. Para isto dispõe de um conjunto de elementos poderosos e flexíveis, que permitem representar o modelo desejado de forma simples e abrangente. O SAVAD constitui-se de um sistema especialista onde, a partir de um modelo de redes de filas proposto pelo usuário, o sistema escolhe e aplica a técnica mais adequada para solucionar este modelo, seja esta analítica ou numérica, fornecendo as medidas de desempenho relevantes.

O SAVAD possibilita a ampliação dos tipos de elementos de modelagem e das soluções analíticas. Esta ampliação é facilitada pelas características de modularidade, extensibilidade e reutilização de software da abordagem orientada a objetos, utilizada na construção do SAVAD. Essas características permitem, a nível de projeto, prever e simplificar expansões futuras.

O SAVAD também prevê a possibilidade de sua utilização como interface para outros programas de solução de modelos de redes de filas mais específicos.

No projeto do SAVAD foram exploradas todas as vantagens básicas da abordagem orientada a objetos tais como modularidade, reusabilidade, abstração de dados e polimorfismo. A linguagem utilizada (C++) oferecem estas facilidades além de algumas outras, mais específicas, como por exemplo, acesso "friend" entre classes.

A inexistência de técnicas consolidadas de projeto orientado a objetos aliada à inadequação das metodologias estruturadas convencionais dificultaram bastante o desenvolvimento do SAVAD. As propostas de representação de diversos autores (Booch, Pressman, Wasserman, Korson, Henderson-Sellers, Wirfs-Brock, Gibbs, Coad-Yordon, etc.), não estão consolidadas e apresentam notações ainda divergentes entre si e que eventualmente não representam com a concisão e presteza necessárias todos os aspectos do projeto ou da linguagem de programação empregada.

As dimensões e a complexidade do projeto do SAVAD (75 classes e 23.800 linhas de código) exigem a utilização de técnicas de projeto eficientes. Foram feitas combinações e extensões para as diversas técnicas apresentadas pelos autores para a especificação do projeto do

SAVAD, conforme descrito no capítulo 3. Desta forma este trabalho contribui também para o desenvolvimento da metodologia de projeto orientado a objetos em projetos de porte significativo.

A validação do SAVAD foi feita através de testes sistemáticos e abrangentes. As soluções analíticas foram testadas comparando os resultados fornecidos pelo SAVAD com os existentes na bibliografia especializada.

O programa executável do SAVAD resultante é de dimensão razoavelmente pequena, tendo em vista as funções executadas. Deve-se considerar que esta é uma versão inicial do aplicativo, em que não foram feitas otimizações especiais. Os requisitos de hardware e software para o SAVAD são bastante comuns atualmente. Estes fatores permitem uma fácil disseminação e utilização do SAVAD.

Como sugestão para otimização e continuação do projeto do SAVAD, propomos:

- a) Expansão do SAVAD com a implementação de novas soluções analíticas para modelos de redes de filas. Estas inclusões deverão ter impacto mínimo no projeto básico. Maiores detalhes serão fornecidos na próxima seção.
- b) Implementação de uma interface gráfica para a especificação do modelo. Tal interface pode "traduzir" as descrições gráficas do modelo para os elementos do SAVAD existentes. Podem ser aproveitadas todas as classes e métodos já existentes, reduzindo-se e simplificando-se as alterações a serem feitas.
- c) Acoplamento de outros programas de solução de modelos de rede de fila (externamente). Maiores detalhes serão fornecidos na próxima seção.
- d) Inclusão de novos elementos de modelagem ao SAVAD. Maiores detalhes serão fornecidos na próxima seção.
- e) Aperfeiçoamento da metodologia do projeto orientado a objetos, utilizando a experiência do projeto do SAVAD.
- f) Desenvolvimento de um software de documentação automática que permita a análise do código fonte de sistemas desenvolvidos em C++ e a geração das informações básicas dos diagramas de Booch e herança/cliente-servidor.
- g) Portar o SAVAD para outros ambientes operacionais, em particular o UNIX e o WINDOWS.
- h) Aprimoramento da integração do SAVAD e do simulador digital, tornando-os mais acoplados entre si.

- i) Apresentação dos resultados de forma gráfica.
- j) Inclusão de um módulo de explanação para informar ao usuário as opções feitas pelo SAVAD.

## 5.1 Expansão do SAVAD

As características de modularidade, extensibilidade e reutilização de software inerentes à abordagem orientada a objetos foram amplamente utilizadas no projeto do SAVAD. Por outro lado, o projeto prevê a possibilidade de adaptações e expansões que se farão necessárias, reduzindo-se seus impactos no projeto ao mínimo indispensável.

Os principais itens do SAVAD que são passíveis de atualização são:

- Inclusão de novas soluções analíticas.
- Inclusão de novos elementos de modelagem.
- Acoplamento de outros programas de solução de modelos de rede de fila (externamente).

Para se fazer uma análise do impacto no projeto devido a uma expansão, deve-se analisar as classes afetadas direta ou indiretamente pela mudança, analisando os relacionamentos cliente-servidor e as classes herdeiras. Para isto pode-se usar os protocolos das classes e os diagramas cliente-servidor e herança/cliente-servidor.

### a) Etapas Necessárias para a Inclusão de Novas Soluções Analíticas.

Como podemos verificar no diagrama cliente-servidor (Apêndice B) e no diagrama geral de herança (Apêndice A), as classes correspondentes a estas soluções são independentes entre si. Apenas a classe *zMotorInf* utiliza os serviços destas classes. Desta forma a inclusão de novas soluções será simples e com impacto mínimo.

- Identificação das classes de objetos que irão implementar as novas soluções.
- Identificação dos atributos das novas classes.
- Identificação das funções (métodos) das novas classes. Provavelmente serão similares às das classes já existentes para aproveitar as propriedades de polimorfismo e herança.
- Alteração das classes *zMotorInf* (motor de inferência) e *zBaseConh* (base de conhecimento) para incluir as características dos novos modelos, permitindo seu reconhecimento e a implementação de novas soluções analíticas.

**b) Etapas Necessárias para a Inclusão de Novos Elementos de Descrição do Modelo**

Como podemos verificar nos diagramas cliente-servidor e no diagrama geral de herança, as classes correspondentes a estes elementos são independentes entre si. Estas classes têm um maior grau de acoplamento com as demais classes do SAVAD, por isto a análise deve ser mais cuidadosa.

- Identificação das classes de objetos que irão implementar os novos elementos.
- Identificação dos atributos das novas classes.
- Identificação das funções (métodos) das novas classes. Provavelmente serão similares às das classes já existentes para aproveitar as propriedades de polimorfismo e utilizar os mecanismos de herança.
- Alteração das classes *zElemento*, *zRede*, *zVerifica*, *zMotorInf* (motor de inferência) e *zBaseConh* (base de conhecimento) para refletir as características dos novos elementos, permitindo seu reconhecimento e solução.
- Alteração do formato do arquivo com o modelo de redes de filas para incorporar os novos elementos.
- Alteração do programa de Simulação Digital para que este possa se adaptar ao novo formato de arquivo de modelo de rede e incorporar as propriedades destes novos elementos na simulação.

**c) Etapas Necessárias para o Acoplamento Externo de Outros Programas de Solução de Modelos de Redes de Filas.**

A alteração implica em modificações na interface do SAVAD. O nível de integração deste novo programa com o SAVAD determinará o volume das alterações a serem realizadas.

Caso o programa de solução seja completamente autônomo, as alterações no SAVAD serão mínimas, podendo implicar porém em padrões de diálogo com o usuário distintos (a interface do SAVAD diferente da interface do programa de solução). Se a interface do novo programa for implementada no SAVAD, a quantidade de alterações dependerá dos dados a serem fornecidos/recebidos pelo usuário.

- Identificação das classes de objetos que irão implementar a nova solução.
- Identificação dos atributos das novas classes.
- Identificação das funções (métodos) das novas classes.

- Identificação das mensagens e dos relacionamentos com outras classes já existentes.
- Alteração das classes *zAmbiente*, *zMenuPrincipal*, *zRede*, *zVerifica*, *zMotorInf*, *zBaseConh* e *zParam* para incluir as características deste novo programa de solução de modelos, permitindo seu reconhecimento e acionamento do programa específico.
- Alteração do formato do arquivo com o modelo de redes de filas para incorporar as características específicas deste programa.
- Alteração do programa de Simulação Digital para se adaptar ao novo formato de arquivo de modelo de rede mesmo que não solucione o novo modelo de rede.

## APÊNDICE A

### DIAGRAMAS DE HERANÇA DAS CLASSES

DIAGRAMA DE HERANCA DAS CLASSES

ELEMENTOS DE MODELAGEM E CLASSE zRede

REFERENCIA: 1 / 6

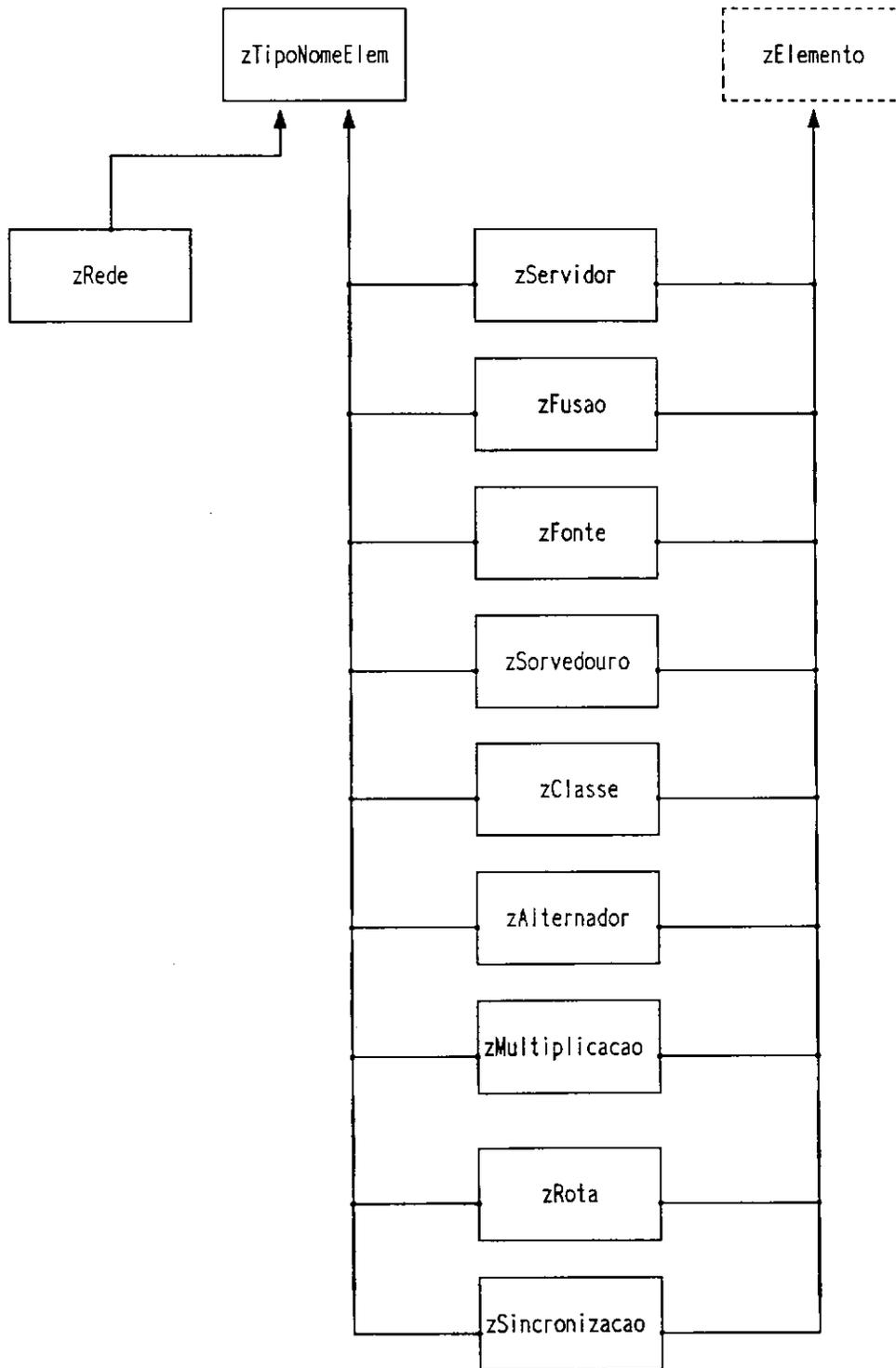


DIAGRAMA DE HERANCA DAS CLASSES

CLASSES QUE OPERAM COM A INTERFACE COM O USUARIO

REFERENCIA: 2 / 6

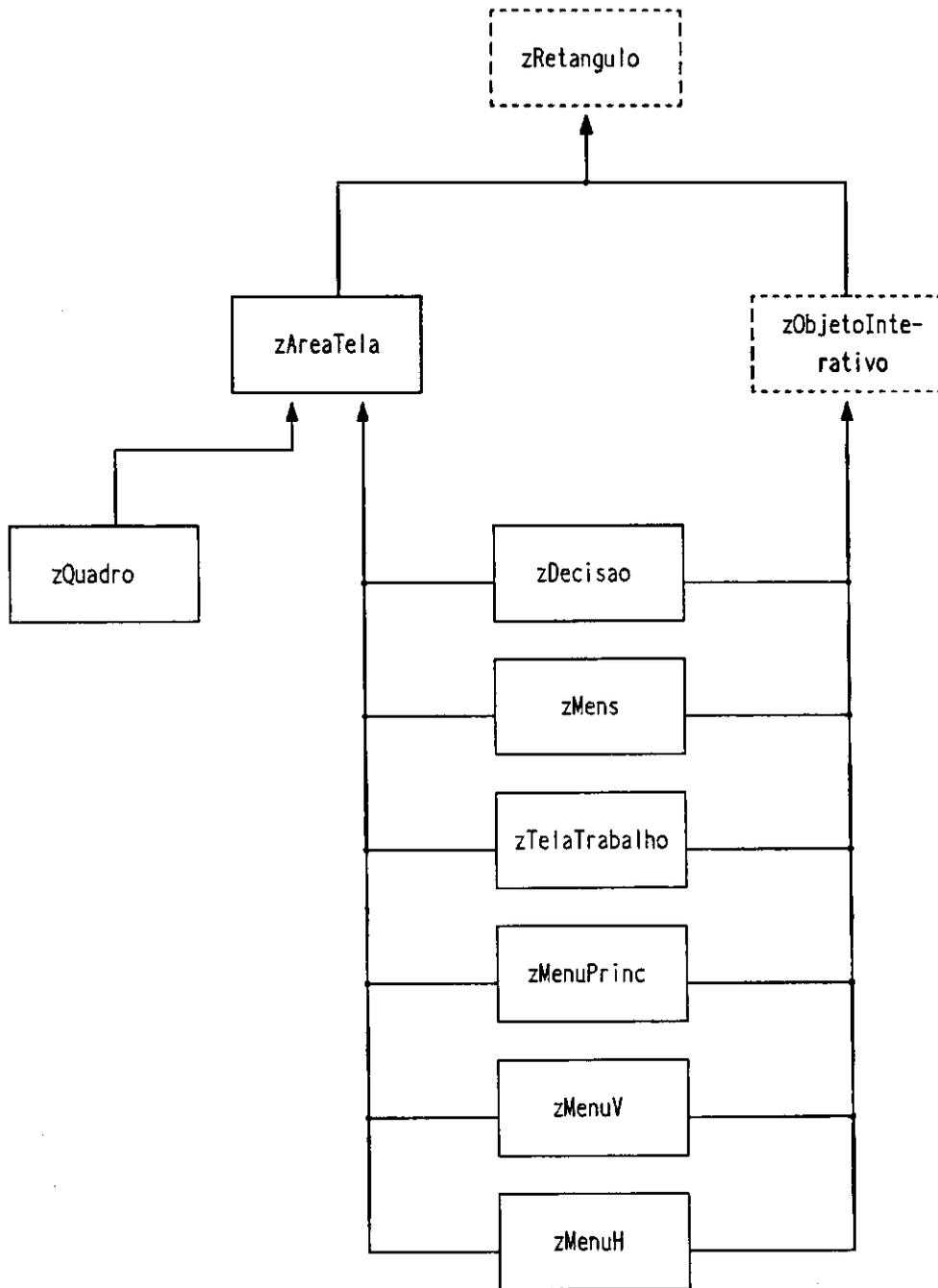


DIAGRAMA DE HERANCA DAS CLASSES

CLASSES QUE FAZEM A EDICAO DE STRING, NUMEROS, DATA, etc.

REFERENCIA: 3 / 6

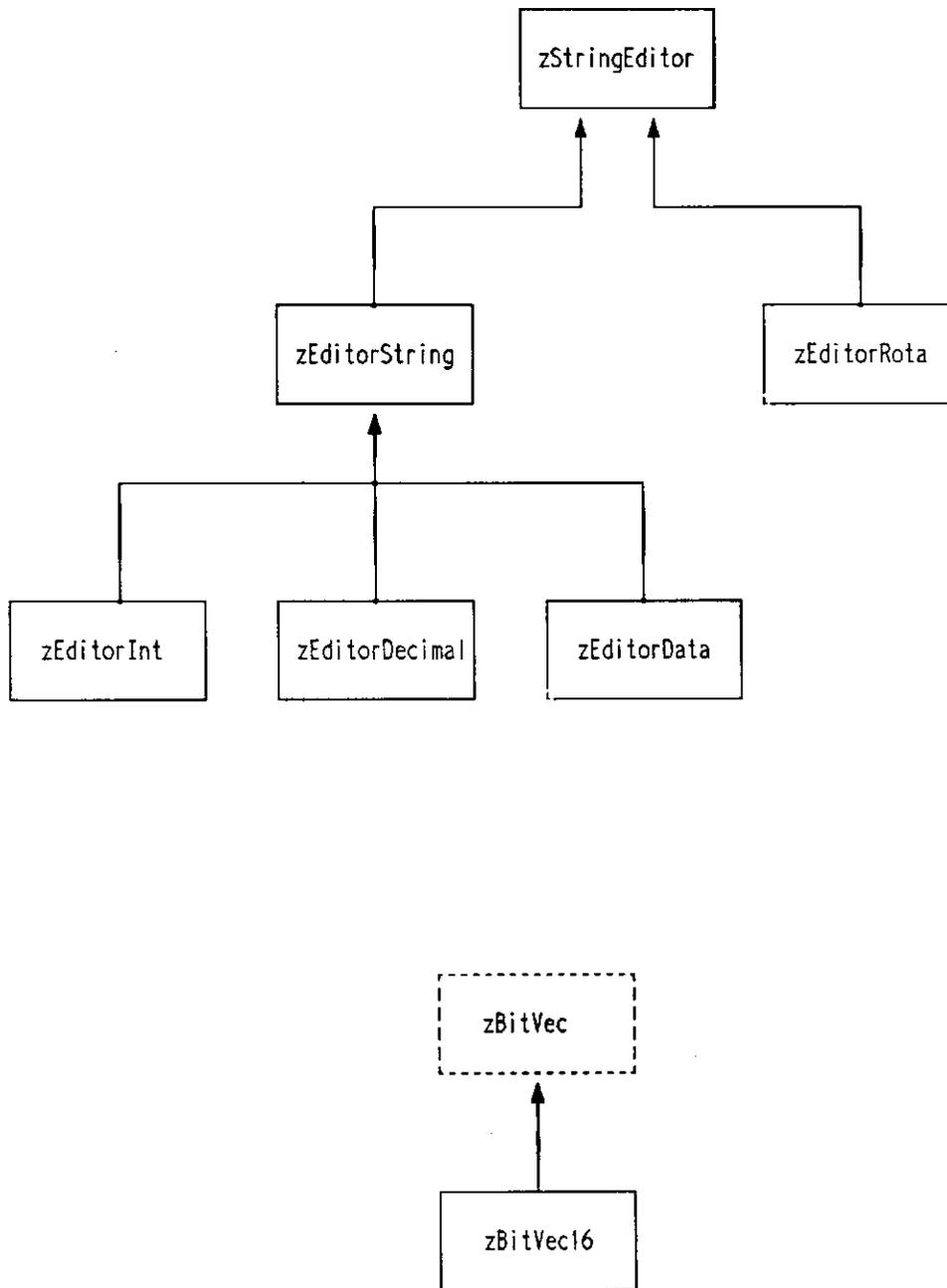


DIAGRAMA DE HERANCA DAS CLASSES

CLASSES DE APOIO (LISTAS, PILHAS, ARRAY, etc)

REFERENCIA: 4 / 6

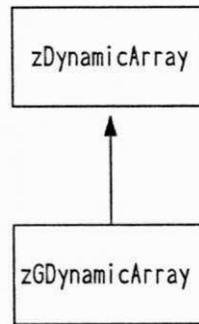
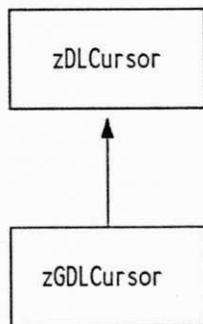
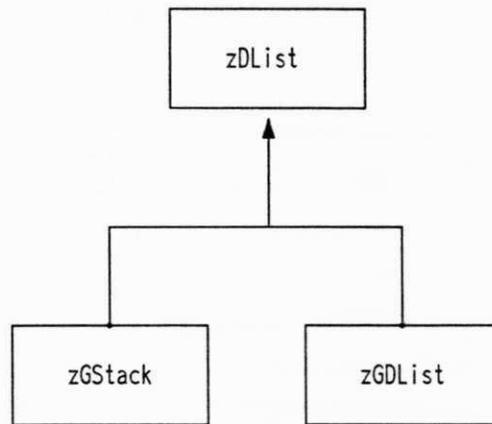


DIAGRAMA DE HERANCA DAS CLASSES

CLASSES DE SOLUCAO DE REDES TIPO M/M/

REFERENCIA: 5 / 6

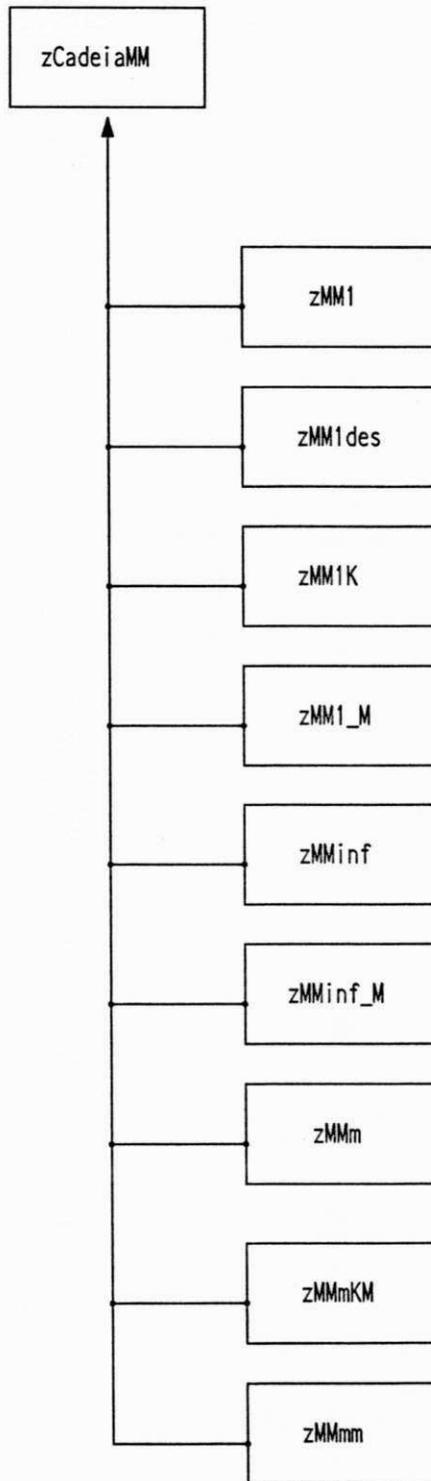
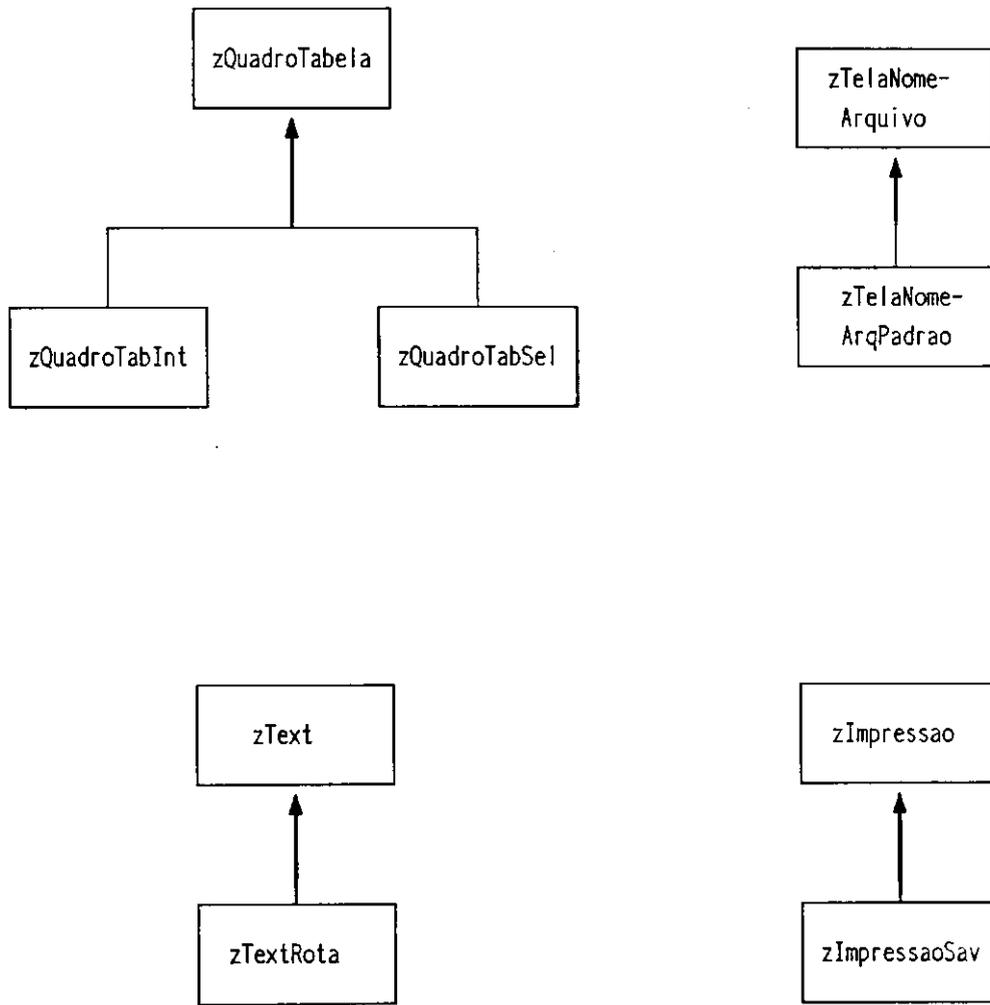


DIAGRAMA DE HERANCA DAS CLASSES

CLASSES DE APOIO (TABELAS, IMPRESSAO, TEXTO)

REFERENCIA: 6 / 6



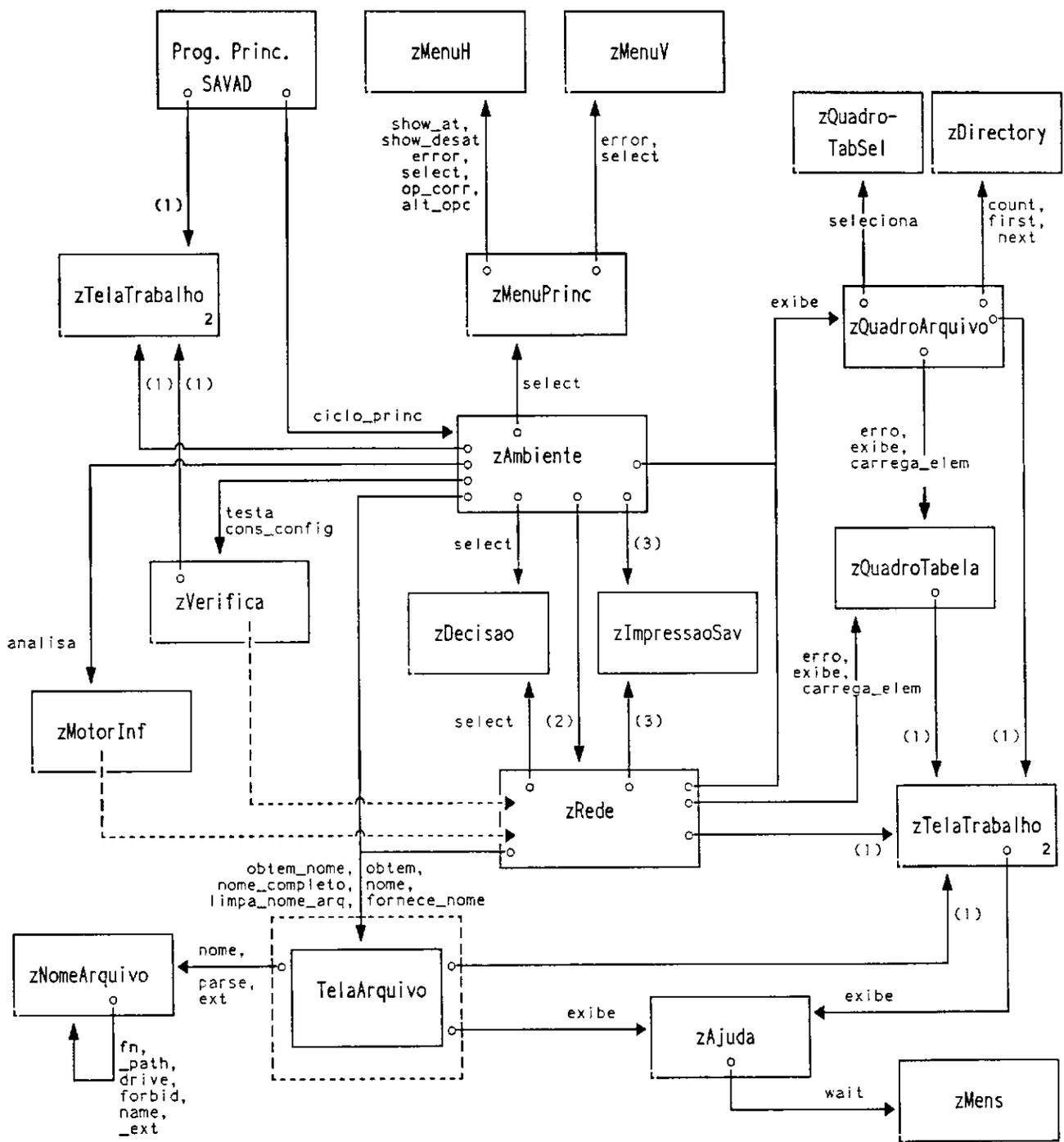
## APÊNDICE B

### DIAGRAMAS CLIENTE-SERVIDOR

DIAGRAMA GERAL CLIENTE-SERVIDOR

INICIALIZACAO, ACESSO GLOBAL AO MODELO E MONITORACAO DA INTERFACE

REFERENCIA: 1 / 12



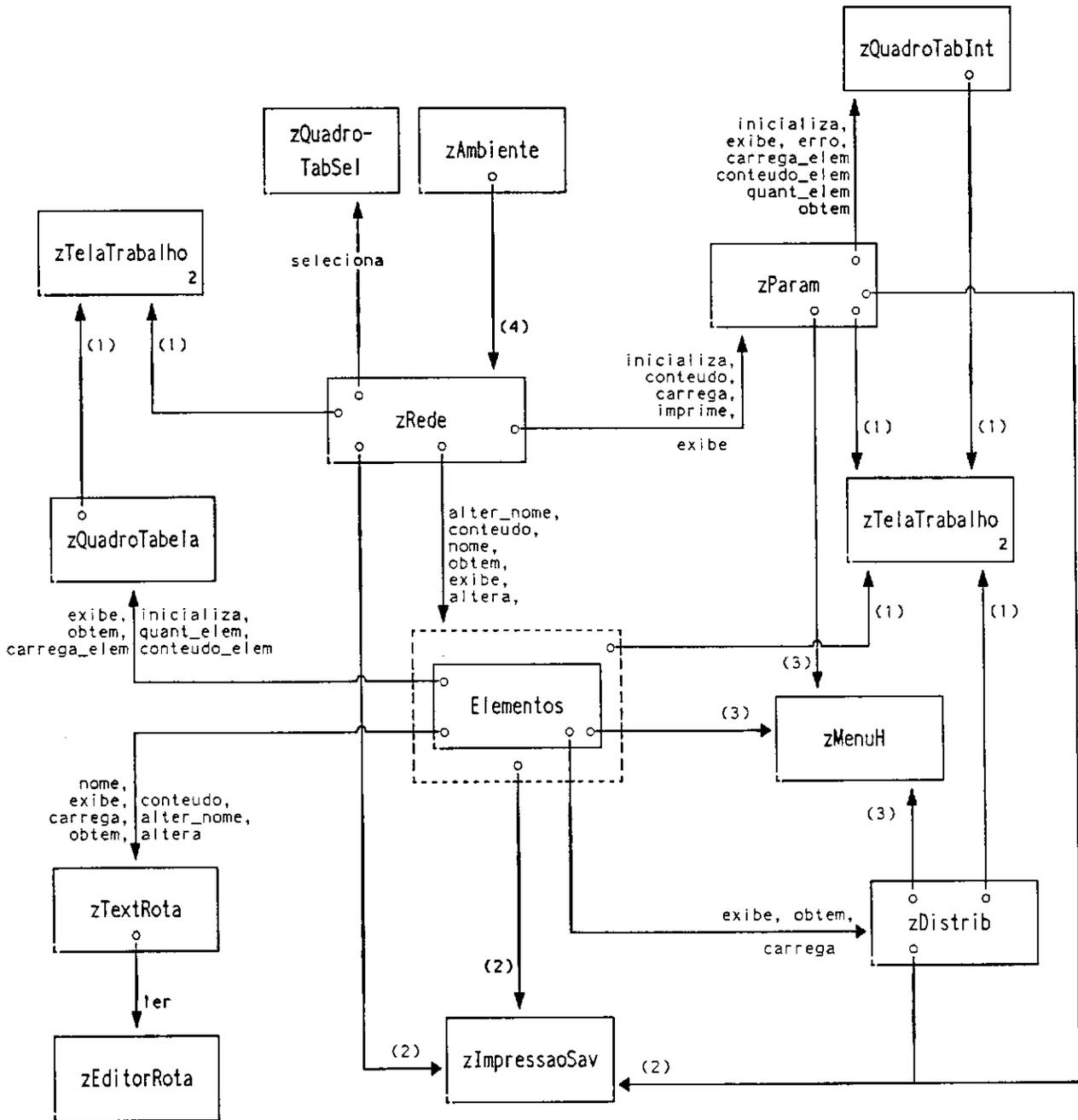
(1) = mens\_aj, mens\_erro, titulo\_quadro, atrib, apaga\_FF, PF,

mostra\_FF, limpa\_linha, rowcol, normal, printf,

(2) = modifica\_status, nome\_arq, grava, copia, renomeia, status\_erro,

resumo, cons\_geral, cons\_categoria, cons\_elemento, le

(3) = ativa\_impressora, avanca\_linha, salta\_pagina, sit\_impressora, cabecalho, printf



(1) = limpa\_linha,  
lenumi,  
rowcol,  
printf,  
mens\_aj,  
mens\_erro,

permsPF,  
mostra\_PF,  
atrib,  
PF

(2) = printf,  
avanca\_linha,  
salta\_pagina,  
posicao\_corr,  
linhas\_rest,  
situacao\_impresora

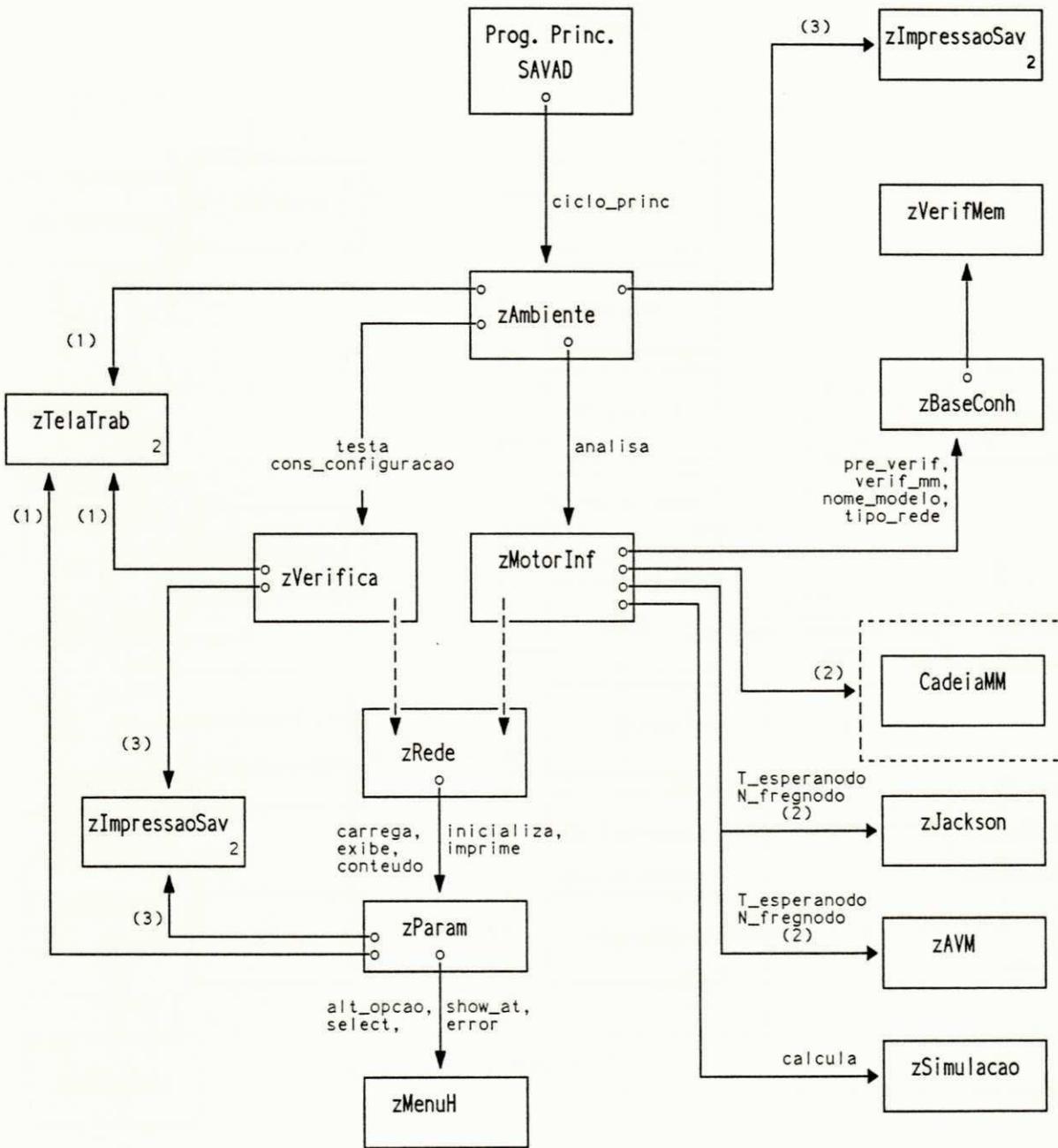
(3) = error,  
alt\_opcao,  
show\_at,  
select

(4) = incluir\_elem,  
alterar\_elem,  
retirar\_elem,  
copla\_elem,  
modifica\_nome\_elem,  
obtem\_parametro,  
cons\_geral,  
cons\_categoria,  
cons\_elemento

DIAGRAMA GERAL CLIENTE-SERVIDOR

VERIFICACAO DE INTEGRIDADE E SOLUCAO DO MODELO

REFERENCIA: 3 / 12



(1) = limpa\_linha,  
rowcol,  
mens\_erro,  
mens\_ajuda,  
printf,  
le\_num1,  
le\_numf,

titulo\_quadro,  
atrib,  
limpa\_tela,  
mostra\_impre,  
le\_nome

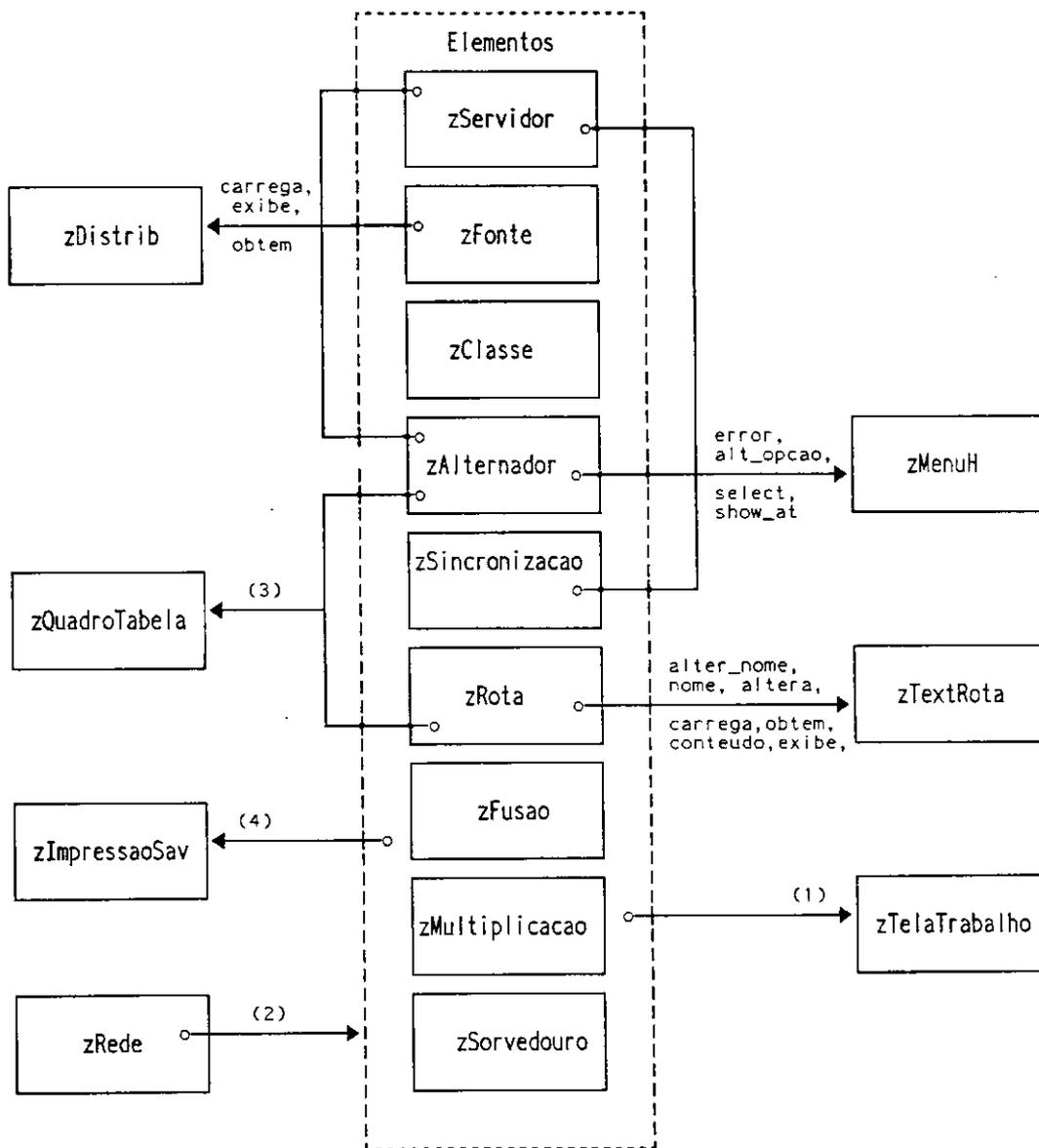
(2) = T\_esperafila,  
T\_respsist,  
N\_fregsist,  
Fator\_utiliz,  
estabilidade,  
error,  
calcula

(3) = printf,  
salta\_pagina,  
ativa\_imprensa,  
situacao\_imprensa,  
avanca\_linha

DIAGRAMA GERAL CLIENTE-SERVIDOR

EXPANSÃO DO SUB-SISTEMA ELEMENTOS

REFERENCIA: 4 / 12



(1) = limpa\_linha,  
rowcol,  
mens\_erro,  
mens\_aj,  
printf,  
lenumi,  
lenumf,  
le\_nome

(2) = nome,  
exibe,  
carrega,  
obtem,  
conteudo,  
altera,  
alter\_nome

(3) = exibe,  
inicializa,  
obtem,  
quant\_elem,  
conteudo\_elem,  
carrega\_elem

(4) = printf,  
avanca\_linha,  
salta\_pagina,  
posicao\_corr,  
linhas\_rest,  
situacao\_imprensa

DIAGRAMA GERAL CLIENTE-SERVIDOR

EXPANSAO DO SUB-SISTEMA CADEIA\_MM

REFERENCIA: 5 / 12

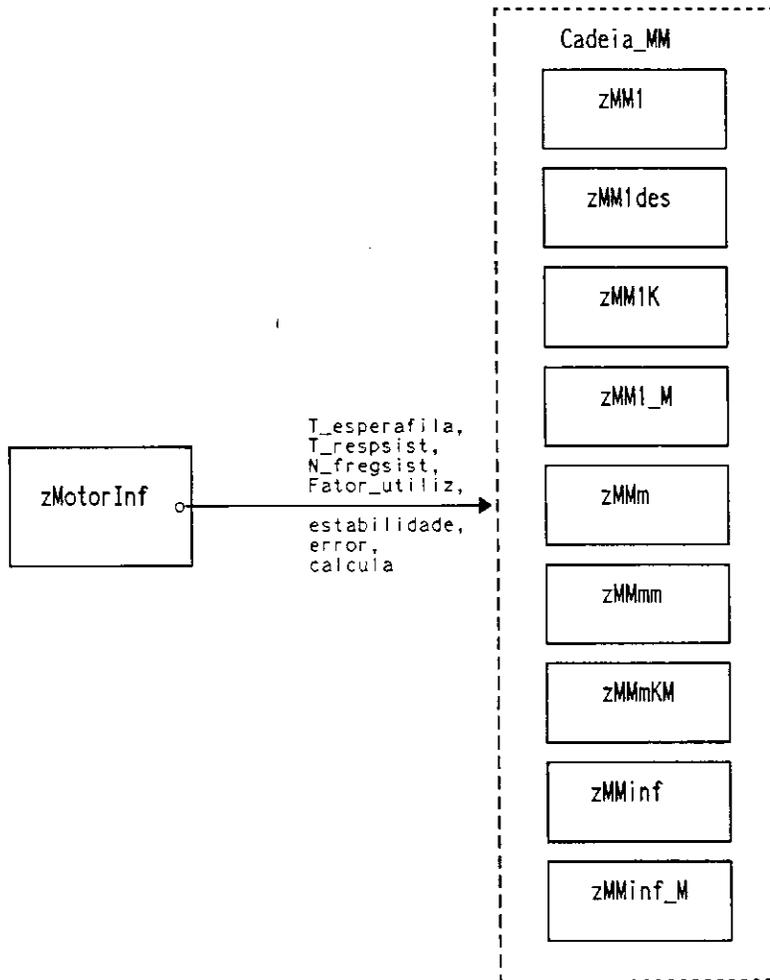


DIAGRAMA GERAL CLIENTE-SERVIDOR

DIAGRAMA DA CLASSE zFilaEvent

REFERENCIA: 8 / 12

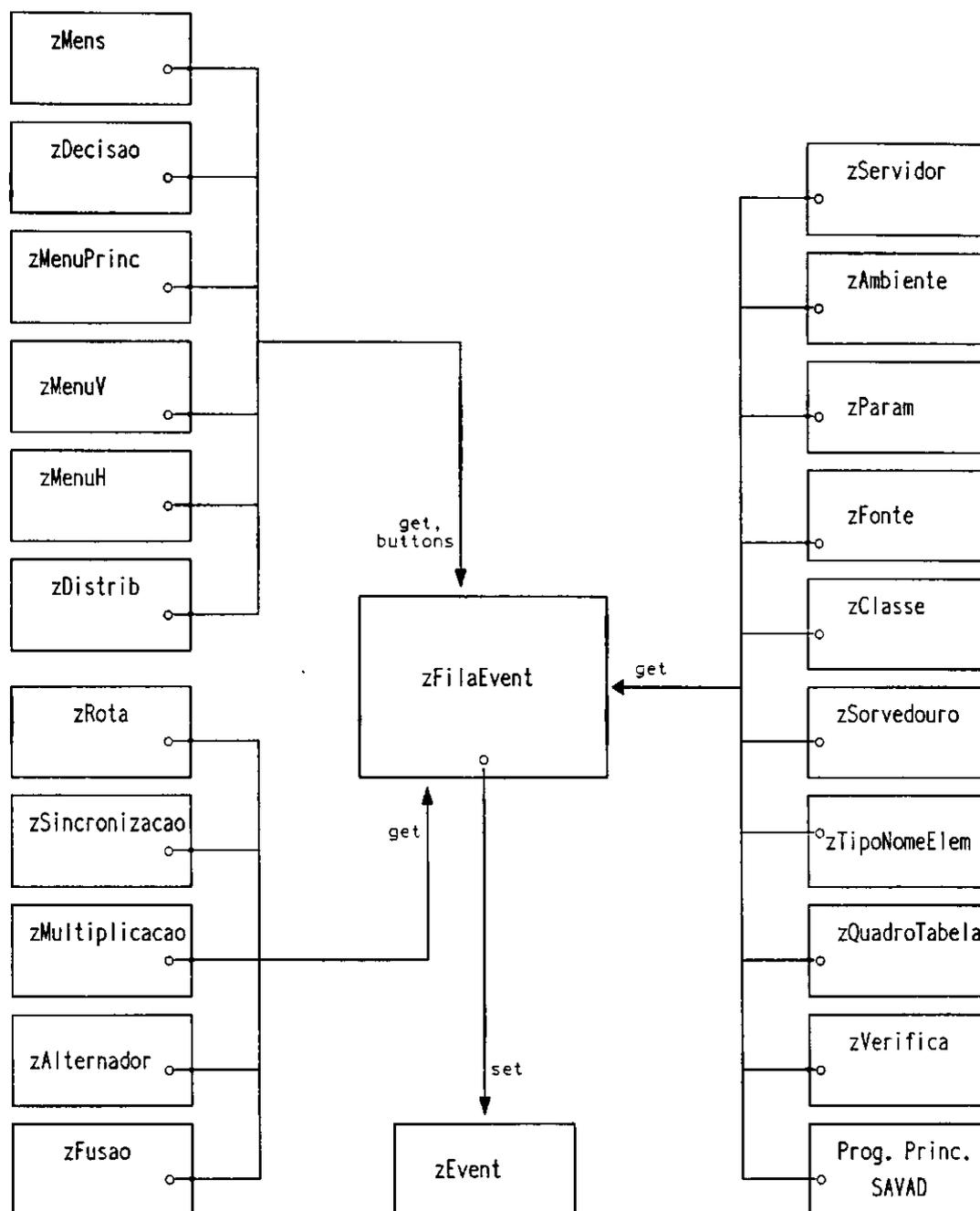
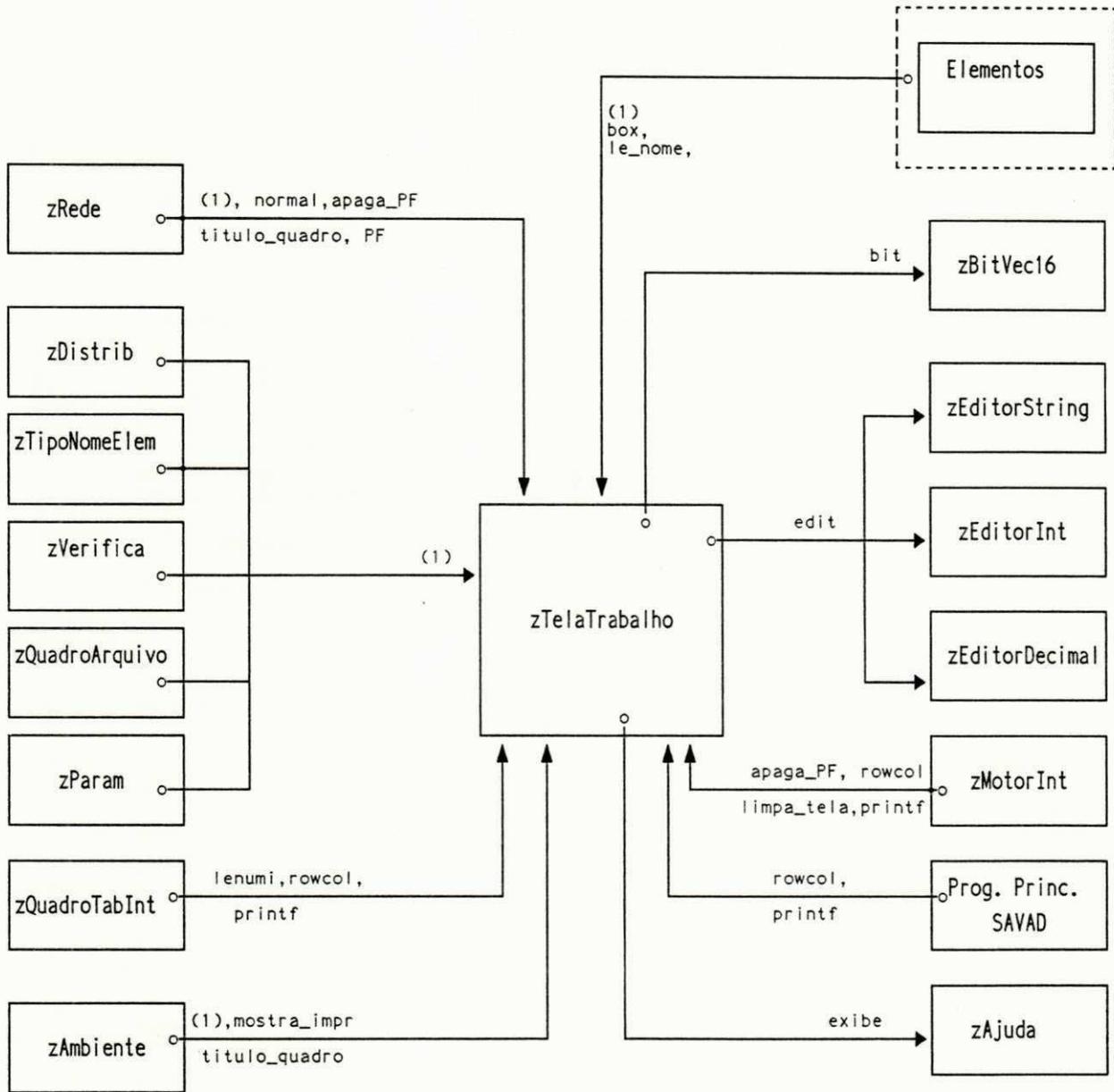


DIAGRAMA GERAL CLIENTE-SERVIDOR

DIAGRAMA DA CLASSE zTelaTrabalho

REFERENCIA: 9 / 12

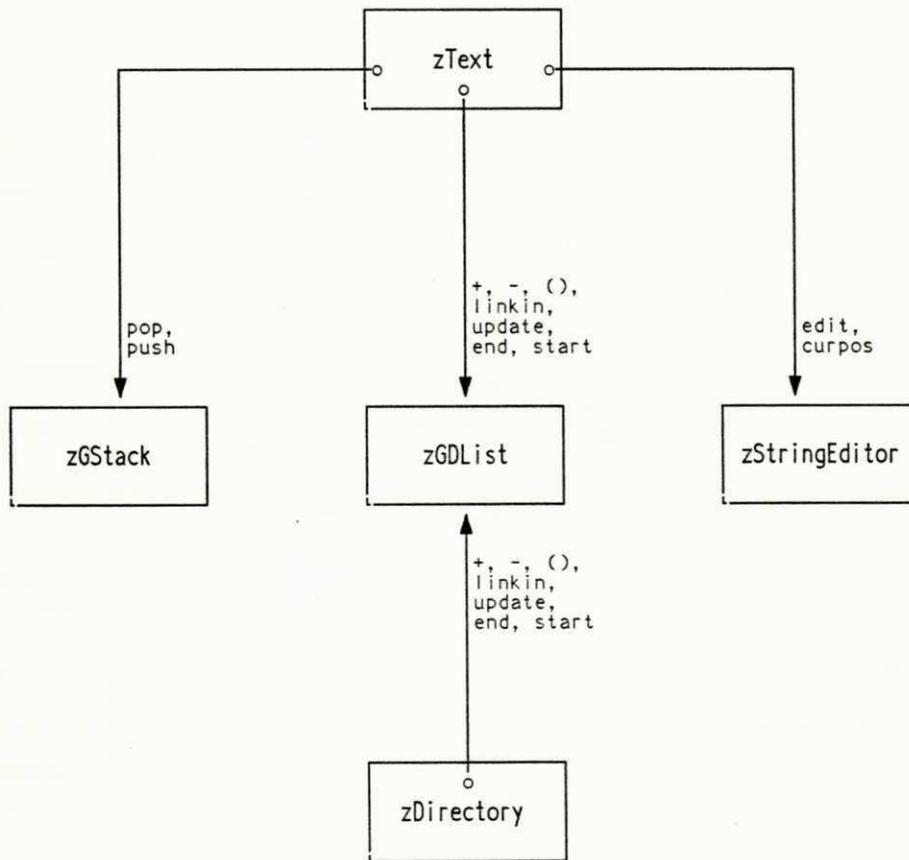


(1) = atrib, mens\_aj,  
rowcol, mens\_erro,  
PF, box,  
printf, PF,  
mostra\_PF, lenumi,  
limpa\_linha, lenumf  
permPFs,

DIAGRAMA GERAL CLIENTE-SERVIDOR

DIAGRAMA DAS CLASSES ASSOCIADAS A CLASSE zText

REFERENCIA: 10 / 12



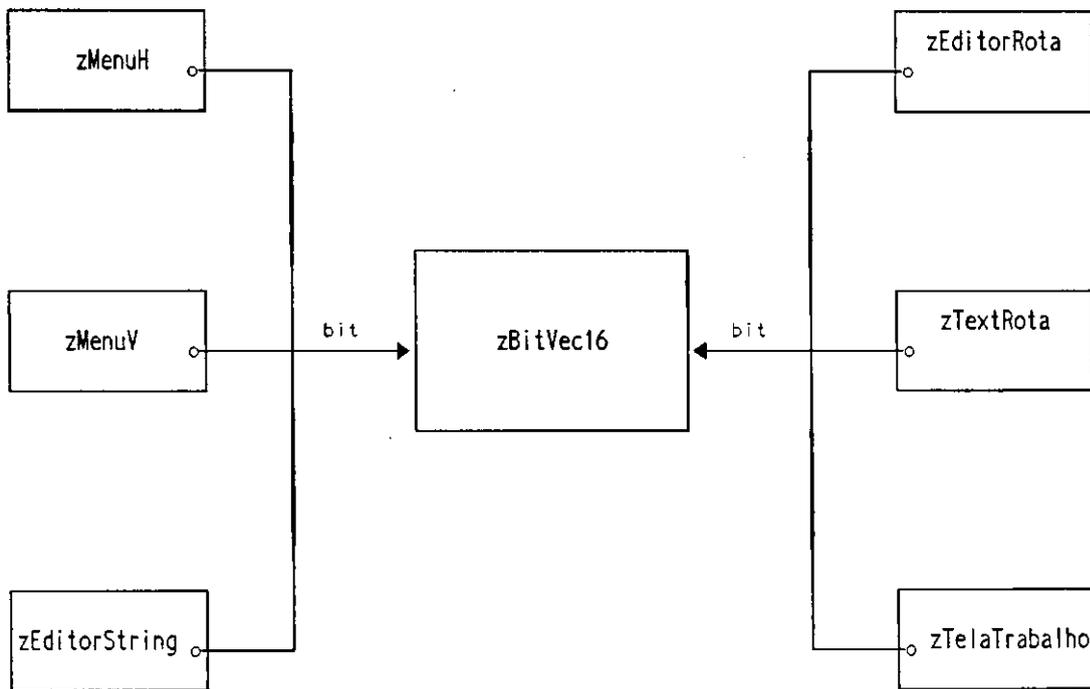
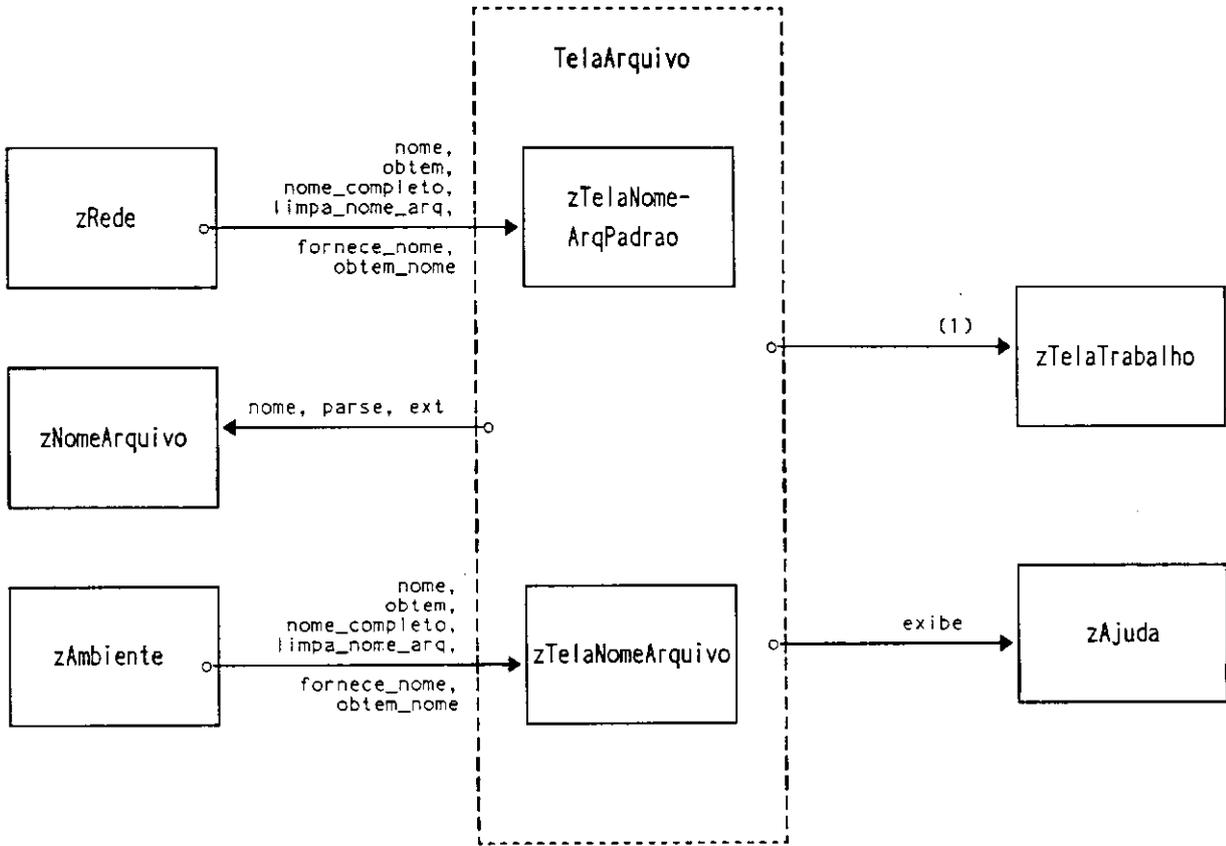


DIAGRAMA GERAL CLIENTE-SERVIDOR

EXPANSÃO DO SUB-SISTEMA "TelaArquivo"

REFERENCIA: 12 / 12



(1) = mens\_aj, mens\_erro, titulo\_quadro, atrib, apaga\_PF, PF, mostra\_PF, limpa\_linha, rowcol, normal, printf,

## APÊNDICE C

### MENSAGENS DE ERRO/AJUDA

#### a) Mensagens de erro na verificação de integridade do modelo

"Erro: As rotas não foram definidas "  
"Erro: A rede não possui elementos definidos "  
"Erro: A rede não possui classes definidas "  
"Erro: Classe da rota não definida - "  
"Erro: Elemento não definido - "  
"Erro: Fila especificada para elemento indevido - "  
"Erro: Número de fila superior ao limite - "  
"Erro: Elemento com entrada multireferenciada - "  
"Erro: Elemento com saída multireferenciada - "  
"Erro: População da rota superior a sua capacidade - "  
"Erro: Elemento/fila não referenciada - "  
"Erro: Elemento com entrada indevida - "  
"Erro: Elemento com saída indevida - "

#### b) Mensagens de erro durante a operação

"Erro: Elemento já existente no modelo de rede"  
"Erro: Informação inválida"  
"Erro: Número informado inválido"  
"Erro: Inconsistência lógica no programa"  
"Erro: Rota incompleta "  
"Erro: Elementos repetidos"  
"Erro: Ultrapassado limite da rota "  
"Erro: Elementos de sequência incompatível com quantidade de filas"  
"Erro: Elementos com nome de tamanho incorreto  
(maior que o permitido ou em branco)"  
"Erro: Memória insuficiente"  
"Erro: Elemento não existente no modelo de rede"  
"Erro: Tipo de elemento inconsistente com nome"  
"Erro: Arquivo não existente"  
"Erro: Nome de arquivo inválido"  
"Erro: Execução de Comando DOS não pode ser efetuada"  
"Erro: Número de servidores superior ao comprimento da fila"  
"Erro: Associação de filas incompatível com comprimento de filas"  
"Erro: Estouro em capacidade de processamento"

"Erro: Opção de menu inexistente"  
"Erro: Especificação de fila incorreta"  
"Erro: Arquivo de verificação não pode ser aberto"  
"Erro: Arquivo de rede com erros! Não pode ser processado"  
"Erro: Quantidade de simulações incorreta. Mínimo: 1 - Máximo: 40"  
"Erro: Nível de confiança inválido "  
"Erro: Número de sementes da tabela difere do número de simulações"  
"Erro: Fator de desânimo inválido "  
"Erro: Nome de elemento inválido"  
"Erro: Encadeamento de elementos na rota inválida"  
"Erro: Percentual de distribuição de fluxo na rota inválido"  
"Erro: Especificação de rota incorreta"

### c) Mensagens de ajuda no contexto durante a operação

"Informe o nome do elemento de rede  
(não incluir '.' no nome)"  
"Informe o valor correspondente  
(número inteiro maior que zero)"  
"Informe o nome da fila"  
"Não existem arquivos deste tipo"  
"Informe: número da fila  
ou número da fila (repetição)  
ou número da fila (\*)"  
"O modelo de rede não possui elementos deste tipo"  
"Informe o comando DOS"  
"Atenção: Módulo ainda não implementado"  
"Informe o valor correspondente (número inteiro maior que 1)"  
"Informe o valor correspondente"  
"Informe o valor correspondente  
(número inteiro maior ou igual a zero)"  
"Informe o valor correspondente  
(quant. de filas igual a 1 ou 2)"  
"Informe o nome da classe correspondente"  
"Para retornar ao SAVAD digite 'EXIT' "  
"Informe o valor correspondente (número inteiro)"  
"Informe a especificação do(s) arquivo(s)"  
"Informe o nível de confiança  
(10, 20, 40, 50, 60, 80, 90, 95, 98, 99, 99.9)"  
"Informe o número de simulações a executar  
(mínimo: 1 - máximo: 40)"  
"Informe o fator de desânimo (acima de 0.0)"

- "Informe a população da rede:  
Se existir . . : Inteiro maior que zero  
Se não for definida: Zero"
- "A simulação digital vai ser iniciada  
(tecle "ENTER" para continuar)"
- "Simulação digital concluída  
(tecle "ENTER" para continuar)"
- "Solução do modelo não disponível"

## APÊNDICE D

### LIMITES UTILIZADOS NA IMPLEMENTAÇÃO DO SAVAD

Número máximo de nós na rede - Teorema de Jackson:	80
Número máximo de nós - AVM:	60
Número máximo de redes - AVM:	20
Número máximo de elementos em uma rota:	250
Número máximo de simulações de cada vez:	40
Comprimento dos nomes dos elementos:	16
Número máximo de entradas em um escalonador cíclico:	70

# APÊNDICE E

## MANUAL DO USUÁRIO

**UNIVERSIDADE FEDERAL DA PARAÍBA**  
**CENTRO DE CIÊNCIAS E TECNOLOGIA**  
**DEPARTAMENTO DE SISTEMAS E COMPUTAÇÃO**  
**COORDENAÇÃO DE PÓS-GRADUAÇÃO EM INFORMÁTICA**

**SAVAD: SISTEMA DE AVALIAÇÃO DE DESEMPENHO DE  
MODELOS DE REDES DE FILAS**

**MANUAL DO USUÁRIO**

**Campina Grande - PB**

**Novembro/1993**

## SUMÁRIO

1. INTRODUÇÃO.....	E-1
1.1 Organização do Manual.....	E-2
2. MODELAGEM DE REDES DE FILAS .....	E-3
2.1 Elementos do SAVAD .....	E-3
2.2 Determinação da Solução do Modelo .....	E-9
3. INTERFACE DO SAVAD.....	E-10
3.1 Descrição Geral da Interface.....	E-10
3.2 Ativação do SAVAD.....	E-11
3.3 Teclas Especiais e Procedimentos Padrões.....	E-12
3.4 Opções do Menu Principal .....	E-13
3.5 Descrição das Opções do SAVAD.....	E-15
3.5.1 Arquivo .....	E-15
3.5.2 Edição .....	E-19
3.5.3 Consulta .....	E-25
3.5.4 Verificação .....	E-28
3.5.5 Processamento .....	E-29
3.5.6 Opções/Comandos MS-DOS .....	E-32
3.5.7 Sair.....	E-33
4. ESPECIFICAÇÃO DE UM MODELO DE REDE.....	E-34
5. FORMATO DOS DADOS DO SAVAD .....	E-39
5.1 Estação de Serviço .....	E-39
5.2 Fonte.....	E-42

5.3 Sorvedouro .....	E-43
5.4 Ponto de Multiplicação .....	E-43
5.5 Ponto de Fusão .....	E-44
5.6 Ponto de Sincronização .....	E-44
5.7 Ponto Escalonador .....	E-45
5.8 Rota .....	E-48
5.9 Classe .....	E-50
6. INSTALAÇÃO DO SAVAD .....	E-51
6.1 Equipamento .....	E-51
6.2 Instalação .....	E-51
6.3 Operação .....	E-52
6.4 Arquivos .....	E-52

## 1. INTRODUÇÃO

O Sistema de Avaliação de Desempenho de Modelos de Redes de Filas (SAVAD) constitui-se em um sistema especialista para avaliação de modelos de redes de filas. Exemplos de sistemas que podem ser modelados dessa forma são redes de computadores, sistemas de tráfego, processos industriais, agências bancárias, etc. SAVAD escolhe e executa, eficientemente, a técnica mais adequada para a solução do modelo proposto pelo usuário (analítica ou numérica) e fornece as medidas de desempenho relevantes solicitadas. O SAVAD dispõe de diversas facilidades para simplificar a tarefa do usuário na especificação do seu modelo proporcionando um ambiente integrado, amigável e inteligente além de fornecer segurança, rapidez e confiabilidade no processamento do modelo.

O usuário do SAVAD necessita apenas conhecimentos elementares de modelagem e avaliação de desempenho de sistemas de redes de filas.

Utilizando menus e janelas, o usuário pode:

- Especificar cada um dos elementos componentes do modelo.
- Testar a integridade do modelo descrito.
- Especificar se o modelo deve ser processado utilizando algoritmos baseados na Teoria das Filas ou utilizando a Simulação Digital. Se a técnica for Simulação Digital permite ao usuário fazer a parametrização necessária.
- Submeter o modelo para a determinação de sua solução.
- Consultar o modelo, o resultado de sua verificação de integridade ou sua solução.
- Fazer operações com arquivos (copiar, eliminar, exibir diretório, etc.).
- Ter acesso ao sistema operacional para executar qualquer comando.

O SAVAD é um sistema de uso simples e que oferece ajuda ao usuário, sensível ao contexto, fornecendo auxílio para o item em que o usuário está operando a cada instante.

## 1.1 Organização do Manual

O manual está dividido em 6 capítulos, estruturados da seguinte forma:

**Capítulo 1:** Apresenta uma visão geral do manual.

**Capítulo 2:** Descreve os elementos que permitem compor um modelo de rede de fila a ser submetido ao SAVAD.

**Capítulo 3:** Apresenta uma visão geral da interface, fornecendo as convenções a serem usadas no resto do manual. São descritos alguns procedimentos comuns. Descreve as opções do SAVAD, mostrando a funcionalidade de cada uma delas.

**Capítulo 4:** Mostra os passos a serem seguidos pelo usuário para a especificação de um modelo e sua solução.

**Capítulo 5:** Apresenta o formato dos campos usados para a especificação dos elementos do modelo de redes de filas.

**Capítulo 6:** Descreve os procedimentos para a instalação do SAVAD bem como os requisitos de software e hardware necessários.

## 2. MODELAGEM DE REDES DE FILAS

Uma rede de filas é um sistema com múltiplas filas interconectadas de acordo com uma topologia, operando assíncrona e concorrentemente. Em uma rede de filas, clientes (ou fregueses) ficam em uma fila de algum nó até que sejam atendidos de acordo com uma disciplina especificada, passando em seguida para o nó seguinte da rede. A população da rede corresponde ao número de clientes que transitam em um dado instante pelos diversos nós da rede.

Para a modelagem das redes de filas são utilizados elementos que descrevem o comportamento do problema a ser analisado. Apresentamos a seguir os elementos utilizados pelo SAVAD na modelagem de redes de filas.

### 2.1. Elementos do SAVAD

Os elementos do SAVAD possuem flexibilidade e facilidade suficientes para permitir a modelagem de forma sofisticada.

Temos os seguintes elementos:

#### a) Clientes

Clientes são entidades temporárias que circulam através dos elementos (nós) do modelo solicitando serviços.

O caminho a ser seguido por um cliente é denominado *rota*. Associado a um cliente existe uma *classe*. Clientes de uma mesma classe possuem os mesmos atributos. As classes permitem que clientes com comportamentos diversos possam ser caracterizados e analisados de formas distintas.

Clientes são componentes virtuais do modelo na medida em que não são especificados explicitamente pelo usuário. Informações sobre clientes são obtidas durante o processamento através dos demais elementos do modelo.

Podemos caracterizar como clientes mensagens que trafegam em uma rede de computadores, peças em uma linha de montagem ou pessoas que procuram atendimento em uma agência bancária.

#### b) Estações de Serviço

Estações de serviço, representam os recursos de um modelo de redes de filas. Podemos citar como estações de serviço canais de transmissão em uma rede de computadores, caixas atendendo clientes em uma agência bancária, etc.

Uma estação de serviço pode ter um ou mais servidores e apenas um cliente é atendido de cada vez por um servidor. Os clientes chegam a uma estação de serviço de acordo com um processo de chegada. Ao chegar à estação, se algum servidor estiver livre, o cliente será atendido, caso contrário entrará em uma fila de espera.

Os intervalos entre chegadas de clientes sucessivos em uma estação de serviço são determinados por uma Função de Distribuição de Probabilidade (FDP) do tipo:

- Exponencial
- Uniforme
- Normal
- Determinística
- Geral (definida pelo usuário através de coordenadas de uma dada Função de Distribuição de Probabilidade).

São permitidos os seguintes tipos de Estações:

- Servidor simples (um único servidor).
- Servidor múltiplo (diversos servidores).
- Servidor infinito (número ilimitado de servidores).

A fila dos servidores simples e dos servidores múltiplos pode ter comprimento limitado ou não. Servidores infinitos não têm filas, isto é, sempre que um cliente chega à estação de serviço, há um servidor disponível para atendê-lo.

O atendimento de clientes é determinado por uma disciplina de escalonamento que pode ser de um dos seguintes tipos:

- FCFS ("First Come, First Served")
- LCFS ("Last Come, First Served")
- "Header of Line"
- Tempo
- Aleatória

Uma estação de serviço pode ter uma só entrada ou várias. Este último caso pode ocorrer quando a estação for compartilhada por várias rotas ou no caso de uma rota com realimentação.

A estação de serviço pode dispor de uma única saída ou de várias. Caso existam várias saídas, as probabilidades de envio de clientes para cada uma delas podem ser iguais ou diferentes. Se as probabilidades de passagem de clientes forem diferentes, devem ser especificadas pelo usuário, respeitando-se o valor 1 (um) para sua soma. Caso sejam iguais não necessitam ser especificadas.

### c) Fontes

Fontes geram clientes que circulam em modelos de redes abertas. Os clientes são gerados segundo uma Função de Distribuição de Probabilidade similar àquelas disponíveis para as estações de serviço (exponencial, uniforme, determinística, normal e geral).

### d) Sorvedouros

Sorvedouros são elementos que eliminam clientes em modelos de redes abertas.

### e) Pontos de Controle

Pontos de controle são elementos que permitem controlar o fluxo de clientes nos modelos de redes. Esses elementos foram projetados para fornecer maior flexibilidade e facilidade na construção do modelo. Os pontos de controle oferecidos pelo SAVAD são os seguintes:

- 1. Ponto de multiplicação:** Multiplica um único cliente que chega a este nó em vários. Pode possuir uma entrada e "n" saídas. Pode ser usado para modelar pacotes de dados com tipos diferentes de informação que devem receber tratamentos distintos no receptor tais como "pacotes de dados" e "reconhecimento". Também pode ser usado para representar um demultiplexador.
- 2. Ponto de fusão:** Faz a fusão de dois ou mais clientes em um único cliente. Pode possuir "n" entradas e uma saída. Pode ser usado para modelar uma multiplexação de informações. No receptor a mensagem será reconstituída e entregue aos seus destinatários.

Os clientes que serão fundidos devem aguardar nas diversas entradas para que a fusão ocorra de forma simultânea para todos os clientes, gerando um novo cliente na saída.

- 3. Ponto de sincronização:** Destina-se a bloquear clientes até que uma condição seja satisfeita. Para isto o número de clientes em cada uma das suas filas de entrada deve atingir um limite especificado pelo usuário.

Um ponto de sincronização pode possuir as seguintes configurações:

- a) Duas entradas e duas saídas correspondentes.
- b) Duas entradas e uma saída associada a uma das entradas.
- c) Uma entrada e uma saída correspondente.

No primeiro caso a liberação de clientes para as saídas ocorrerá desde que a quantidade de clientes nas duas filas de entrada atinja os limites especificados para essas filas, simultaneamente.

No segundo caso, a liberação de clientes para a saída ocorrerá apenas a partir de uma das filas de entrada e desde que a quantidade de clientes nas duas filas de entrada atinja os limites especificados nas duas filas, simultaneamente.

No último caso, a liberação de clientes para a saída desejada dependerá do número de clientes e do limite especificado para a fila de entrada.

Podem existir as seguintes situações, de acordo com os limites especificados para as filas de entrada ("n"):

- Liberação de "n" clientes. A especificação do número de clientes é feita usando um *inteiro positivo*.
- Liberação de *até "n"* clientes. A especificação do número de clientes é feita usando um *inteiro negativo*.
- Passagem *livre* de clientes na outra saída. A especificação é feita usando "0" (*zero*).

**4. Ponto escalonador:** Permite o escalonamento de clientes de acordo com uma disciplina estabelecida pelo usuário. Um ponto escalonador pode possuir uma ou mais filas. Os clientes de uma mesma fila devem ser de mesma classe. As filas podem ter comprimento limitado ou não. A liberação dos clientes de uma fila obedece a uma disciplina especificada. Se uma fila selecionada para liberar clientes estiver vazia o direito de liberação passa para a próxima fila.

O escalonamento de filas ocorre conforme uma das disciplinas:

- **Cíclica:** As filas são escalonadas conforme uma sequência contendo seus números associados. Após a seleção da última fila da sequência o escalonamento volta para a primeira fila. As filas podem ser referenciadas diversas vezes dentro da sequência, aumentando dessa forma sua prioridade ou obedecendo a um controle desejado.

A disciplina de liberação de clientes nas filas pode ser feito de forma:

- a) Não exaustiva (somente um cliente é liberado em cada fila).
  - b) Limitada (até "n" clientes são liberados em cada fila).
  - c) Exaustiva (Todos os clientes são liberados).
- **Aleatória:** As filas são escalonadas de forma aleatória, podendo liberar um cliente de cada vez.
  - **Livre:** Neste caso qualquer fila que não estiver vazia pode tentar liberar um cliente. Se mais de uma fila tentar liberar um cliente ao mesmo tempo ocorre uma colisão e estas filas ficam bloqueadas por um período de tempo determinado por uma Função de Distribuição de Probabilidade especificada pelo usuário.

**f) Classe**

As classes são usadas para determinar tipos diferentes de clientes em circulação no modelo. Associada a cada classe existe uma prioridade (inteiro não negativo). Como exemplos de classes diversas em um mesmo sistema podemos citar em uma rede de computadores, pacotes de dados e pacotes de reconhecimento.

**g) Rota**

As rotas estabelecem os diversos caminhos existentes na rede por onde os clientes circulam. O modelo de rede é definido pelo estabelecimento de rotas com classes especificadas unindo os diversos elementos. O roteamento no SAVAD é fixo.

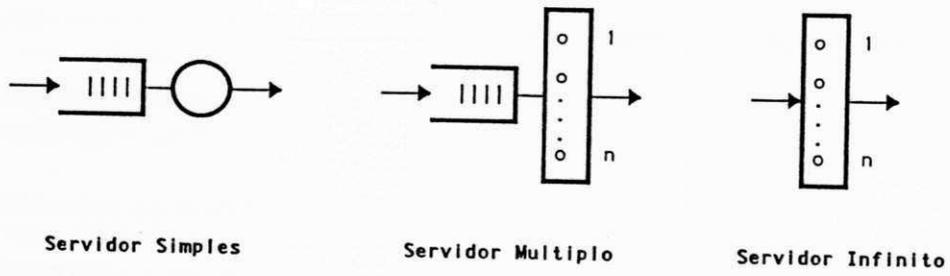
Associadas às rotas existem classes de clientes de modo que possam ser dados tratamentos diferenciados para os diversos tipos de clientes.

Rotas podem ser abertas ou fechadas. Rotas abertas necessitam de elementos que gerem e eventualmente eliminem clientes. Rotas Fechadas possuem um número limitado de clientes em circulação. Rotas podem possuir ou não realimentação.

Clientes de rotas distintas podem compartilhar elementos.

No SAVAD uma rota é especificada através da definição da sequência dos seus elementos constituintes, na ordem desejada.

Na figura 2.1 temos os símbolos utilizados para a representação gráfica dos elementos do SAVAD.



a) Tipos de Estações de Serviço

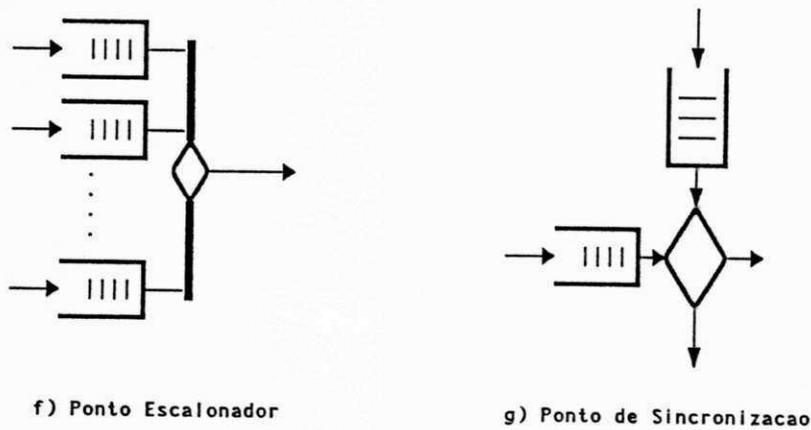
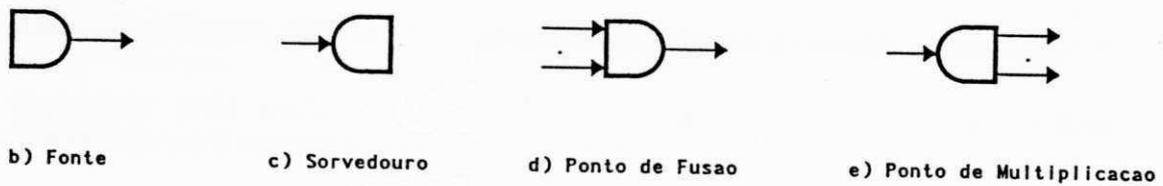


Fig 2.1 - Elementos de Modelagem do SAVAD

## 2.2 Determinação da Solução do Modelo

O processamento de um modelo de redes de filas para a determinação das medidas de desempenho com o auxílio do SAVAD é executado seguindo as seguintes etapas:

1. O usuário representa o problema existente em um modelo de filas utilizando os elementos de modelagem do SAVAD.
2. Os elementos do modelo são descritos individualmente para o SAVAD.
3. O SAVAD faz a verificação da integridade do modelo especificado.
4. O modelo é solucionado. O SAVAD procurará a técnica de solução mais adequada, verificando inicialmente a possibilidade de utilização de técnicas analíticas. Caso não exista solução analítica, o SAVAD submete o modelo à simulação digital. Opcionalmente, o usuário pode submeter o modelo à simulação digital mesmo que exista solução analítica.
5. Apresentação para o usuário, das medidas de desempenho do modelo solucionado.

O usuário pode analisar o modelo com outras configurações, para verificar o seu comportamento em diversas situações.

### 3. INTERFACE DO SAVAD

A comunicação entre o usuário e o SAVAD processa-se através de sua interface (tela e teclado). O SAVAD oferece uma interface flexível, inteligente e amigável, o que permite ao usuário facilmente especificar modelos de redes de filas.

Apresentamos a seguir uma descrição da interface do SAVAD.

#### 3.1 Descrição Geral da Interface

A interface é composta basicamente de uma janela principal onde são feitas todas as operações, um menu principal e um menu auxiliar com as teclas de PF's ativas no rodapé da tela (figura 3.1). O usuário, de forma interativa, pode descrever o seu modelo, escolher a técnica desejada para solucionar o modelo e receber as medidas de desempenho do seu processamento. Durante a descrição do modelo, a interface, automaticamente, valida cada um dos elementos especificados, e, antes de solucionar o modelo, valida-o globalmente. A interface oferece facilidades tais como manipulação de arquivos e acesso direto aos comandos do sistema operacional (MS-DOS).

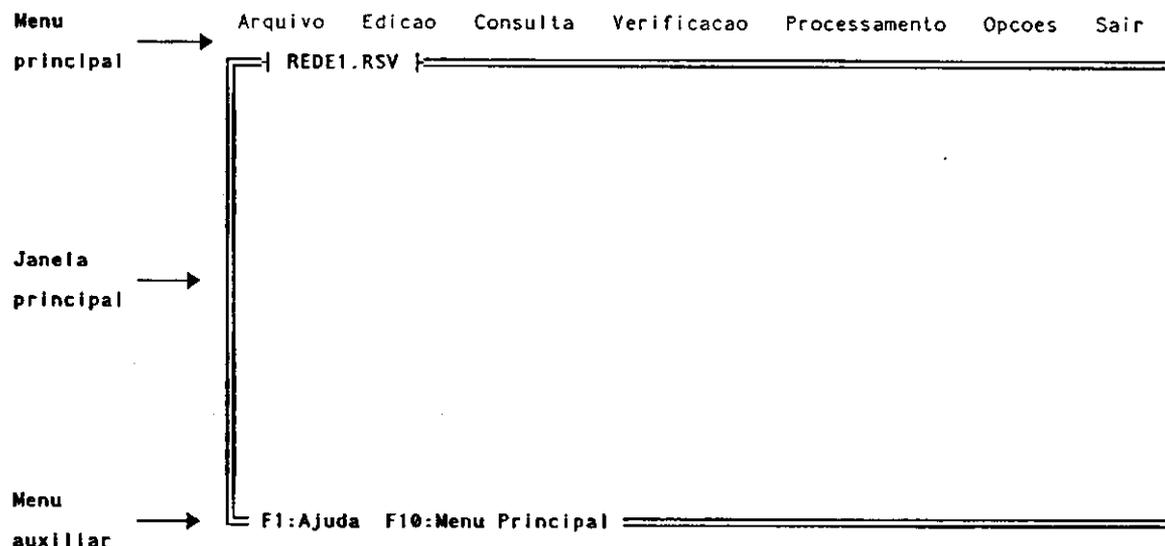


Fig. 3.1 - Interface Básica

Existe disponível em todo o SAVAD um mecanismo de ajuda sensível ao contexto ("help on-line") acionando-se <F1> (figura 3.2).

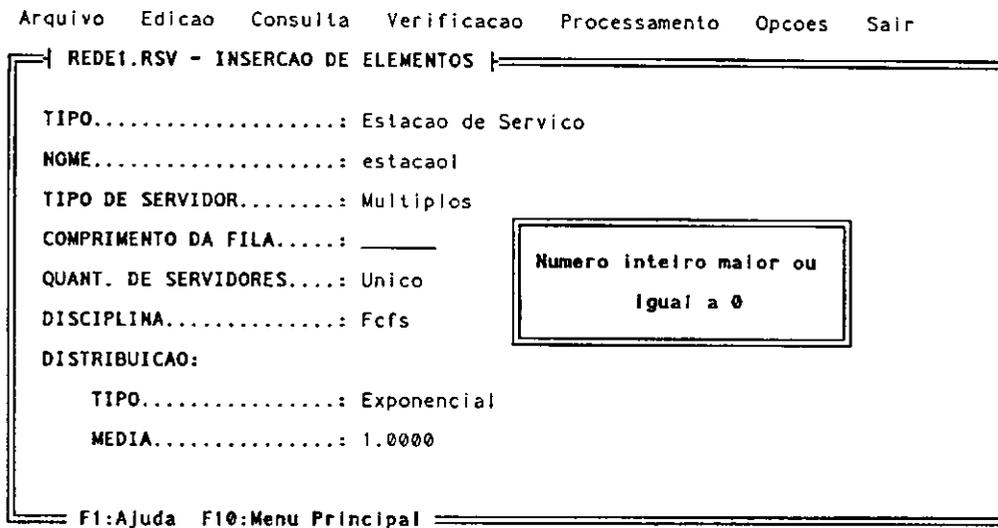


Fig. 3.2 - Exemplo de Ajuda Sensível ao Contexto

### 3.2 Ativação do SAVAD

O SAVAD pode ser ativado de uma das formas abaixo:

a) Digitar: *SAVAD nome\_do\_arquivo\_de\_rede* <Enter>

Se o arquivo com o modelo de rede não existe, o SAVAD pergunta ao usuário se este deseja criar um novo arquivo. Caso o usuário não queira criar um novo arquivo, será criado um arquivo de trabalho padrão de nome "SEMNOME.RSV".

Se o arquivo solicitado existir, o SAVAD passa a operar com ele.

b) Digitar: *SAVAD* <Enter>

Carrega o SAVAD e torna disponível para uso um arquivo padrão de trabalho, de nome "SEMNOME.RSV".

Caso necessário, o usuário poderá, a partir do menu principal, especificar o arquivo de modelo de rede a ser utilizado.

### 3.3 Teclas Especiais e Procedimentos Padrões

Teclas especiais e procedimentos operacionais utilizados para a edição de campos:

#### a) Cursor no campo sendo editado:

##### **Campo com opções:**

Quando o campo estiver sendo editado será apresentado um menu horizontal com todas as opções disponíveis.

Na primeira vez que o este campo do elemento de modelagem estiver sendo editado, será destacada a opção padrão. Nas demais vezes, será destacada a opção feita anteriormente.

Quando o campo não estiver sendo editado, será apresentado no vídeo, a opção feita (ou a padrão).

##### **Campo a ser digitado:**

Na inclusão de um elemento de rede no modelo, os campos são apresentados inicialmente com seus valores padrões. Em caso de alteração de um elemento de rede do modelo, o SAVAD apresenta os valores anteriores. Os campos podem ser editados e modificados.

O conteúdo do campo será validado imediatamente e, em caso de erro, emitida uma mensagem ao usuário, esperando pela redigitação do novo valor.

#### b) Tecla ↑:

Volta para o item anterior.  
Equivalente à tecla <ESC>.

#### c) Tecla ↓:

Vai para o item seguinte.  
Equivalente à tecla <Enter> quando acionado em um menu horizontal ou durante a edição de um campo.

#### d) Tecla <Enter>:

Vai para o item seguinte.  
Se o campo não tiver sido preenchido adequadamente, apresenta mensagem de erro e espera pela redigitação de um novo valor.

#### e) Tecla → ou ← :

Modifica opções em curso em menu horizontal  
Vai para o campo seguinte/anterior se estiver no final/início do campo.

**f) Tecla <ESC>:**

Cancela entrada ou alterações sendo feitas no campo do elemento da rede sendo editado. Volta para o campo imediatamente anterior.

**g) Tecla <F1>:**

Fornecer ajuda para o preenchimento do campo, sensível ao contexto.

**h) <F10>:**

Vai para o menu principal.

Cancela a edição do elemento que está sendo editado. ←

As opções de <PF> ou teclas especiais ativas estarão listadas no rodapé da tela.

A edição do elemento da rede é concluída teclando-se <Enter> no último campo do elemento.

Caso a edição do elemento não seja concluída, por qualquer motivo, as alterações no (ou a criação do) elemento serão desconsideradas.

### 3.4 Opções do Menu Principal

O menu principal, sempre visível ao usuário, fornece-lhe as opções básicas do SAVAD. É a partir deste menu que o usuário pode comandar novas ações, as quais, ao serem concluídas ou interrompidas, fazem o SAVAD retornar ao menu principal.

O item do menu principal passível de ativação é mostrado em vídeo reverso. Para selecionar a função desejada, o usuário usa as teclas de ← e → para se posicionar no menu e acionar <Enter>. Uma outra forma é teclar diretamente a letra em destaque no item desejado do menu.

Associado ao menu principal existe um menu secundário com sub-funções associadas às funções do menu principal. A ativação de cada menu é similar à do menu principal, usando-se as teclas de ↑ e ↓ para se deslocar no menu (figura 3.3).

Pode-se voltar para o menu principal de qualquer ponto do SAVAD, acionando-se a tecla <F10> ou acionando-se <ESC> tantas vezes quantas forem necessárias. Neste caso a operação em execução é cancelada.

Teclando-se → ou ←, o SAVAD vai para o item seguinte do menu principal mostrando o sub-menu vertical correspondente.

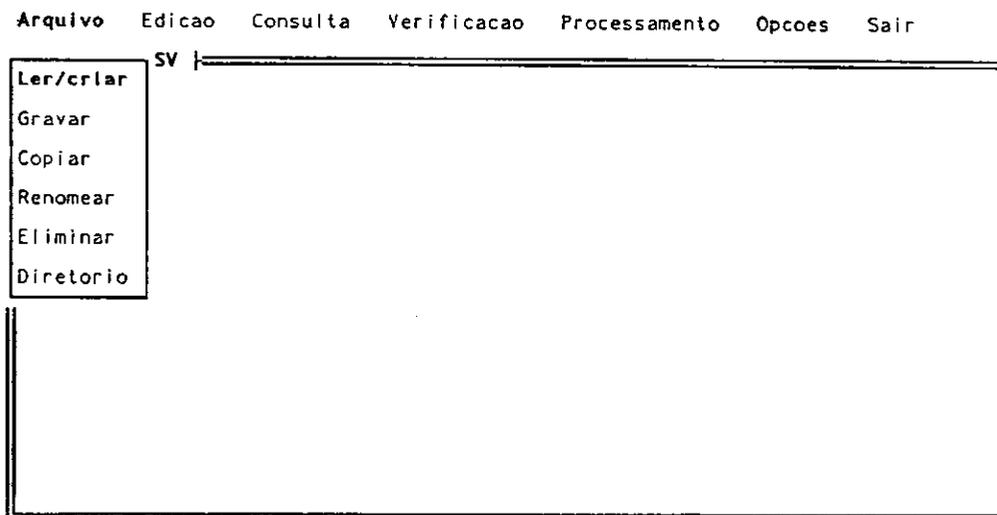


Fig. 3.3 - Menu Principal/Secundario

De qualquer ponto do programa, para seu término, deve-se retornar ao menu principal e acionar a opção **Sair** ou teclar <ESC>.

O menu principal oferece as seguintes opções:

- **Arquivo:** Manipulação dos arquivos de descrição de modelos.
- **Edição:** Especificação e alteração dos elementos do modelo.
- **Consulta:** Consulta/impressão das características do modelo, resultado da verificação de integridade do modelo e suas medidas de desempenho.
- **Verificação:** Teste da integridade do modelo especificado.
- **Processa:** Solução do modelo de redes.
- **Opções/DOS:** Permite ao usuário executar diretamente comandos do sistema operacional (MS-DOS).
- **Sair:** Salva eventuais alterações efetuadas no modelo e encerra o processamento do SAVAD (retornando para o nível do sistema operacional).

Na seção seguinte vamos analisar mais detalhadamente cada uma das opções existentes.

### 3.5 Descrição das Opções do SAVAD

Apresentamos a seguir uma descrição de cada uma das funções disponíveis no SAVAD.

#### 3.5.1 - Arquivo

As operações definidas são:

- Ler e carregar (para uso do SAVAD) um arquivo com o modelo de filas já existente ou criar um novo arquivo.
  - Gravar o arquivo com o modelo com as alterações feitas.
  - Copiar um arquivo com o modelo sendo utilizado.
  - Eliminar arquivos existentes no disco.
  - Renomear arquivos existentes no disco.
  - Mostrar o diretório desejado do disco.
- 
- **Ler/criar**

Torna disponível para análise ou atualização um arquivo com um modelo especificado. Caso o arquivo especificado não exista, será criado um novo arquivo (figura 3.4).

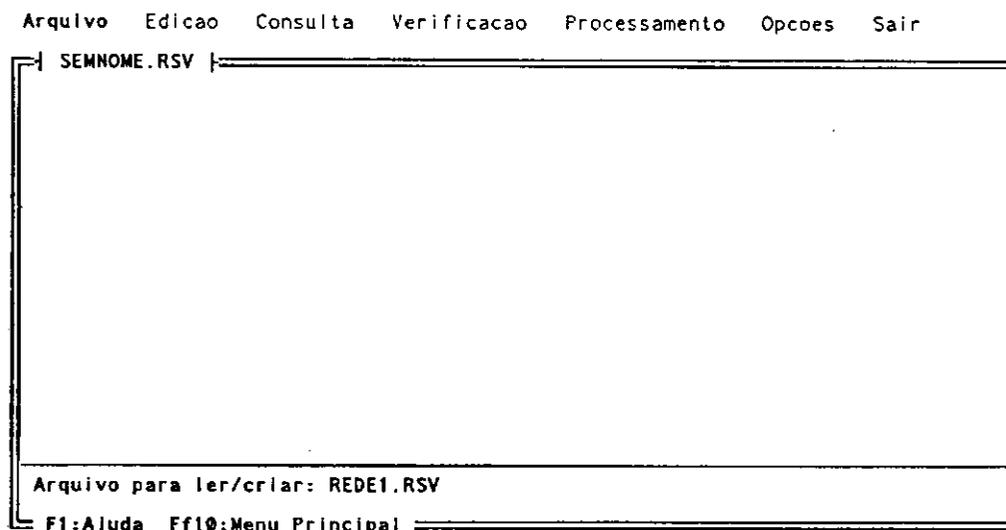


Fig. 3.4 - Criação/leitura de um Arquivo com um Modelo de Rede

Se o usuário teclar <F1> será apresentado um quadro com a relação de nomes de arquivos de rede (com extensão ".RSV") existentes. Neste caso o usuário poderá selecionar diretamente dentre os arquivos apresentados no quadro o arquivo desejado, conforme mostrado na figura 3.5.

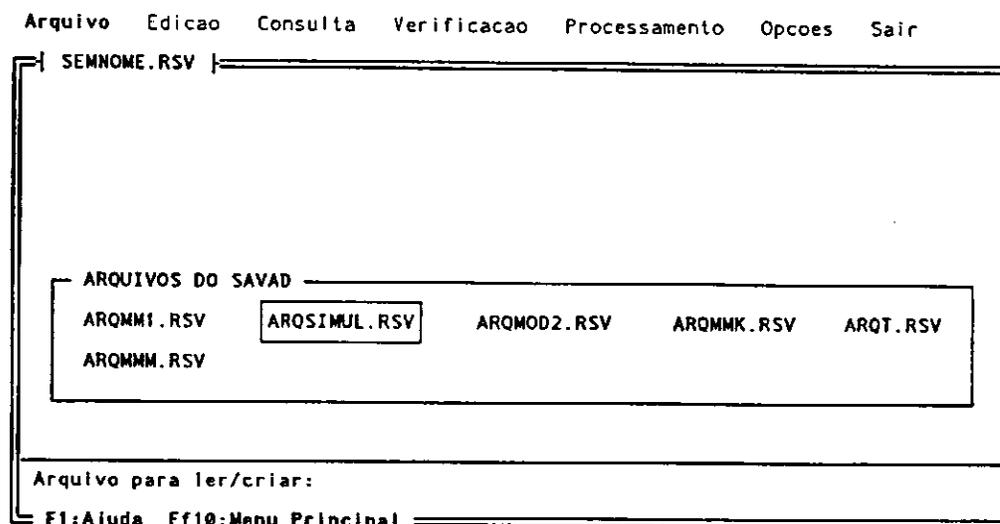


Fig. 3.5 - Criacao/leitura de um Arquivo com um Modelo de Rede, Selecionando a partir de tabela com os arquivos existentes (F1)

• Gravar

Grava em arquivo, no disco, o modelo correntemente em especificação e/ou alteração.

Caso não tenham ocorrido alterações no arquivo desde a última gravação, a gravação não é efetuada (não é necessária).

Esta operação deve ser feita a intervalos regulares, para evitar perdas de dados provocadas por falha de equipamento, disco cheio ou falta de energia elétrica.

- **Copiar**

Copia o arquivo com o modelo em uso para um outro arquivo.

Se um arquivo com o nome a ser gravado já existe no disco, o SAVAD pergunta ao usuário se deseja efetuar a gravação, substituindo o arquivo anterior.

O usuário deve informar o nome do arquivo de destino (figura 3.6).

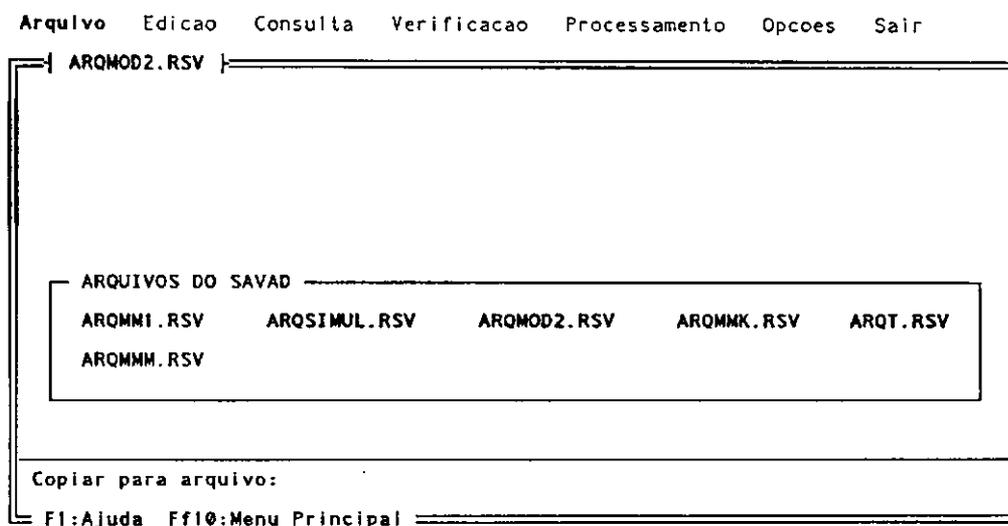


Fig. 3.6 - Cópia de um Arquivo com um Modelo de Rede, Mostrando a Relação dos Arquivos de Modelo Existentes (F1)

Se teclar <F1> será apresentado um quadro com relação de nomes de arquivos de rede existentes.

- **Renomear**

Faz a alteração do nome do arquivo sendo utilizado.

O usuário deve informar o novo nome do arquivo. Se o arquivo informado estiver correntemente em uso o SAVAD não efetua qualquer ação.

Se o usuário teclar <F1> será apresentado um quadro com a relação de arquivos de rede existentes.

- **Eliminar**

Permite ao usuário eliminar arquivos de qualquer tipo existente no disco (não necessariamente arquivos do tipo "RSV").

O usuário deve informar o nome do arquivo a ser cancelado. O arquivo a ser eliminado não pode ser o arquivo correntemente em uso.

Se o usuário teclar <F1> será apresentado um quadro com a relação de arquivos existentes no diretório corrente. O arquivo pode ser selecionado dentre os apresentados no quadro de maneira similar à da opção *LER/criar*.

- **Diretório**

Fornece ao usuário uma relação de arquivos existentes de acordo com o padrão especificado pelo usuário.

Podem ser usados os qualificadores "?" e "\*", bem como analisados diretórios diferentes do corrente (figura 3.7).

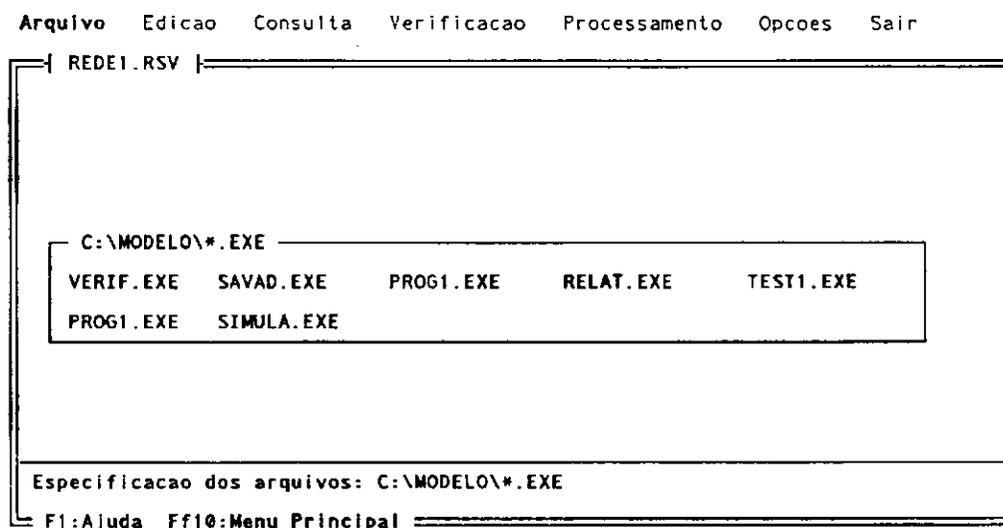


Fig. 3.7 - Relacao de Arquivos do Tipo "\*.EXE" Existentes no Diretorio C:\modelo

### 3.5.2 - Edição

Permite a especificação e atualização dos elementos de modelagem.

A edição consiste basicamente das operações:

- a) Inserção de novos elementos no modelo.
- b) Alteração de elementos já existentes no modelo.
- c) Eliminação de elementos existente no modelo.
- d) Cópia de elementos do modelo.
- e) Modificação do nome de elementos do modelo.

As operações são feitas sempre no arquivo correntemente em uso. Se nenhum arquivo tiver sido lido ou criado pelo usuário, as operações serão efetuadas no arquivo padrão de trabalho, "SEMNOME.RSV". Posteriormente este arquivo padrão poderá ser gravado com este ou com outro nome.

Os procedimentos básicos de operação da edição são comuns a todo o sistema, funcionando da forma descrita anteriormente.

Todas as funções da edição possuem um conjunto de mecanismos de operação comuns. Estas operações dizem respeito à seleção do tipo do elemento a ser operado e do seu nome.

Após a seleção da função de edição desejada, é apresentado um menu com os tipos de elementos disponíveis conforme figura 3.8.

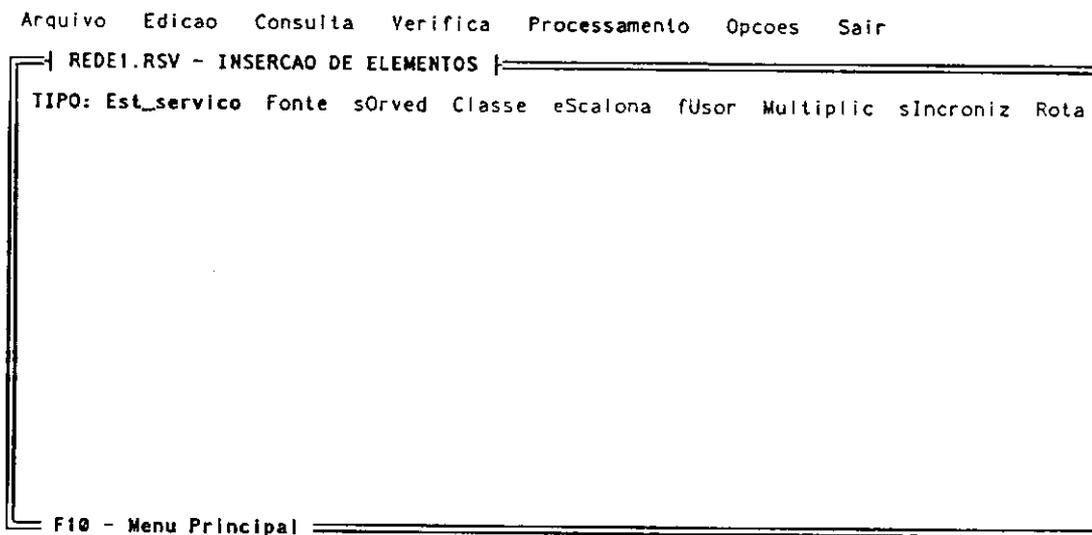


Fig. 3.8 - Tipos de Elementos

Após a seleção do tipo de elemento a ser editado, o usuário deve informar o nome do elemento do modelo (figura 3.9).

```
Arquivo  Edicao  Consulta  Verificacao  Processamento  Opcoes  Sair
REDE1.RSV - ALTERACAO DE ELEMENTOS
TIPO.....: Estacao de Servico
NOME.....: _
ELEMENTOS NA REDE
SERV1  LINHA_COM1  LINHA_COM2  SERV_3
CONTR_A  CONTR_B
F1:Ajuda  F10:Menu Principal
```

Fig. 3.9 - Edicao/Alteracao de um Elemento do Tipo Estacao de Servico  
Mostrando a Relacao de Elementos ja' Existentes no Modelo (F1)

O nome do elemento pode ter até 16 caracteres alfanuméricos e não deve incluir ponto (".") ou espaços.

Se o usuário teclar <F1>, será apresentado um quadro com a relação dos nome dos elementos existentes na rede, do tipo especificado. Com exceção da operação de inserção, o elemento de modelagem pode ser selecionado diretamente dentre os existentes no quadro.

Após a conclusão da edição de um elemento, o SAVAD solicita informar um outro elemento do mesmo tipo que o anterior.

Caso se queira editar elementos de um outro tipo, deve-se retornar para o menu de tipo de elemento teclando <ESC> ou ↑ e fazer a seleção do tipo de elemento desejado.

Ao final da operação o usuário pode voltar ao menu principal, teclando <ESC>, ↑ ou <F10>.

Uma descrição detalhada dos formatos e opções dos campos dos elementos de modelagem é fornecida no capítulo 5.

Estes procedimentos são comuns a todas as funções de edição descritas a seguir.

• **Inserir Elementos**

Permite acrescentar novos elementos a um modelo.

São apresentados os campos do elemento criado, com valores padrão, permitindo ao usuário a digitação dos seus valores definitivos (figura 3.10). Se o elemento informado já existir na rede, o SAVAD apresenta mensagem de erro.

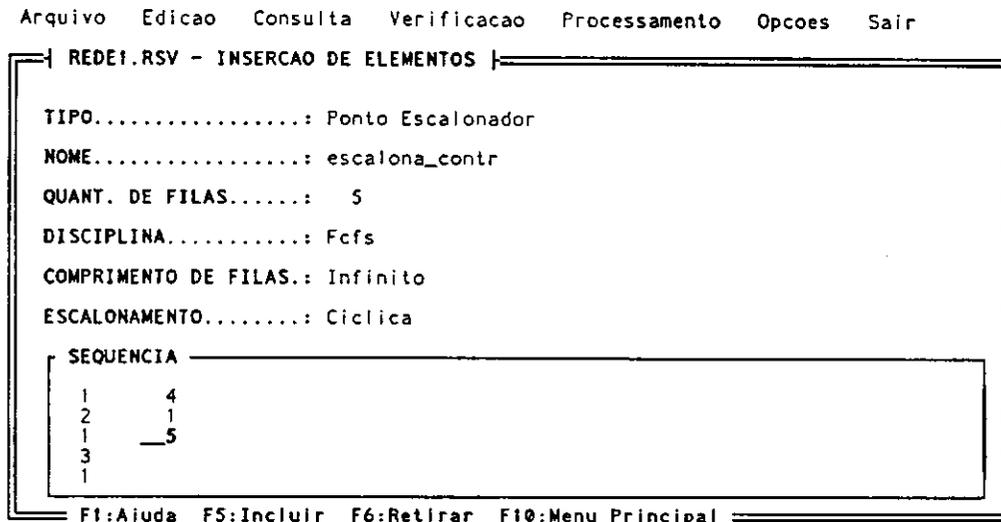


Fig. 3.10 - Edicao/Insercao de um Elemento do Tipo Ponto Escalonador

Digitando-se <ESC> ou ↑ em qualquer dos campos, o SAVAD retorna para o item anterior. Se o cursor estiver no primeiro campo, o SAVAD sai da edição deste elemento.

Digitando-se <F1> será apresentada uma mensagem de ajuda, sensível ao contexto, correspondente ao campo sendo editado.

Digitando-se →, ←, ↑ ou ↓ pode-se caminhar entre os diversos campos do elemento do modelo.

Se o campo a ser editado tiver um número limitado de opções, será apresentado um menu horizontal, com a opção padrão em destaque. Após a seleção, o campo exibirá apenas a opção feita

A entrada de dados é concluída após a digitação de <Enter> ou ↓ no último campo do elemento. Neste caso, o elemento é efetivamente criado e incluído na rede. Outras formas de retorno (imediate com <F10> ou campo a campo com ←, ↑ ou <ESC>) não criam o elemento.

A entrada de dados é concluída após a digitação de <Enter> ou ↓ no último campo do elemento. Neste caso, o elemento é efetivamente criado e incluído na rede. Outras formas de retorno (imediate com <F10> ou campo a campo com ←, ↑ ou <ESC>) não criam o elemento.

• **Alterar Elementos**

Permite alterar elementos existentes de um modelo.

Os elementos da categoria "sorvedouro" não podem ser alterados pois não têm campos, sendo constituídos apenas de seus nomes. Caso necessário, estes elementos, podem ser eliminados do modelo ou ter seus nomes modificados.

Após o usuário informar o nome do elemento a ser alterado, são apresentados os campos com os valores existentes possibilitando ao usuário sua atualização (figura 3.11).

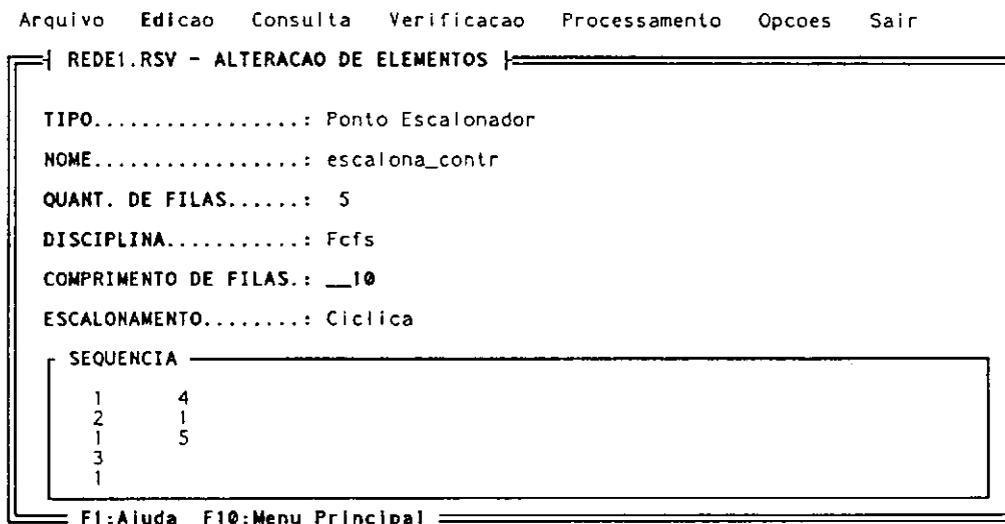


Fig. 3.11 - Edicao/Alteracao de um Elemento do Tipo Ponto Escalonador

O procedimento de edição é similar ao de inclusão de elementos.

A alteração dos dados é concluída após a digitação de <Enter> ou ↓ no último campo do elemento.

• **Eliminar Elementos**

Permite a eliminação de elementos de um modelo.

São apresentados os campos do elemento a ser cancelado, permitindo ao usuário que os seus valores sejam visualizados. O usuário deve então confirmar o cancelamento do elemento (figura 3.12).

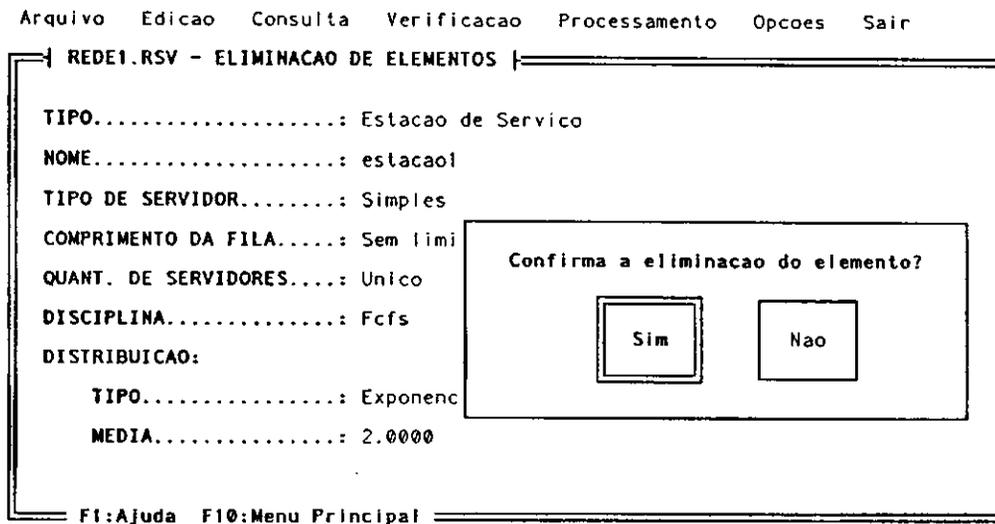


Fig. 3.12 - Edicao/Eliminacao de um Elemento do Modelo de Rede

O elemento será retirado do modelo em caso de confirmação, pelo usuário.

- **Copiar Elementos**

Permite fazer cópia de elementos de um modelo. O elemento criado será uma duplicata do elemento original, com outro nome. Tal facilidade permite que um grande número de elementos similares sejam criados a partir de um elemento básico. Posteriormente estes elementos podem ser alterados, se necessário.

Após a seleção do tipo de elemento da rede a ser copiado, o usuário deve informar o nome do elemento original e o nome do elemento de destino que será criado (figura 3.13).

```
Arquivo  Edicao  Consulta  Verificacao  Processamento  Opcoes  Sair
| REDE1.RSV - COPIA DE ELEMENTOS |
TIPO.....: Estacao de Servico
NOME.....: servi
NOVO ELEMENTO...: _____
F1:Ajuda  F10:Menu Principal
```

Fig. 3.13 - Edicao/Copia de um Elemento do Tipo Estacao de Servico

- **Modificar Nomes de Elementos**

Permite a alteração dos nomes de elementos de um modelo.

Após a seleção do tipo de elemento da rede com nome a ser alterado, o usuário deve informar o nome original do elemento e o nome desejado. O novo nome do elemento não deve existir no modelo.

### 3.5.3 Consulta

Permite a consulta acerca da configuração de um modelo, de seus elementos, do resultado da verificação de integridade ou de sua solução.

O resultado da consulta pode ser apresentado diretamente no vídeo ou impresso.

Para a impressão da consulta, seleciona-se o item "Imprime" antes de solicitar a consulta desejada. Enquanto o modo de impressão estiver ativo, será exibido no rodapé direito da tela a mensagem: "IMPRESSAO". Neste modo, todas as consultas serão impressas.

Para o usuário desativar o modo de impressão, deve-se selecionar novamente no menu de consulta o item "Imprime" (figura 3.14).

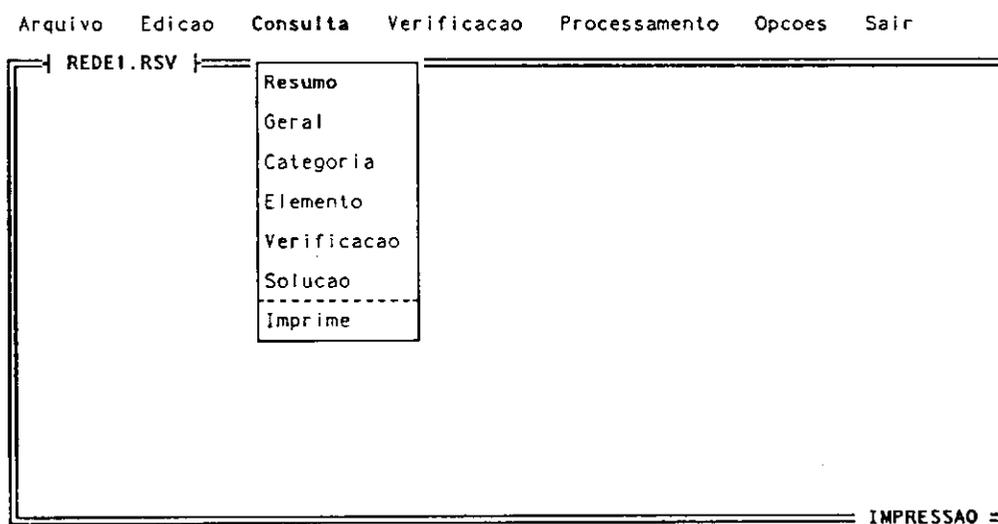


Fig. 3.14 - Menu Principal/Consulta com a Impressao Ativada

• **Resumo**

Apresenta um sumário com os dados consolidados de um modelo, a técnica utilizada na solução bem como os parâmetros em caso de simulação digital (figura 3.15).

```

Arquivo  Edicao  Consulta  Verificacao  Processamento  Opcoes  Sair
-----
| REDE1.RSV - RESUMO |
|-----|
| NOME DA REDE.....: rede1.rsv |
| NUMERO DE ELEMENTOS |
| ESTACAO DE SERVICO....: 5 |
| FONTE.....: 1 |
| SORVEDOURO.....: 2 |
| PONTO ESCALONADOR.....: 1 |
| PONTO DE FUSAO.....: 2 |
| PONTO DE MULTIPLICACAO: 0 |
| PONTO DE SINCRONIZACAO: 1 |
| CLASSES.....: 2 |
| ROTAS |
| ABERTAS.....: 2 |
| FECHADAS.....: 1 |
| SITUACAO.....: A rede foi verifica COM erro |
| TIPO DE PROCESSAMENTO...: Simulacao |
|-----|
| Enter:Continua |
    
```

Fig. 3.15 - Consulta/Resumo das Características do Modelo de Rede.

• **Geral**

São apresentadas (ou impressas), sucessivamente, categoria por categoria, as especificações de todos os elementos de um modelo. A figura 3.16 mostra um exemplo da especificação de uma das fontes do modelo.

```

Arquivo  Edicao  Consulta  Verificacao  Processamento  Opcoes  Sair
-----
| REDE1.RSV - CONSULTA GERAL |
|-----|
| TIPO.....: Fonte |
| NOME.....: gerador_sinais |
| DISTRIBUICAO: |
| TIPO.....: Exponencial |
| MEDIA.....: 2.0000 |
|-----|
| F1:Ajuda  F5:Primeiro  F6:Anterior  F7:Seguinte  F8:Ultimo  F10:Menu Principal |
    
```

Fig. 3.16 - Consulta Geral dos Elementos do Modelo, Apresentando uma das Fontes.

À medida que são apresentados os elementos, o usuário pode iterativamente, consultar os elementos anteriores ou posteriores.

Teclas:

<Home> ou <F5>: Apresenta o primeiro elemento.

↑, ←, <PgUp> ou <F6>: Apresenta o elemento anterior.

↓, →, <PgDn>, <F7> ou <Enter>: Apresenta o elemento seguinte ou encerra a apresentação se for o último elemento.

<End> ou <F8>: Apresenta o último elemento.

<ESC>: Fim de consulta da categoria

<F10>: Vai para o Menu principal

- **Categoria**

São apresentadas (ou impressas), sucessivamente, as especificações de todos os elementos do modelo, pertencentes a uma categoria selecionada pelo usuário.

Após o usuário selecionar o tipo de elemento a ser consultado, é apresentada uma relação com os elementos do tipo especificado existentes no modelo. A seguir são exibidos sucessivamente todos os elementos de rede do tipo selecionado.

À medida que são apresentados os elementos da categoria selecionada, o usuário pode consultar os elementos anteriores ou posteriores, de forma similar à da consulta geral.

- **Elemento**

Apresenta (ou imprime) as informações correspondentes a um elemento especificado de um modelo.

O usuário especifica o tipo e o nome do elemento desejado.

- **Verificação**

Apresenta (ou imprime) o resultado da análise de integridade feita anteriormente no modelo em uso.

Se o modelo não tiver sido verificado previamente (vide "*Menu Principal/Verificacao*"), o SAVAD apresenta mensagem de erro.

## • Solução

Apresenta (ou imprime) a solução de um modelo (medidas de desempenho) em uso que tenha sido calculado anteriormente.

Se o modelo de rede não tiver sido solucionado previamente (vide "Menu Principal/Processamento"), o SAVAD apresenta uma mensagem de erro.

### 3.5.4 Verificação

Faz uma análise acerca da integridade da especificação de um modelo.

Tal verificação se destina a permitir que eventuais falhas existentes no modelo sejam detectadas e corrigidas antes que as rotinas de processamento do modelo sejam ativadas, gerando inconsistências ou ocasionando falhas.

O SAVAD verifica os seguintes itens de um modelo:

- a) Referências de rotas (elementos) não definidas
- b) Elementos de multiplicação, fusão, sincronização, fonte ou sorvedouro não encadeados adequadamente.
- c) Elementos definidos mas não referenciados.
- d) Cadeias abertas sem fonte/sorvedouro.
- e) Somatório das capacidades das filas em uma rota fechada que são inferiores à sua população.

O resultado da verificação de integridade do modelo de rede é armazenado em um arquivo especial, de mesmo nome que o do modelo, com extensão "VSV". Desta forma o resultado pode ser consultado ou impresso posteriormente (vide "*Menu Principal - Consulta Verificação*").

Com base na verificação é gerada uma condição de STATUS, informando se o modelo de rede foi ou não verificado e se tem erros.

Um modelo que tenha sido verificado e que não tenha erros, poderá ser submetido para processamento sem necessitar de outras verificações.

Um modelo de rede que tenha sido verificado e tenha erros, não poderá ser submetido para processamento. O modelo deve ser previamente corrigido.

Ao ser alterado em alguma edição, o modelo passa a ser considerado como não verificado.

Um modelo será sempre verificado antes de ser processado. Se o modelo não tiver sido testado pelo usuário, a verificação será ativada automaticamente antes do início do processamento (figura 3.17).

```

Arquivo  Edicao  Consulta  Verificacao  Processamento  Opcoes  Sair
-----
-> REDE1.RSV - VERIFICACAO
VERIFICACAO DE INTEGRIDADE DO MODELO DE REDE

ROTA EM ANALISE: rotal
-----
Erro: Elemento com entrada multireferenciada - "escalon.1"
Erro: Elemento nao definido - "sinc"
Erro: Popul. da rota superior a sua capacidade - "20 x 11"

ANALISE INDIVIDUAL DOS ELEMENTOS/FILAS
-----
Erro: Elemento/fila nao referenciada - "escalon1" - entrada: 2
Erro: Elemento/fila nao referenciada - "escalon1" - entrada: 3

Fim de verificacao - Erros encontrados: 5

Enter:Continua

```

Fig. 3.17 - Resultado da Verificacao de Integridade do Modelo de Redes

### 3.5.5 Processamento

Possibilita ao usuário solicitar o processamento do modelo bem como fazer a especificação dos parâmetros para a simulação, se for necessário.

Este menu apresenta duas opções:

**Automático:** O SAVAD faz o processamento do modelo de rede determinando suas medidas de desempenho, utilizando técnicas analíticas baseados na Teoria das Filas. Caso não exista solução analítica para o modelo será utilizada Simulação Digital.

**Simulação digital:** O modelo será solucionado utilizando a Simulação Digital.

Antes do processamento, o SAVAD faz uma verificação de integridade do modelo, caso este não tenha sido verificado anteriormente (vide "Menu Principal/Verificação").

Se forem encontrados erros de especificação no modelo, o procedimento é interrompido, devendo o usuário fazer as correções necessárias.

- **Automático**

O SAVAD efetua o processamento do modelo de rede de filas para a determinação das medidas de desempenho. O processamento pode ser feito utilizando algoritmos baseados na Teoria das Filas ou através de Simulação Digital.

Procedimentos utilizados:

- a) O SAVAD efetua uma análise preliminar no modelo para verificar se existe uma solução baseada na Teoria das Filas.
- b) Se existir solução baseada na Teoria das Filas, o cálculo é feito analiticamente.
- c) Em caso contrário, o SAVAD procura efetuar a simulação digital. Nesse caso, o SAVAD solicitará ao usuário os parâmetros necessários para a simulação do modelo, conforme descrito na opção "Simulação digital".

Os resultados com a solução do modelo são armazenados em arquivo especial, de mesmo nome que o do modelo, porém com extensão "SSV". Desta forma o resultado pode ser consultado ou impresso posteriormente através da "Consulta/Solução" (figura 3.18).

```
Arquivo  Edicao  Consulta  Verificacao  Processamento  Opcoes  Sair
| REDE1.RSV - SOLUCAO |
TIPO DE SOLUCAO: Analitica
MODELO DE REDE.: M/M/Infinito
PARAMETROS DE DESEMPENHO:
  Tempo medio de espera em fila.....: 0.5000
  Tempo medio de resposta do sistema..: 1.0000
  Numero medio de fregueses no sistema: 1.0000
  Fator de utilizacao.....: 0.5000
Enter:Continua
```

Fig. 3.18 - Solucao do Sistema M/M/INFINITO

- **Simulação digital**

O SAVAD efetua o processamento do modelo de rede de filas para a determinação das medidas de desempenho através de Simulação Digital, independente da existência de soluções analíticas para o modelo.

O usuário deve informar os parâmetros que determinam o funcionamento da simulação do modelo (figura 3.19).

```
Arquivo  Edicao  Consulta  Verificacao  Processamento  Opcoes  Sair
┌── REDE1.RSV - PARAMETROS PARA SIMULACAO ──┐
│
│ CONDICAO DE TERMINO.....: Numero de amostras
│ TOTAL DE AMOSTRAS.....: 20000
│ NIVEL DE CONFIANCA(Z)....: 90.00
│ ANALISE DE SIMUL.(PARSE): Sim
│ NUMERO DE SIMULACOES.....: ____4
│
│ SEMENTES _____
│
│
│
└── F1:Ajuda  F10:Menu Principal ───┘
```

Fig. 3.19 - Parametros para a Simulacao Digital de um Modelo.

Devem ser informados os seguintes itens:

**a) Condição de término da simulação:**

A simulação pode terminar tendo como condição um número especificado de amostras ou de unidades de tempo simulado.

**b) Número de amostras ou unidades de tempo:**

Deve ser especificado o número de amostras utilizado na simulação. Deve ser inteiro positivo com limite máximo de 99.999.999.

### **c) Nível de confiança**

Deve ser informado o nível de confiança desejado para a simulação. São permitidos os seguintes níveis:

10% , 20% , 40% , 50% , 60% , 80% , 90% , 95% , 98% , 99% e 99.9%

### **d) Análise da simulação (parse)**

O usuário pode especificar se deseja que seja gerado uma análise ("parse") para a simulação digital. Posteriormente o usuário poderá acompanhar a simulação, passo a passo, consultando o arquivo de solução.

### **e) Número de simulações**

O usuário especifica a quantidade de simulações sucessivas a que o modelo deve ser submetido. Podem ser feitas até 40 simulações.

### **f) Sementes**

O SAVAD gera uma quantidade de sementes igual ao número de simulações. O usuário pode retirar, alterar ou incluir novas sementes. O número de sementes deve permanecer igual ao número de simulações. São gerados números pseudo-aleatórios, inteiros e primos inferiores a 32767.

## **3.5.6 Opções/Comandos MS-DOS**

Permite ao usuário executar diretamente comandos do sistema operacional.

O usuário sai do ambiente do SAVAD e passa para o ambiente do MS-DOS, possibilitando a execução de quaisquer comandos desejados.

Para retornar ao SAVAD, digita-se "exit" no ambiente MS-DOS.

### 3.5.7 Sair

Encerra o processamento do SAVAD e retorna ao Sistema Operacional. Acionando a tecla <ESC> no menu principal é equivalente a selecionar esta opção.

Caso o arquivo com o modelo de rede em processamento tenha sofrido alguma alteração e não tenha sido gravado anteriormente, o SAVAD pergunta ao usuário se quer salvá-lo antes de encerrar os trabalhos e se quer realmente encerrar o processamento (figura 3.20).

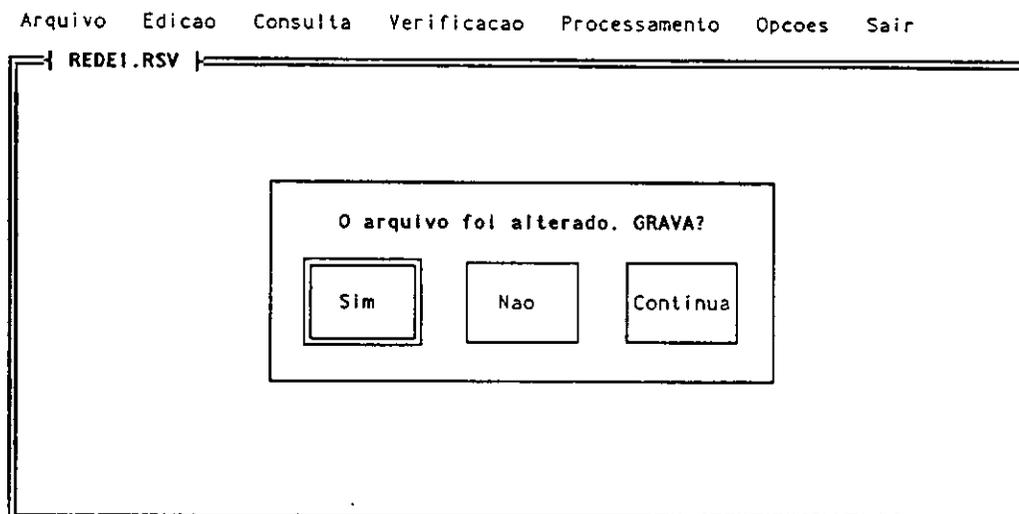


Fig. 3.20 - Encerramento do SAVAD

#### 4. ESPECIFICAÇÃO DE UM MODELO DE REDE

Apresentamos a seguir um exemplo simples de modelagem usando o SAVAD. O exemplo consiste de uma conexão com uma entidade que envia mensagens para outra através de um canal de comunicação. Neste mecanismo não existe reconhecimento para as mensagens recebidas.

Características da entidade emissora:

- Distribuição dos tempos de interchegada de mensagens: Exponencial
- Taxa média de chegada: 1 mensagem por segundo.
- Capacidade de armazenamento: Sem limite.
- Disciplina de escalonamento: FCFS

Características do canal de comunicação:

- Fila sem restrição de comprimento.
- Distribuição do tempo de transmissão: Exponencial
- Taxa média de transmissão: 2 mensagens por segundo
- Disciplina de escalonamento: FCFS

A figura 4.1 mostra o modelo da rede descrita anteriormente:

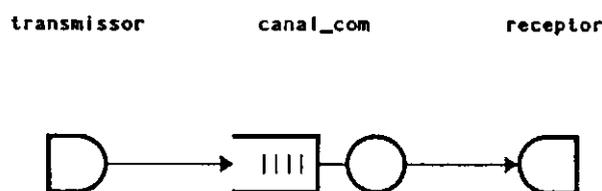


Fig. 4.1 - Modelo da Rede

- A geração de mensagens é feita por uma fonte.
- O canal de comunicação é representado por uma estação de serviço.
- A entidade receptora é representada por um sorvedouro.

Inicialmente, deve-se criar o arquivo com o modelo de rede desejado:

- Selecionar <A>rquivo no menu principal.
- Selecionar <L>er/criar um arquivo no menu secundário.
- Designar o nome do arquivo desejado.

Especificação dos elementos que compõem o modelo de rede: servidor, fonte, sorvedouro, classe e rota.

- Selecionar a opção <E>dicação no menu principal.
- Selecionar a opção <I>nclusão de elemento no menu secundário.
- Selecionar no menu o tipo de elemento a ser incluído no modelo.
- Designar o nome do elemento a ser incluído.
- Informar as características do elemento sendo incluído.
- Proceder da forma acima para cada um dos elementos do modelo (figuras 4.2, 4.3, 4.4, 4.5 e 4.6).

```

Arquivo  Edicao  Consulta  Verificacao  Processamento  Opcoes  Sair
-----
| REDEXEMP.RSV - INSERCAO DE ELEMENTOS |
-----
TIPO.....: Fonte
NOME.....: transmissor
DISTRIBUICAO:
  TIPO.....: Exponencial
  MEDIA.....: 1.0000
-----
F1:Ajuda  F10:Menu Principal
    
```

Fig. 4.2 - Definição da Fonte

```

Arquivo  Edicao  Consulta  Verificacao  Processamento  Opcoes  Sair
-----
| REDE1.RSV - INSERCAO DE ELEMENTOS |
-----
TIPO.....: Estacao de Servico
NOME.....: canal_com
TIPO DE SERVIDOR.....: Simple
COMPRIMENTO DA FILA.....: Sem limite
QUANT. DE SERVIDORES.....: Unico
DISCIPLINA.....: Fcfs
DISTRIBUICAO:
  TIPO.....: Exponencial
  MEDIA.....: 2.0000
FATOR DE DESENCORAJAMENTO: 0.0000
-----
F1:Ajuda  F10:Menu Principal
    
```

Fig. 4.3 - Definição da Estacao de Servico

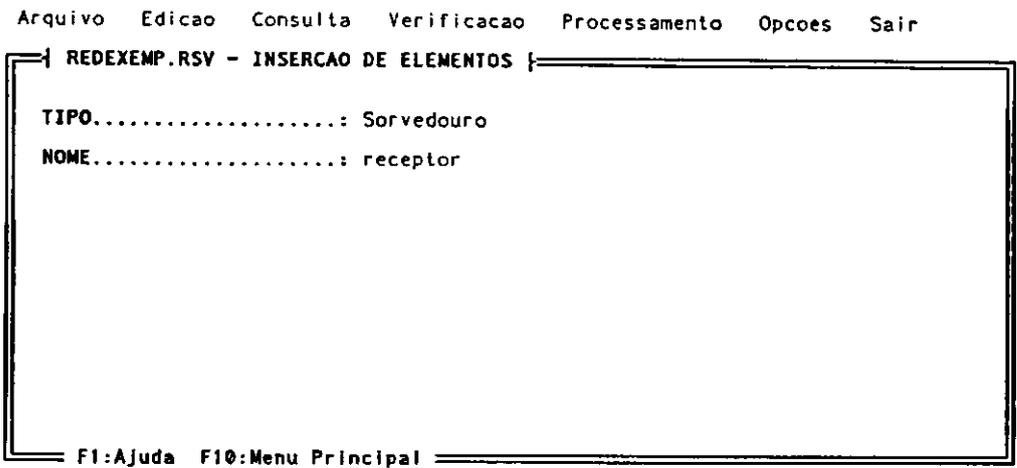


Fig. 4.4 - Definição do Sorvedouro das Mensagens

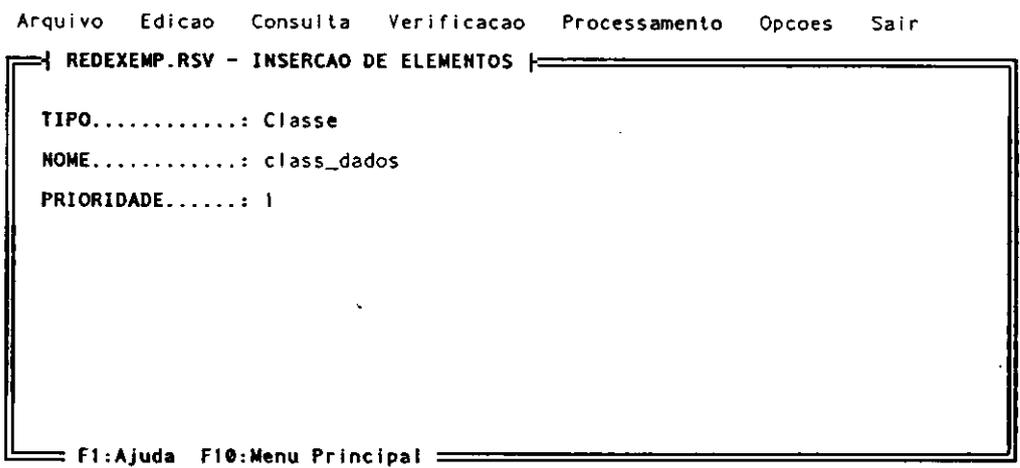


Fig. 4.5 - Definição da Classe dos Clientes que Trafegam no Modelo

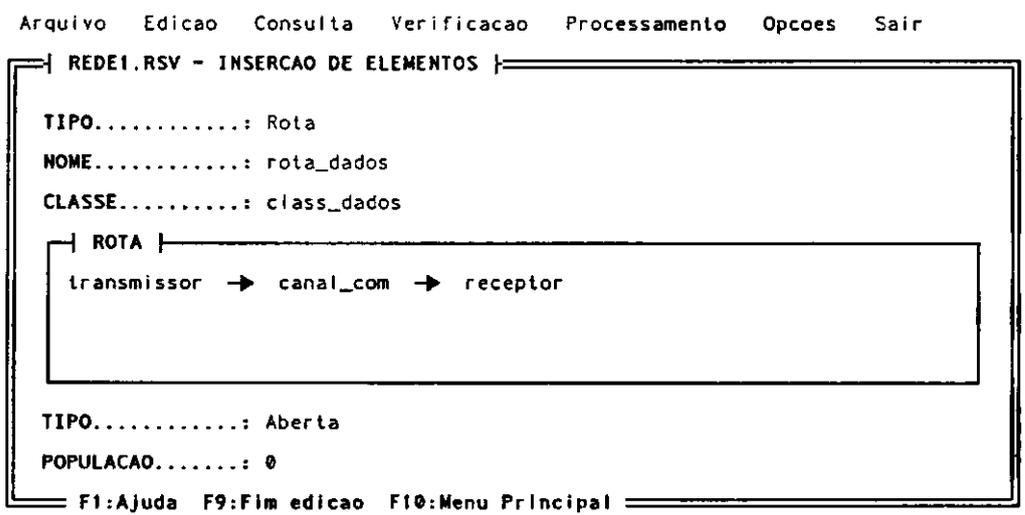


Fig. 4.6 - Definição da Rota do Modelo

Fazer a verificação de integridade do modelo de rede:

- Selecionar a opção <V>**erificação** no menu principal.
- Verificar o resultado. Caso tenha algum erro, deve-se voltar para a fase de edição e, com a opção <A>**ltera elemento** do menu secundário, fazer as correções. Se o modelo não apresentar problemas, pode-se passar para a fase de processamento para a determinação da solução do modelo. O resultado da verificação fica armazenado e pode ser consultado/impresso posteriormente utilizando-se a opção <C>**onsulta**/**<V>erificação**

Fazer o processamento do modelo para a determinação da solução:

- Selecionar a opção <P>**rocessamento** no menu principal.
- Caso o usuário deseje submeter o modelo a uma Simulação Digital, selecionar a opção <S>**imulação digital** no menu secundário e informar como os seus parâmetros. Neste caso isto não será necessário.
- Selecionar a opção <A>**utomático** no menu secundário. Caso a solução tenha sido definida como analítica mas o modelo não possua solução baseada na Teoria das Filas, o SAVAD solicita do usuário os parâmetros e soluciona o modelo usando simulação digital.
- A solução do modelo é apresentada ao usuário. O resultado fica armazenado podendo ser consultado/impresso usando a opção <C>**onsulta**/**<S>olução** (fig. 4.7).

```
Arquivo  Edicao  Consulta  Verificacao  Processamento  Opcoes  Sair
-----
| REDEXEMP.RSV - SOLUCAO |
|
| TIPO DE MODELO DE REDE...: M/M/1
| SOLUCAO.....: Analitica
| MEDIDAS DE DESEMPENHO:
|   Tempo medio de espera em fila.....: 0.500
|   Tempo medio de resposta do sistema.: 1.000
|   Numero medio de fregues no sistema: 1.000
|   Fator de utilizacao.....: 0.500
|
| Enter:Continua
```

Fig. 4.7 - Resultado do Processamento Analítico do Modelo

O modelo de rede pode ser consultado a qualquer momento da forma abaixo:

- Selecionar <C>onsulta no menu principal.
- Caso se deseje que o resultado da consulta seja impresso, seleciona-se <I>mprime no menu secundário. Após esta seleção, a palavra **IMPRESSAO** será exibida no canto inferior direito da tela. Todas as consultas posteriores serão impressas. Para retornar à condição de consulta na tela, deve-se selecionar esta opção novamente.
- Selecionar a opção desejada no menu secundário.

Encerramento do programa:

- Selecione <S>air no menu principal.
- Caso o arquivo com o modelo de rede tenha sido alterado mas não tenha sido gravado, o SAVAD permite que o isto seja feito antes do encerramento dos trabalhos.

## 5. FORMATO DOS DADOS DO SAVAD

O presente capítulo fornece uma descrição detalhada de todos os campos dos elementos utilizados na especificação de modelo de redes de filas, com os formatos, faixas e limites associados.

Nas figuras das telas apresentadas, os campos que têm menu associados serão exibidos com todas as opções disponíveis, com destaque para a opção padrão. O menu horizontal, correspondente a estes campos, só é efetivamente apresentado quando o campo estiver sendo manipulado. Normalmente, o campo mostra apenas a opção que tenha sido feita anteriormente.

### 5.1 - Estação de Serviço

A figura 5.1 apresenta os campos do elemento Estação de Serviço. Os formatos desses campos são descritos a seguir.

```

Arquivo  Edição  Consulta  Verificacao  Processamento  Opcoes  Sair
| REDEI.RSV - INSERCAO DE ELEMENTOS |
|-----|
TIPO.....: Estacao de Servico
NOME.....: nome_estacao
TIPO DE SERVIDOR.....: Simple Multiplo Infinito
COMPRIMENTO DA FILA.....: Sem limite
QUANT. DE SERVIDORES.....: Unico
DISCIPLINA.....: Fcfs Lcfs Hol Tempo Aleatoria
DISTRIBUICAO:
  TIPO.....: Exponencial Uniforme Normal Deterministica Geral
|-----|
|          DADOS COMPLEMENTARES DA DISTRIBUICAO          |
|-----|
FATOR DE DESENCORAJAMENTO: 0.0000
|-----|
F1:Ajuda  F10:Menu Principal
    
```

Fig. 5.1 - Definicao dos Campos de um Elemento do Tipo Estacao de Servico

**a) Tipo de servidor:**

Formato: menu horizontal.

Opção padrão: **Simples**

**b) Comprimento da fila:**

Formato: número inteiro maior ou igual a 0 (zero).

Se TIPO DE SERVIDOR for "Simples" ou "Múltiplo" o valor informado para o comprimento da fila deve ser maior ou igual a 0 (zero).

Se TIPO DE SERVIDOR for "Infinito" o valor não será digitado. Será exibido 0 (sem fila).

Se for digitado 0 (zero) a fila será considerada de comprimento ilimitado.

**c) Quantidade de servidores:**

Formato: número inteiro positivo maior que 0.

Se TIPO DE SERVIDOR for "Simples" a quantidade de servidores não será digitada. SAVAD exibe "Único".

Se TIPO DE SERVIDOR for "Múltiplo" digitar a quantidade de servidores. O seu valor deve ser igual ou inferior ao comprimento da fila.

Se TIPO DE SERVIDOR for "Infinito" a quantidade de servidores não será digitada. SAVAD exibe "Infinito".

**d) Disciplina:**

Formato.: menu horizontal

Opção padrão: **Fcfs**

Se o TIPO DE SERVIDOR for "Infinito" o menu não será exibido.

Se o TIPO DE SERVIDOR for "Múltiplo" mas a quantidade de servidores for igual ao comprimento da fila (não ocorre espera), o menu não será exibido.

**e) Distribuição/Dados complementares:**

Opção padrão: **Exponencial**

- **TIPO = "Exponencial"**

MÉDIA: 9999.999  
 Formato: número real

- **TIPO = "Uniforme"**

LIMITE INFERIOR: 99999.999  
 LIMITE SUPERIOR: 99999.999

Formato: números reais. O limite superior deve ser igual ou maior que o limite inferior.

- **TIPO = "Normal"**

MEDIA: 99999.999  
 VARIANCIA: 99999.999

Formato: números reais.

- **TIPO = "Determinística"**

CONSTANTE: 99999.999

Formato: número real.

- **TIPO = "Geral"**

O usuário descreve a Função Distribuição de Probabilidade através de suas coordenadas. Podem ser especificados até 30 coordenadas. A figura 5.2 apresenta o formato da distribuição do tipo geral.

**DISTRIBUICAO:**

TIPO.....: Geral

NUMERO DE AMOSTRAS: 13

	X	Y		X	Y
4)	99.999999	99.999999	5)	99.999999	99.999999
6)	99.999999	99.999999	7)	99.999999	99.999999
8)	99.999999	99.999999	9)	99.999999	99.999999
10)	99.999999	99.999999	11)	99.999999	99.999999

Fig. 5.2 - Amostras Correspondentes a Distribuicao Geral

**Número de amostras:**

Formato: Número inteiro superior a 0 e igual ou inferior a 30.

Se existirem mais de 8 amostras ocorrerá um rolamento dos dados na tela. Apenas quatro linhas estarão disponíveis a cada momento na tela.

Para ter acesso a qualquer dos números das amostras: usar as setas, ocorrendo rolamento na tela, se necessário. Para encerrar a digitação: teclar <Enter> em *amostra\_x* após o último número da amostra.

**f) Fator de desencorajamento:**

Formato: número real maior que 0.

O campo será solicitado caso a estação de serviço tenha servidor único com comprimento de fila ilimitado.

**5.2 - Fonte**

A figura 5.3 apresenta os campos do elemento Fonte. Os formatos desses campos são descritos a seguir.

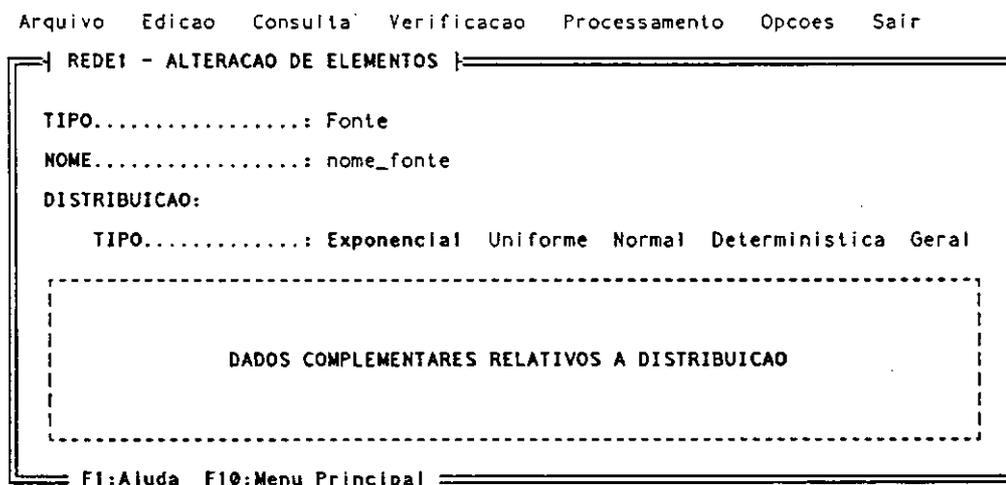


Fig. 5.3 - Alteracao da Fonte em um Modelo

**a) Distribuição:**

Similar àquelas distribuições mostradas na estação de serviço com a diferença de poder exibir até 10 linhas com amostras, em caso de distribuição geral.

### 5.3 - Sorvedouro

O sorvedouro não possui campos específicos além de seu próprio nome (figura 5.4).

```

Arquivo  Edicao  Consulta  Verificacao  Processamento  Opcoes  Sair
-----
| REDE1 - ALTERACAO DE ELEMENTOS |
-----
TIPO.....: Sorvedouro
NOME.....: nome_sorvedouro

-----
F1:Ajuda  F10:Menu Principal
    
```

Fig. 5.4 - Alteracao de Sorvedouro em um Modelo

### 5.4 - Ponto de Multiplicação

A figura 5.5 apresenta os campos do elemento Ponto de Multiplicação. Os formatos desses campos são descritos a seguir.

```

Arquivo  Edicao  Consulta  Verificacao  Processamento  Opcoes  Sair
-----
| REDE1 - ALTERACAO DE ELEMENTOS |
-----
TIPO.....: Ponto de Multiplicacao
NOME.....: nome_pt_multiplic
QUANT. DE SAIDAS.....: 5

-----
F1:Ajuda  F10:Menu Principal
    
```

Fig. 5.5 - Alteracao de Ponto de Multiplicacao em um Modelo

#### a) Quantidade de saídas

Formato: número inteiro maior ou igual a 2





**a) Quantidade de filas:**

Formato: número inteiro maior que 0.

**b) Comprimento da fila:**

Formato: número inteiro maior ou igual a 0 (zero).  
Se digitar 0 (zero) a fila será considerada sem limite.

**c) Disciplina:**

Formato : Menu  
Opção padrão: Fcfs

**d) Escalonamento:**

Formato : Menu  
Opção padrão: Cíclica

Após a especificação do tipo de escalonamento, devem ser fornecidas as seguintes informações:

- **Cíclica**

Devem ser informados as filas que formam a sequência de saídas do escalonador e que serão acionadas sequencialmente.

Formato: *nnnn*, *nnnn(rrr)* ou *nnnn(\*)*,

onde,

*nnnn* - Número da fila do alternador.

*(rrr)* - Fator de repetição da fila.

Número inteiro maior que 0 (zero).

A fila será acionada sistematicamente enquanto tiver mensagens a serem enviadas até o limite especificado.

*(\*)* - Fator de repetição infinito (exaustivo).

A fila será acionada sistematicamente enquanto ainda existirem mensagens para ser enviadas.

O número da fila informada deve ser superior a 0 e igual ou inferior à quantidade de filas especificada anteriormente.

Pode haver repetição de filas intermediárias, de forma a estabelecer prioridades ou controles com sequências definidas.

O número líquido de filas (filas com números de identificação diversas) deve ser igual à quantidade de filas informada anteriormente.

Podem haver no máximo 70 informações no conjunto.

Para terminar a digitação: <Enter> em fila vazia após a última fila digitada (figura 5.9).

ESCALONAMENTO.....: Cíclica

SEQUENCIA		
1	5	2
2	6(*)	8
1	2	
3	7	
4	8(2)	

Fig. 5.9 - Escalonamento Cíclico das Filas

• Aleatória

Esta opção não solicita dados adicionais.

• Livre

Esta opção solicita a informação da Função Distribuição de Probabilidade para acesso às filas (figura 5.10).

ESCALONAMENTO.....: Livre

DISTRIBUICAO:  
TIPO.....: Exponencial Uniforme Normal Deterministica Geral

DADOS COMPLEMENTARES DA DISTRIBUICAO
--------------------------------------

Fig. 5.10 - Escalonamento Livre com a Distribuicao Associada

a) Distribuição:

Similar àquelas distribuições mostradas para a estação de serviço com a diferença do SAVAD poder exibir até 8 linhas com amostras, em caso de distribuição geral.

### 5.8 - Rota

A especificação da(s) rota(s) corresponde à descrição dos "caminhos" que os clientes do modelo de rede de filas vão percorrer. Desta forma os demais elementos do modelo serão encadeados de forma lógica, fornecendo a estrutura ao modelo de rede desejado.

Na descrição, os nomes dos elementos devem ser relacionados em formato livre, separados por espaço, sinal de maior (">") ou vírgula.

Ex: *fonte1 > est\_trab1 > est\_trab2 > sorved*

Em caso de estações de trabalho com múltiplas saídas, deve-se informar as diversas opções de saída entre parênteses, separados por espaço ou por vírgulas. Caso as probabilidades de envio de clientes através das diversas saídas não sejam iguais, deve-se informar o percentual de distribuição de clientes entre cada uma delas antes do nome de cada elemento de destino. A soma dos percentuais, caso informado, deve ser igual a 1,0 (100 %).

Ex: *fonte1 > est\_trab1 > (0.3 est\_trab1, 0.7 sorved)*

Ex: *fonte1 > est\_trab1 > (estaçãox, estaçãoy, estaçãoz)*  
*estaçãox > est\_trab1*  
*estaçãoy > sorved*  
*estaçãoz > est\_test1 > est\_test2 > est\_trab1*

Após a conclusão da edição o SAVAD faz uma formatação da rota especificada de forma a facilitar a visualização e operação (figura 5.11).

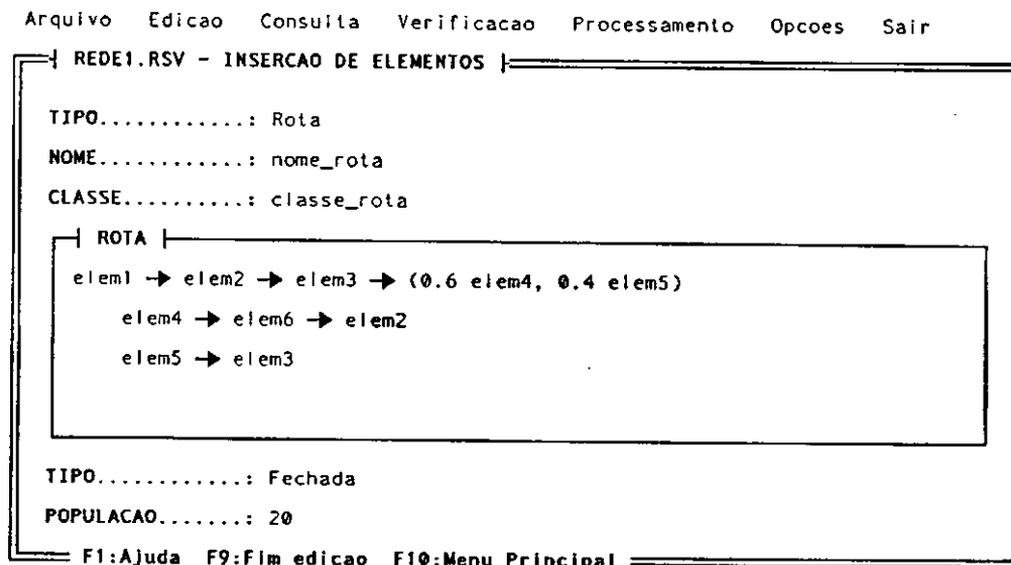


Fig. 5.11 - Definição de Rota de um Modelo

Representação das filas nos pontos escalonadores e pontos de sincronização que compõem a rota:

*nnnnnnnnnnnnnnnnnn.ffff*

Onde,

*nnnnnnnnnnnnnnnnnn* - Nome do elemento, com até 16 caracteres.

*ffff* - Número da fila, no elemento, se houver.

**a) Classe:**

Formato: Nome do elemento de classe desejado.  
Até 16 caracteres, formato livre.

**b) Tipo:**

O SAVAD identifica o tipo e exibe "*Aberta*" ou "*Fechada*", conforme a descrição da rota feita anteriormente.

**c) População:**

Formato: Número inteiro maior que 0.

A população deve ser informada se a rota for fechada.

Pode ser informada se a rota for aberta mas possuir uma população estável.

### 5.9 Classe

A figura 5.12 apresenta os campos do elemento Classe. Os formatos desses campos são descritos a seguir.

```
Arquivo  Edicao  Consulta  Verificacao  Processamento  Opcoes  Sair
| REDEXEMP.RSV - INSERCAO DE ELEMENTOS |
TIPO.....: Classe
NOME.....: nome_classe
PRIORIDADE.....: 1
F1:Ajuda  F10:Menu Principal
```

Fig. 5.12 - Definição de Elemento do Tipo Classe

**a) Prioridade:**

Formato: Número inteiro maior ou igual a 0

## 6. INSTALAÇÃO DO SAVAD

### 6.1 Equipamento

#### a) Configuração mínima

- Microcomputador compatível com IBM-PC com MS-DOS versão 3.3 ou superior.
- Memória de 640 KBytes
- 2 unidades de disquete de 360 KBytes

#### b) Configuração recomendável

- Microcomputador compatível com IBM-PC com MS-DOS versão 3.3 ou superior.
- Memória de 640 KBytes
- Disco rígido.
- Microprocessador Intel 386 ou superior, idealmente com co-processador aritmético.

### 6.2 Instalação

#### a) Instalação em disquete

- Colocar o disco de instalação do SAVAD na unidade de disquete A:
- Digitar INSTALA B:
- Serão solicitados 2 disquetes:
  - Disquete 1: Conterá o SAVAD.
  - Disquete 2: conterá o simulador.

#### b) Instalação em disco rígido

- Colocar o disco de instalação do SAVAD na unidade de disquete A:
- Digitar INSTALA C:
- Será criado um diretório C:\SAVAD para onde os arquivos serão copiados.

### 6.3 Operação

#### a) Operação em disquete

- Colocar o disquete 1 (com o programa *SAVAD*) na unidade de disquete **A:**
- Colocar o disquete com os modelos de redes de filas na unidade de disquete **B:** (se os arquivos estiverem no mesmo disquete do *SAVAD*, permanecer na unidade **A:**).
- Ativar o SAVAD: Digitar: *SAVAD* <Enter>
- Ao especificar qual o arquivo com o modelo de redes a ser utilizado, incluir o nome da unidade de disquete ("**B:**").
- Ao acionar a opção <**P**>rocessamento no menu principal para processar o modelo e receber o aviso de que a simulação vai ser iniciada, retirar o disquete 1 (*SAVAD*) da unidade de disquete **A:** e colocar o disquete 2 (*Simulador*).
- Ao terminar a simulação digital, retirar o disquete 2 (*Simulador*) da unidade de disquete **A:** e colocar o disquete 1 (*SAVAD*).

#### b) Operação em disco rígido

- Ter acesso ao diretório do SAVAD: Digitar *CD \SAVAD* <Enter>.
- Ativar o SAVAD: Digitar: *SAVAD* <Enter>

### 6.4 Arquivos

O SAVAD opera basicamente com 3 tipos de arquivos (extensão padronizada):

- Arquivos com modelos de redes de filas: Tipo "*RSV*".
- Arquivos com o resultado da verificação de integridade dos modelos: Tipo "*VSV*".
- Arquivos com as medidas de desempenho dos modelos de redes calculadas: Tipo "*SSV*".

## REFERÊNCIAS BIBLIOGRÁFICAS

- [Birtwistle, 1973] BIRTWISTLE, G.M. et al.; "*Simula Begin*", Auerbach Publisher Inc. Philadelphia, Pa., 1973.
- [Booch, 1991] BOOCH, G.; "*Object Oriented Design*," The Benjamim/Cummings Publishing Company Inc., 1991, 579p.
- [Brasileiro, 1987] BRASILEIRO, M.A.G.; "*Modelagem de Aplicações Críticas no Tempo em Redes Locais com Passagem de Ficha*", Tese de Doutorado, DEE da UFPB, 1987.
- [Brasileiro et al., 1991] BRASILEIRO, M.A.G., CABRAL, M.I.C. & SILVA, H.M.; "*Um Sistema Especialista para Avaliar o Desempenho de Redes de Computadores*", 9o. SBRC Florianópolis, maio de 1991, p. 431-445.
- [Brasileiro et al., 1992] BRASILEIRO, M.A.G. & CABRAL, M.I.C.; "*An Expert System for Performance Evaluation of Computer System Models*", Summer Computer Simulation Conference 1992, Nevada, EUA, julho de 1992.
- [Cabral et al., 1989] CABRAL, M.I., BRASILEIRO, M.A. & SILVA, H.M.; "*SAVAD - Uma Ferramenta para Avaliar o Desempenho de Sistemas Distribuídos*," Seminaire Franco-Brésilien Sur Les Systems Informatiques Repartis, Florianópolis, setembro de 1989, p. 281-288.
- [Cabral, 1989] CABRAL, M.I.C.; "*Projeto de Pesquisa*", CNPq/UFPb, Campina Grande, 1989.
- [Cabral, 1990] CABRAL, Maria Izabel; "*Curso de Avaliação de Desempenho em Redes de Computadores*", I JONNAI - UFPB/IBM, Campina Grande, 1990.
- [Cabral, 1992] CABRAL, Maria Izabel; "*Um Ambiente de Simulação Inteligente para avaliar o desempenho de Sistemas Distribuídos*", Projeto de Pesquisa, CNPq/UFPb, Campina Grande, 1992.
- [Cabral et al., 1992] CABRAL, M.I., SOUTO, F.A.C., SILVA, H.M. & BRASILEIRO, M.A.; "*The Object-Oriented Design of an Intelligent Interface for Modelling Networks of Queues*", XII International Conference of Chilean Computer Science Society, Santiago, Chile, outubro de 1992.
- [Cabral et al., 1993] CABRAL, M.I., CONCEIÇÃO FILHO, H.C., SOUTO, F.A.C., SILVA, H.M. & BRASILEIRO, M.A.; "*A Simulator for Performance Evaluation of Models for Networks of Queues*", 1993 European Simulation Multiconference - ESM93, Lyon, França, junho de 1993.

- [Carvalho, 1990] CARVALHO, S.; "*Curso de Programação Orientada a Objetos*", I JONNAI - UFPB/IBM, Campina Grande, 1990.
- [Coad & Yourdon, 1992] COAD, Peter & YOURDON, Edward; "*Análise Baseada em Objetos*", Editora Campus, Rio de Janeiro, 1992, 225p.
- [Conceição Filho et al., 1992] CONCEIÇÃO FILHO, H.C, SOUTO, F.A.C., CABRAL, M.I.C & BRASILEIRO, M.A.G; "*Um Simulador para Avaliação de Desempenho de Sistemas de Redes de Filas*", 4a. Semana de Informática da UFBA, Salvador, abril de 1992, p. 21-30.
- [Eckel, 1989] ECKEL, B.; "*Using C++*", Osborne McGrawHill, 1989.
- [Gibbs et al., 1990] GIBBS, Simon; TSICHRITZIS, Dennis; CASAIS, Eduardo; NIERSTRASZ, Oscar & PINTADO, Xavier; "*Class Management for Software Communities*", Communications of the ACM, vol. 33, número 9, setembro de 1990, p. 90-103.
- [Harman & King, 1988] HARMAN, P. & KING, D.; "*Sistemas Especialistas*", Editora Campus, Rio de Janeiro, 1988.
- [Henderson-Sellers & Edwards, 1990] HENDERSON-SELLERS, Brian & EDWARDS, Julian M.; "*The Object-Oriented Systems Life Cycle*", Communications of the ACM, vol. 33, número 9, setembro de 1990, p. 143-159.
- [Jonathan, 1988] JONATHAN, Miguel; "*INTRODUÇÃO A PROGRAMAÇÃO ORIENTADA PARA OBJETOS EM SMALLTALK*", VIII Congresso SBC, Rio de Janeiro, julho de 1988.
- [Jordan, 1990] JORDAN, David; "*Implementation Benefits of C++ Language Mechanisms*", Communications of the ACM, vol. 33, número 9, setembro de 1990, p. 61-64.
- [Khoshnevis & Austin, 1986] KHOSHNEVIS, B. & AUSTIN, W.M.; "*An intelligent interface for systems dynamics modeling*," in Simulation and AI, Society for Computer Simulation (SCS), San Diego, California, p. 81-86, 1986.
- [Kleinrock, 1975] KLEINROCK, L.; "*Queueing Systems, vol 1: Theory*", Wiley Interscience, N.Y., 1975.
- [Korson & McGregor, 1990] KORSON, Tim & MCGREGOR, John; "*Understanding Object-Oriented: A Unifying Paradigm*", Communications of the ACM, vol. 33, número 9, setembro de 1990, p. 40-60.
- [Kreutzer, 1986] KREUTZER, W.; "*System Simulation: Programming Styles and Languages*", Addison-Wesley, London, 1986.
- [Kuehn, 1979] KUEHN, P.J; "*Multiqueue System with Nonexhaustive Cyclic Service*", B.S.T.J., Vol. 58, número 3, março de 1979.

- [Lages & Nogueira, 1986] LAGES, N. A. C. e NOGUEIRA, J. M. S.; "*Introdução aos Sistemas Distribuídos*", Editora da UNICAMP, 1986.
- [Lehmann, 1986] LEHMANN, A.; "*Expert system for interactive simulation of computer system dynamics*," in Simulation and AI, Society for Computer Simulation (SCS), San Diego, Califórnia, p. 21-26, 1986.
- [Lehmann et al., 1986] LEHMANN A., KNODLER, B., KWEE, E. & SZCZERBICKA, H.; "*Dialog-oriented and knowledge-based modeling in a typical environment*," in Simulation and AI, Society for Computer Simulation (SCS), San Diego, California, p. 133-138, 1986.
- [Li & Unger, 1987] LI, XINING & UNGER, BRIAN; "*Languages for distributed simulation*", in Simulation Series, Vol 18, num. 3, 1987, p. 35-39.
- [Moura et al., 1986] MOURA, J.A., SAUVÉ, J.P., GIOZZA, W.F. & ARAÚJO, J.F.; "*Redes Locais de Computadores - Protocolos de Alto Nível e Avaliação de Desempenho*", McGraw-Hill, São Paulo, 1986, 446p.
- [Potier, 1986] POTIER, DOMINIQUE; "*The modeling package QNAP2 and application to Computer Network simulation*", in Computer Networks and Simulation III, Elsevier Science Publishers, North-Holland, p. 235-265, 1986.
- [Pressman, 1987] PRESSMAN, Roger; "*Software Engineering*", Second Edition, McGraw-Hill International Editions, 1987, 567p.
- [Reiser, 1979] REISER, Martin; "*A Queueing Network Analysis of Computer Communication Networks with Window Flow Control*", IEEE Transactions on Communications, Agosto de 1979, vol 27, número 8, p. 1119-1209.
- [Sauer & MacNair, 1985] SAUER, C. & MACNAIR, E.; "*The Evaluation of the Research Queueing Package RESQ*", in Modelling Techniques and Tools for Performance Analysis; North-Holland Publ. Comp., p. 5-24, 1985.
- [Sauer & Chandy, 1981] SAUER, C.H. & CHANDY, K.M.; "*Computer Systems Performance Modeling*", Prentice-Hall, New Jersey, 1981.
- [Schriber, 1974] SCHRIBER, T.J.; "*Simulation Using GPSS*", John Wiley & Sons, 1974.
- [Souto et al., 1993] SOUTO, F.A.C., CABRAL, M.I., SILVA, H.M.; "*Projeto do SAVAD - Sistema de Avaliação de Desempenho de Modelos de Redes de Filas*", Relatório Técnico, UFPb, 1993.
- [Souto et al., 1993] SOUTO, F.A.C., CABRAL, M.I., BRASILEIRO, M.A.G. & SILVA, H.M.; "*Projeto Orientado a Objetos de um Sistema Especialista para Avaliação de Desempenho de Modelos de Redes de Filas*", XII Congresso da SBC, Rio de Janeiro, outubro de 1992

- [Stroustrup, 1986] STROUSTRUP, B.; *"The C++ Programming Language"*, Addison-Wesley, Mass., 1986.
- [Szmit, 1993] SZMIT, RICARDO; *"Programação Orientada para Objeto com Turbo-Pascal 6.0"*; LTC - Livros Técnicos e Científicos Editora Ltda, Rio de Janeiro, 1993, 147p.
- [Wasserman et al., 1990] WASSERMAN, Anthony I.; PIRCHER, Peter A. & MULLER Robert J.; *"The Object-Oriented Structured Design Notation for Software Design Representation"*, IEEE Computer, março de 1990, p. 50-63.
- [Wasserman et al., 1991] WASSERMAN, Anthony I. & PIRCHER, Peter A.; *"Object-Oriented Structured Design and C++"*, Computer Language, janeiro de 1991, p. 41-52.
- [Wirfs-Brock et al., 1990] WIRFS-BROCK, Rebecca J. & JOHNSON, Ralph E.; *"Surveying Current Research in Object-Oriented Design"*, Communications of the ACM, vol. 33, número 9, setembro de 1990, p. 104-124.
- [Zortech, 1990] ZORTECH C++ COMPILER v2.1, 1990, 379p.