

Talisman

Versão 4.40

Manual de operação

Arndt von Staa

LES - Laboratório de Engenharia de Software
Departamento de Informática
PUC-Rio

Novembro 1996

Ambiente de Engenharia de Software Assistido por Computador Talisman

Manual de Operação

Versão 4.4

Outubro 1996

LES/DI/PUC-Rio

Rio de Janeiro, RJ, Brasil

© Copyright 1996 Arndt von Staa

Todos os direitos reservados e protegidos.

Talisman é marca registrada da STAA Informática Ltda.

Este manual faz referência às seguintes marcas registradas:

- IBM-PC, IBM-XT, IBM-AT, da International Business Machines Corporation
- MS-DOS, da Microsoft Corporation
- Windows, da Microsoft Corporation
- WordStar, da MicroPro International Corporation
- Turbo C, da Borland International

Sobre este manual

Este manual tem por objetivo apresentar, de forma sucinta, como operar o ambiente Talisman. É pressuposto o usuário de Talisman conheça as linguagens de representação apoiadas. Para eliminar dificuldades decorrentes de terminologia, cada capítulo apresenta, de forma resumida, os conceitos e elementos de linguagem utilizados.

Sumário

1.	APRESENTAÇÃO.....	1.1
2.	INSTALAÇÃO E ATIVAÇÃO.....	2.1
2.1	Requisitos para a utilização de Talisman.....	2.1
2.2	Instalação de TALISMAN.....	2.1
2.3	Arquivo de diretivas de instalação.....	2.2
2.4	Configuração do ambiente Talisman	2.3
2.5	Início de execução de TALISMAN	2.3
2.5.1	Preparação do ambiente de execução	2.3
2.5.2	Comando de início de execução	2.4
2.5.3	Ativação de Talisval	2.5
2.5.4	Ativação de GERALOG	2.5
2.5.5	Ativação de CONVDOC	2.6
2.6	Auxílio e aprendizado do uso	2.7
3.	CONVENÇÕES	3.1
3.1	Sintaxe da descrição dos comandos.....	3.1
3.1.1	Convenções adotadas para descrever comandos.....	3.1
3.1.2	Sintaxe do texto do comando.....	3.1
3.1.3	Nomes das teclas de controle de “caixa” usados	3.1
3.1.4	Exemplos de descrição de comando	3.1
3.2	Padrão de “feedback” de estado no vídeo.....	3.2
3.3	Comandos padronizados definidos globalmente.....	3.2
3.4	Operação com “Mouse”.....	3.3
3.4.1	Cursor do “mouse”	3.3
3.4.2	Botões simples do “mouse”	3.3
3.4.3	Botões com movimentação	3.3
4.	MENU PRINCIPAL.....	4.1
4.1	Abrir base de software	4.1
4.2	Selecionar treinamento	4.1
4.3	Seleção da linguagem de representação corrente.....	4.1
4.4	Identificação da base de software	4.2
4.5	Editar diagramas	4.2
4.6	Editar dicionários.....	4.2
4.7	Validar	4.2
4.8	Imprimir.....	4.2
4.9	Importar	4.3
4.10	Ativar Serviços	4.3
4.11	Seleção de arquivos	4.3

5.	EDITOR DE DICIONÁRIOS.....	5.1
5.1	Seleção de dicionário.....	5.1
5.2	Ações sobre os objetos selecionados	5.2
6.	EDITOR DE FORMULÁRIOS	6.1
6.1	Conceitos básicos.....	6.1
6.2	Tipos de campos de formulários	6.2
6.3	Comandos e regras específicas.....	6.3
6.3.1	Regras para igualdade de nomes	6.3
6.3.2	Regras específicas para campos <i>NOME</i>	6.3
6.3.2.1	Regras de atualização de nomes.....	6.3
6.3.2.2	Comandos específicos para campos <i>NOME</i>	6.3
6.3.3	Regras específicas para campos <i>RELAÇÃO</i>	6.3
6.3.3.1	Regras de atualização de relações	6.3
6.3.3.2	Comandos específicos para campos <i>RELAÇÃO</i>	6.4
6.4	Controles do editor de formulários	6.4
6.4.1	Modo de inserção de caracteres	6.4
6.4.2	Modo de endentação do início de linhas.....	6.4
6.4.3	Modo de acentuação	6.4
6.4.3.1	Regras para a acentuação	6.5
6.5	Posicionamento do cursor e da janela	6.5
6.6	Inserir linhas	6.5
6.7	Excluir linhas ou caracteres	6.6
6.8	Operações de troca e procura.....	6.6
6.9	Operações com blocos de texto.....	6.6
6.9.1	Regras gerais para blocos de texto.....	6.6
6.9.2	Marcação de bloco.....	6.6
6.9.3	Término da marcação de bloco	6.7
6.9.4	Operações com bloco corrente.....	6.7
6.9.5	Ler, gravar e imprimir blocos.....	6.7
6.10	Terminar a edição de formulário.....	6.8
6.11	Comandos diversos	6.8
6.11.1	Caracteres especiais	6.8
6.11.2	Parâmetros de controle do editor de formulários	6.8
7.	EDITOR DE ESTRUTURAS.....	7.1
7.1	Conceitos básicos.....	7.1
7.2	Ativação do editor.....	7.3
7.3	Foco de atenção	7.4
7.4	Navegação na estrutura	7.5
7.5	Criação e exclusão de blocos	7.5
7.6	Cópia e movimentação de blocos.....	7.5

7.7	Estabelecer chamadas de pacotes	7.6
7.7.1	Estabelecer: <i>bloco chama pacote</i>	7.6
7.7.2	Estabelecer: <i>Pacote é chamado por bloco</i>	7.6
7.8	Empacotar, desempacotar	7.6
7.9	Navegação para outros editores	7.7
7.10	Terminar a edição de estruturas	7.7
7.11	Outros comandos	7.7
8.	EDITOR DE DIAGRAMAS	8.1
8.1	Conceitos básicos	8.1
8.2	Instâncias de elementos	8.2
8.3	Estruturas de diagramas	8.2
8.4	Foco de atenção	8.2
8.5	Posicionar o foco de atenção	8.3
8.5.1	Posicionar o cursor e a janela	8.3
8.5.2	Posicionar por seleção	8.3
8.6	Inserir elementos no diagrama	8.3
8.6.1	Inserir objetos	8.3
8.6.1.1	Regras para nomes de objetos novos	8.4
8.6.1.2	CrITÉRIOS para igualdade de nome	8.4
8.6.1.3	Comandos especiais para nomes	8.4
8.6.2	Inserir instância de imagem de ligação	8.4
8.6.3	Inserir ligação entre instâncias de objetos	8.5
8.6.3.1	Modo de criação dos pontos de passagem das ligações	8.5
8.6.3.2	Procedimento para criar ligações	8.5
8.6.4	Inserir rótulo em ligação existente	8.6
8.6.5	Inserir ponto de passagem em ligação corrente	8.6
8.6.6	Criar folha filho para um objeto	8.6
8.7	Editar	8.6
8.7.1	Editar especificações	8.6
8.7.2	Editar ligação	8.6
8.8	Mover	8.6
8.8.1	Mover objetos e rótulos	8.6
8.8.2	Mover pontos de passagem de ligação	8.6
8.8.3	Procedimento para reorganizar diagramas	8.6
8.8.4	Exclusão	8.7
8.9	Navegar e/ou terminar a edição de diagramas	8.7
8.10	Comandos diversos	8.7
8.11	Parâmetros do editor de diagramas	8.7
9.	MENU SERVIÇOS	9.1
9.1	Configuração do ambiente de desenvolvimento	9.1

9.1.1	Instalação de programas de formulário fornecidos	9.1
9.1.2	Instalar outros programas de formulário	9.1
9.1.3	Edição de programas de formulários.....	9.2
9.2	Configuração da estação de trabalho	9.2
9.2.1	Configurar cores.....	9.2
9.2.2	Configurar diretório	9.3
9.2.2.1	Seleção de arquivo	9.4
9.2.3	Configurar saídas impressas.....	9.5
9.2.3.1	Configurar impressora.....	9.5
9.2.3.2	Configurar relatório	9.5
9.2.3.3	Configurar <i>spool</i>	9.6
9.2.4	Configurar <i>mouse</i>	9.7
9.3	Modos de operação	9.7
9.3.1	Modo <i>auto-save</i>	9.7
9.3.2	Modo LOG.....	9.7
9.3.2.1	Criação de LOG	9.7
9.3.2.2	Executar o LOG	9.7
9.4	Operações diversas	9.8
9.4.1	Editar arquivos sequenciais.....	9.8
9.4.2	Copiar arquivos.....	9.8
10.	MACROS E LIÇÕES	10.1
10.1	Conceitos	10.1
10.2	Ativação	10.1
10.2.1	Ativação de macro	10.1
10.2.2	Ativação de lição	10.1
10.3	Execução.....	10.1
10.4	Criação.....	10.1
10.4.1	Início do registro	10.2
10.4.2	Término do registro	10.2
10.4.3	Operações especiais	10.2
10.5	Edição do corpo de macro ou lição.....	10.4
10.6	Exclusão de macro ou lição	10.4
10.7	Modos de execução de macros	10.4
10.8	Macros instaladas.....	10.5
10.8.1	Macros do Editor de Formulários	10.5
11.	MENUS E DIÁLOGOS.....	11.1
11.1	Conceitos básicos.....	11.1
11.2	Comandos padronizados	11.1
11.3	Selecionar subconjunto	11.2
11.4	Selecionar ação, botões.....	11.2

11.5	Posicionar item	11.2
11.6	Editar itens contendo valores predefinidos	11.2
11.7	Editar item valor	11.3
11.8	Modos de edição de valor	11.3
12.	AUXÍLIO E TREINAMENTO	12.1
12.1	Condição básica para o uso	12.1
12.2	Lições	12.1
12.2.1	Seleção da lição a executar	12.2
12.2.2	Modos de execução de lições	12.2
13.	IMPORTAR E EXPORTAR	13.1
13.1	Exportação	13.1
13.2	Importação	13.1
13.3	Ativação da importação	13.2
13.4	Formato do arquivo de importação	13.3
14.	RECUPERAÇÃO E OTIMIZAÇÃO DE BASES	14.1
14.1	Causas para erros em bases de software	14.1
14.2	O programa TALISVAL	14.2
14.2.1	Operação básica de TALISVAL	14.2
14.3	Recuperar uma base de software	14.3
14.4	Otimizar uma base de software	14.3
14.5	Mensagens do recuperador de bases	14.3
14.5.1	Sintaxe das mensagens de erro	14.4
14.5.2	Mensagens indicativas das fases do recuperador	14.4
14.5.3	Erros da estrutura de acesso básica	14.5
14.5.4	Erros em dicionários	14.6
14.5.5	Erros em objetos	14.7
14.5.6	Erros em diagramas	14.9
14.5.7	Erros de eliminação de lixo	14.12

Página propositalmente deixada em branco

1. APRESENTAÇÃO

Neste manual é descrito como instalar e operar Talisman.

Talisman é um ambiente de engenharia de software configurável. Através de parâmetros contidos na base de conhecimento pode-se adicionar ou alterar as linguagens de representação suportadas. Através de programas de formulários, redigidos pelos usuários de Talisman, pode-se alterar o conteúdo e o formato de especificações, relatórios, código composto a partir da base de software, regras de validação, regras de exportação e outros. O uso cotidiano de Talisman tem confirmado a grande flexibilidade do sistema e a sua capacidade de adaptação a diferentes necessidades dos desenvolvedores de software. O fato dos editores de Talisman serem meta-editores pode parecer confuso a primeira vista. No entanto, assegura uma uniformidade de tratamento a todas as representações de mesma categoria, compensando assim a aridez do aprendizado de seu uso.

Talisman tem propriedades de editor de hiper-documentos. É muito fácil navegar de representação para representação, independentemente do fato de serem textos, estruturas ou diagramas extraídos da base de software.

Talisman assegura a consistência entre as diferentes representações. Isto permite que se editem fatos do sistema objetivo em alguma das representações, tornando-os disponíveis em qualquer outra representação que, porventura, referencie estes fatos. Assim ao alterar o nome de um objeto em um texto qualquer, os diagramas em que aparecem instâncias destes objetos serão devidamente atualizados.

Talisman armazena os fatos relativos ao sistema objetivo em forma de objetos e atributos destes objetos. Os objetos pertencem a classes identificadas na base de conhecimento. Os atributos podem ser:

Nome	todos os objetos possuem um nome. Com raríssimas exceções estes nomes são dicionarizados, facilitando o acesso aos objetos através da consulta aos dicionários.
Aliases	aliases são strings de caracteres. Todos os objetos podem ter até 256 diferentes aliases. Cabe o programador dos formulários utilizados definir o significado de cada um dos aliases em uso.
Fragmentos de texto	fragmentos de texto são seqüências de zero ou mais linhas de texto. O tamanho destes fragmentos de texto é virtualmente ilimitado (mais de 30.000 linhas para cada fragmento). Todos os objetos podem ter até 256 diferentes fragmentos de texto. Cabe o programador dos formulários utilizados definir o significado de cada um dos fragmentos de texto em uso.
Relações	estabelecem vínculos entre objetos. Relações pode ser utilizadas para compor textos linearizados, por exemplo código gerado. Podem ser utilizadas para a navegação entre objetos. E podem ser utilizadas para validar a semântica do conteúdo da base de software.

No seu atual estágio de desenvolvimento, Talisman define 10 linguagens de representação, cerca de 60 classes de objetos e cerca de 140 diferentes nomes de relações. Cada nome de relação estabelece uma semântica definida. Um mesmo nome pode ser utilizado em várias classes. Por esta razão torna-se possível estabelecer as cerca de 600 relações existentes entre classes de objetos, sem tornar excessivamente complexo o entendimento do esquema de dados.

Talisman tem sido utilizado com sucesso no desenvolvimento de diversas aplicações em várias empresas. Em particular Talisman está sendo utilizado para desenvolver e manter a si próprio.

Página propositalmente deixada em branco

2. INSTALAÇÃO E ATIVAÇÃO

2.1 Requisitos para a utilização de Talisman

Talisman opera em máquinas compatíveis com IBM-PC, IBM-XT, IBM-AT, 386, 486, Pentium e requer a seguinte configuração mínima:

- pelo menos 500k de memória principal livres. A memória livre é integralmente utilizada, assim quanto mais memória estiver disponível, melhor será o desempenho. Evite o uso de software residente (TSR), uma vez que consome espaço de memória, comprometendo o desempenho de Talisman.
- um “drive” de disco rígido. São requeridos cerca de 3.500K bytes para poder instalar o programa completo, e cerca de 2.000K bytes exclusivamente para os programas.
- um “drive” disquete utilizado durante a instalação a partir de disquete.
- vídeo, em um dos padrões: CGA, EGA, VGA ou Hércules.

Talisman suporta ainda:

- “mouse”. O “driver” do “mouse” deve obedecer ao padrão MicroSoft. O “mouse” deve ter dois ou mais botões, sendo que serão utilizados somente os botões da esquerda e da direita.
- impressoras compatíveis com o padrão EPSON.
- impressoras laser compatíveis com HP (PCL - Printer Command Language).
- impressoras jato de tinta compatíveis com HP Desk-Jet.

Talisman requer um sistema operacional compatível com MS-DOS 5.0 ou mais recente.

Talisman tem sido utilizado com sucesso em Windows 3.1, 3.11 (*Workgroups*), 95, NT 3.51 e NT 4.0. Tem sido utilizado também em sistemas MAC emulando DOS. Em Windows Talisman opera como aplicação *não Windows*. Deve-se solicitar 640K de memória e, preferencialmente, operar em “full screen mode”. Todos estes parâmetros são editados por intermédio do “PIF Editor” ou do “Program Item Editor” que fazem parte de Windows. Sendo desejado, pode-se utilizar os ícones TALISMAN.ICO e TALISVAL.ICO que se encontram no diretório raiz do ambiente, ver na seção a seguir.

2.2 Instalação de TALISMAN

Talisman é distribuído

- por meio de disquete 3½”. Gere pelo menos uma cópia “backup” do disquete distribuído. Guarde o disquete original. Para a sua maior segurança, opere sempre a partir do disquete cópia. Em adição, proteja o disquete contra gravação.
- por meio de ftp. Neste caso os arquivos de instalação vem agregados em um arquivo auto descompactante. Copie este arquivo para um diretório a partir do qual será feita a instalação e execute este arquivo. Serão produzidos os arquivos de instalação.

Para instalar, posicione-se no diretório ou *drive* onde se encontram os arquivos de instalação e execute:

INSTALAR

O programa guia a instalação passo a passo. Será solicitado o *drive* ou diretório onde se encontram os arquivos de instalação. Este é tipicamente o diretório ou *drive* que contém o programa de instalação. O nome do diretório ou *drive* origem pode ser redefinido. Serão emitidas mensagens de erro caso os arquivos de instalação não sejam encontrados no diretório ou disquete montado no *drive* de instalação.

Será solicitado o diretório *xxx* a partir do qual Talisman deverá ser instalado. O valor *default* de *xxx* é C:\Talisman. Este nome pode ser redefinido, inclusive para *drive* diferente.

Caso ainda não existam, serão criados os diretórios:

<i>xxx</i>	destinado aos programas e arquivos necessários para a execução.
<i>xxx\BSW</i>	destinado às bases de software.

xxx\FORM	destinado aos programas de formulário.
xxx\VAL	conterá o programa TALISVAL que efetua a recuperação e otimização de bases de software.
xxx\TEXTO	destinado a arquivos seqüenciais.
xxx\PROGS	destinado a arquivos de programas gerados com apoio de Talisman.

2.3 Arquivo de diretivas de instalação

O programa de instalação utiliza um arquivo de diretivas. Este arquivo de diretivas é fornecido como parâmetro na linha de comando. Caso o parâmetro não seja fornecido, procura-se o arquivo `INSTALAR.DAT` no diretório ou *drive* origem da instalação. O instalador instala qualquer sistema composto por um ou mais arquivos a serem expandidos. O arquivo de diretivas satisfaz a sintaxe:

```
Cada comando aparece em uma linha.
Linhas iniciando com * são comentários.
<Arquivo> ::= <O que instala><Comandos>
<O que instala> ::= $ título do sistema sendo instalado
<Comandos> ::= 0 - [ <linha comando> ] ;
<Linha comando> ::= ( 'D' <Comando de expansão>
| 'M' <Mensagem>
| <Outros>
)
<outros> ::= $ qualquer caractere na coluna inicial
da linha e que seja diferente de
'D' ou 'M'. Estas linhas são tratadas
como comentários.
<Comando> ::= <Arquivo origem>
<Condição>
<Solicitação>
<Destino default>
<Disco default>
<Arquivo origem> ::= $ arquivo a ser expandido, nome e
extensão
<Condição> ::= $ texto a ser exibido ao usuário para
decida se o conteúdo do arquivo deve
ser instalado ou não.
Se nulo, instala sempre
<Solicitação> ::= $ texto de prompt da solicitação do
diretório onde será instalado
Se nulo, instala sem prompt
<Destino default> ::= $ nome do diretório onde será feita a
expansão. Caso o diretório não exista
será criado. Se for precedido de +,
adiciona o nome de sub-diretório ao
diretório base corrente. Se não
precedido de + registra o diretório
como diretório base corrente.
<Disco default> ::= $ rótulo do disco onde deverá estar o
arquivo a expandir. É usado somente
como interface com o usuário.
<Mensagem> ::= <linha> .
<linha> ::= $ inteiro entre 3 <= e <= 14,
corresponde à linha da mensagem na
janela de interação do instalador .
<texto> ::= $ uma seqüência de caracteres com menos
do que 54 caracteres .
```

Obs. Todos os elementos devem ter o primeiro caractere não branco. Desejando-se formatar as linhas, o primeiro caractere deve ser “.”. Linhas em branco devem conter exatamente o caractere “.”. Os caracteres de comando ‘D’ ou ‘M’ devem estar na primeira coluna da linha. As definições de um elemento de expansão devem estar em linhas consecutivas sem linhas em branco entre elas, sendo que todos os componentes são obrigatórios. As definições de uma mensagem devem estar em linhas consecutivas sem linhas em branco entre elas. A ordem das

linhas de definições de mensagens é qualquer. Entre definições consecutivas podem aparecer linhas em branco ou contendo comentários. Estas linhas serão ignoradas.

2.4 Configuração do ambiente Talisman

Em adição aos arquivos, o próprio ambiente Talisman deve ser configurado. Para evitar dificuldades iniciais, Talisman é fornecido em sua configuração padrão. Esta é formada por:

- i. Linguagem Identificação da Base (BAS), formulário: especificação.
- ii. Linguagem *Análise da Organização* (ANO), formulários: especificação e impressão.
- iii. Linguagem *Sistema de Informação* (SIN), formulários: especificação, validação e impressão.
- iv. Linguagem *Análise de Requisitos* (ARQ), formulários: especificação e impressão.
- v. Linguagem *Gerência de Projetos* (PRJ), formulários: especificação e impressão.
- vi. Linguagem *Entidades e Relacionamentos* (ERA), formulários: especificação, validação simples, e impressão.
- vii. Linguagem *Fluxo de Dados* (DFD), formulários: especificação simples, validação simples, e impressão.
- viii. Linguagem *Estados e transições* (TRN), formulários: especificação e impressão.
- ix. Linguagem *Estruturas Modulares* (DEM), formulários: especificação, validação, impressão, linearização para "C", e exportação.
- x. Linguagem *Documentação* (DOC), formulários: especificação e impressão.
- xi. Linguagem *Sistemas de Informação* (SIN)
- xii. Linguagem *Documentação de programas* (DPG)

Os programas de formulário determinam o conteúdo e o formato das especificações, o conteúdo e o formato dos textos linearizados, impressos, exportados, e as regras de validação. Sendo desejado instalar outros programas de formulário, ou mesmo desenvolver novos programas, utilize o serviço **Configurar formulários** (instala e/ou edita programas de formulários quaisquer), disponível no menu **Serviços**. O capítulo **Programação de formulários** explica como programar e configurar formulários.

Em adição à instalação de programas de formulários, o menu **Serviços** permite, ainda, a alteração de parâmetros de configuração da estação de trabalho. Em particular, permite o ajuste os nomes dos diretórios *default* a serem utilizados. Isto é feito com o serviço **Configurar Diretórios**.

2.5 Início de execução de TALISMAN

2.5.1 Preparação do ambiente de execução

Talisman utiliza diversos arquivos de apoio. Caso estes arquivos não estejam no diretório corrente ao iniciar a execução, os diretórios deverão ser informados através de variáveis de ambiente (comando **SET** do sistema operacional). As variáveis de ambiente são:

```
SET TALISBCO=drive:\diretório
```

informa onde está a base de conhecimento, arquivo TALISMAN.BCO. Esta contém as informações de controle do ambiente Talisman e é imprescindível.

```
SET TALISBAX=drive:\diretório
```

informa onde está a base de auxílio, arquivo TALISMAN.BAX. Este contém as informações de auxílio de operação do ambiente Talisman e é opcional.

```
SET TALISUSO=drive:\diretório
```

informa onde será criado o arquivo de trabalho TALISMAN.\$\$1. O arquivo de trabalho ocupa cerca de 400K. Ele será destruído ao terminar a execução de Talisman. O arquivo de trabalho pode residir em um *ram-disk*.

Observação 1: os nomes de arquivos utilizados em Talisman são sempre nomes completos, formados por:

```
drive:\diretório\nome.extensão
```

Para arquivos residentes no diretório raiz, utilize:

```
drive:\nome.extensão
```

Observação 2: Talisman é um sistema mono-usuário. O compartilhamento de bases (Uso, Conhecimento, Auxílio e Software) pode levar a resultados imprevisíveis, possivelmente destruindo o conteúdo das bases compartilhadas. Compartilhamento pode ocorrer, por exemplo, ativando 2 ou mais sessões Talisman quando operando em Windows, ou acessando, a partir de 2 ou mais estações, arquivos públicos disponíveis em rede. Ao operar com Windows, procure usar sempre o utilitário SHARE que impede o compartilhamento de arquivos abertos para a gravação.

Observação 3: Para operar em rede, coloque o arquivo TALISMAN.EXE em um diretório público ler somente da rede. Os arquivos TALISMAN.BCO, TALISMAN.BAX e TALISMAN.\$\$1 devem estar em um diretório privado do usuário.

2.5.2 Comando de início de execução

O comando de linha de ativação de Talisman admite os seguintes parâmetros:

```
TALISMAN [/v<vídeo>] [/l<linhas texto>]
          [/c<arquivo log>] <base de software>]
```

A ordem de apresentação dos parâmetros é qualquer, e todos os parâmetros são opcionais. Os parâmetros têm o significado a seguir:

vídeo identifica o sistema de vídeo a ser utilizado. Caso o parâmetro não seja fornecido, o sistema utilizado será o que está ativo no momento da ativação de Talisman. Este parâmetro deve ser fornecido se Talisman não conseguir reconhecer o sistema de vídeo em uso, ou caso se deseje utilizar um sistema de vídeo diferente do sistema em uso. Antes de selecionar explicitamente um sistema de vídeo, assegure-se que a placa de vídeo é capaz de suportá-lo. O parâmetro é um de:

vga	utiliza o padrão de vídeo VGA
cga	utiliza o padrão de vídeo CGA
ega	utiliza o padrão de vídeo EGA
hgc	utiliza o padrão de vídeo Hércules

linhas texto determina o número de linhas em modo texto a serem utilizadas. Caso este parâmetro não seja fornecido será utilizado o número de linhas em modo texto em vigor ao ativar Talisman, ou será utilizado o *default* para o sistema de vídeo caso este seja selecionado pelo parâmetro <vídeo>. Valores menores ou iguais a 25, configurarão a placa de vídeo para o menor número de linhas em modo texto (usualmente 25). Valores maiores do que 25 configurarão a placa de vídeo para o maior número de linhas em modo texto (em EGA 43, em VGA 50). O número de linhas de texto em modo gráfico depende exclusivamente da resolução do sistema de vídeo utilizado.

arquivo log nome de um arquivo em formato log de Talisman. Este parâmetro deve ser fornecido somente se for fornecido o parâmetro <base de software>. Caso seja fornecido, a base será aberta, o log será ativado e executado com relação a esta base e, ao terminar a execução do log, a base será fechada e será terminada a execução de Talisman.

base de software nome de uma base de software. Caso este parâmetro seja fornecido, a base de software será aberta ao ativar Talisman. Caso o parâmetro <arquivo log> não seja fornecido, Talisman ativa o menu principal após abrir a base.

Em ambiente Windows 95, ou NT 4.0, pode-se ativar Talisman arrastando uma base de software sobre um ícone de ativação de Talisman.

2.5.3 Ativação de Talisval

O programa Talisval controla e recupera a integridade de bases de software porventura danificados. Permite, também, otimizar (desfragmentar) bases de software.

O comando de linha de ativação de Talisval admite os seguintes parâmetros:

```
TALISMAN [/v<vídeo>] [/l<linhas texto>] [<base de software>]
```

A ordem de apresentação dos parâmetros é qualquer, e todos os parâmetros são opcionais. Os parâmetros têm o significado a seguir:

vídeo identifica o sistema de vídeo a ser utilizado. Caso o parâmetro não seja fornecido, o sistema utilizado será o que está ativo no momento da ativação de Talisman. Este parâmetro deve ser fornecido se Talisman não conseguir reconhecer o sistema de vídeo em uso, ou caso se deseje utilizar um sistema de vídeo diferente do sistema em uso. Antes de selecionar explicitamente um sistema de vídeo, assegure-se que a placa de vídeo é capaz de suportá-lo. O parâmetro é um de:

vga	utiliza o padrão de vídeo VGA
cga	utiliza o padrão de vídeo CGA
ega	utiliza o padrão de vídeo EGA
hgc	utiliza o padrão de vídeo Hércules

linhas texto determina o número de linhas em modo texto a serem utilizadas. Caso este parâmetro não seja fornecido será utilizado o número de linhas em modo texto em vigor ao ativar Talisman, ou será utilizado o *default* para o sistema de vídeo caso este seja selecionado pelo parâmetro <vídeo>. Valores menores ou iguais a 25, configurarão a placa de vídeo para o menor número de linhas em modo texto (usualmente 25). Valores maiores do que 25 configurarão a placa de vídeo para o maior número de linhas em modo texto (em EGA 43, em VGA 50). O número de linhas de texto em modo gráfico depende exclusivamente da resolução do sistema de vídeo utilizado.

base de software nome de uma base de software. Caso este parâmetro seja fornecido, a base de software será aberta ao ativar Talisval. Antes de ativar o menu principal será confirmado se é desejado criar uma cópia backup da base de software e, se afirmativo, o nome do arquivo onde será gravada esta cópia. Tendo em vista que a recuperação de bases de software realizada por Talisval pode alterar substancialmente a estrutura interna destas bases, é **fortemente recomendado** que se gere uma cópia backup antes de realizar qualquer operação de recuperação.

Em ambiente Windows 95, ou NT 4.0, pode-se ativar Talisman arrastando uma base de software sobre um ícone de ativação de Talisman.

2.5.4 Ativação de GERALOG

O programa GERALOG gera arquivos em formato *LOG* de Talisman capazes de criar ou atualizar bases de software através da importação de zero ou mais arquivos de importação. Arquivos *LOG* assim gerados podem ser utilizados para ativar Talisman como comando de linha e são particularmente úteis para criar bases de software sob controle de ferramentas tais como *MAKE*.

O programa GERALOG pode ser ativado de uma das maneiras a seguir:

```
GERALOG /?
```

Nesta modalidade, será apresentada uma descrição sucinta de como ativar GERALOG.

```
GERALOG <base de software> /l <nome arquivo 1> <nome arquivo 2> ...
```

Nesta modalidade, a base de software será criada importando-se os arquivos <nome arquivo 1> <nome arquivo 2> etc. Cabe salientar que deve existir um caractere em branco após o parâmetro /l e entre os nomes de arquivos.

```
GERALOG <base de software> @<nome arquivo parâmetros>
```

Nesta modalidade, a base de software será criada, importando-se os arquivos identificados no arquivo de parâmetros. Note que não existe espaço após o caractere '@'. O arquivo de parâmetros tem o formato:

- cada nome de arquivo a importar deverá estar contido em uma linha;
- o nome do arquivo inicia na primeira coluna;
- caso a primeira coluna contenha um caractere '*', branco ou de controle, a linha será considerada linha de comentário;
- podem ser fornecidos no máximo 100 arquivos.
- Para que o arquivo *log* funcione corretamente, é necessária a satisfação de uma das condições a seguir:
 - os nomes dos arquivos de importação são completos, identificando <drive>:\<diretório>\<nome arquivo>
 - o diretório corrente ao ativar Talisman é o diretório relativo ao qual se consegue acessar os arquivos a importar. Na segunda modalidade de uso, os nomes podem ser relativos ao diretório corrente ao ativar Talisman.

2.5.5 Ativação de CONVDOC

O programa CONVDOC converte arquivos de texto gerados com Talisman para arquivos que possam ser importados por aplicações Windows, em especial WORD. Essencialmente, converte os caracteres acentuados gerados com Talisman (convenção DOS-PC) para os caracteres acentuados utilizados pelas aplicações Windows (Norma ISO-ASCII). Para utilizar forneça o comando:

```
CONVDOC [/f] [/b] [/cArqDefn] /oArqOrigem /dArqDestino
```

Os parâmetros da linha de comando podem vir em qualquer ordem, e têm o seguinte significado:

- | | |
|------------|--|
| /f | quando presente suprime a inclusão dos controles de nova linha em linhas não vazias. Como os caracteres NL e CR serão inseridos somente se a linha estiver vazia, estas passarão a definir fim de parágrafo. |
| /b | quando presente provoca a eliminação de seqüências de brancos e/ou tabulações. Normalmente, seqüências de caracteres em branco são copiadas para o arquivo convertido. |
| ArqDefn | quando presente, identifica o arquivo contendo as definições de conversão. A extensão <i>default</i> é .DAT. Caso este parâmetro não seja fornecido, o programa vai procurar o arquivo CONVDOC.DAT no diretório em que se encontra o programa sendo executado. |
| ArqOrigem | obrigatório, é o nome do arquivo texto origem. A extensão <i>default</i> é .TXT. |
| ArqDestino | obrigatório, é o nome do arquivo texto convertido. A extensão <i>default</i> é .CNV. |

O arquivo de definições de conversão tem o seguinte formato:

- linhas contendo '*' no início e linhas em branco são linhas de comentários.
- todas as linhas de definição devem apresentar duas seqüências de caracteres separadas por um ou mais brancos, cada ocorrência da primeira seqüência será substituída pela segunda. As seqüências podem ter até 20 caracteres. A convenção dos caracteres do arquivo é DOS.
- caracteres especiais são definidos por seqüências de escape. São seguintes os caracteres especiais:

- | | |
|------|--|
| \nnn | nnn número decimal, substitui o conjunto pelo caractere que corresponde ao valor de nnn |
| \L | L letra maiúscula no conjunto @ABCDEFGHIJKLMNOPQRSTUVWXYZ[]^_, substitui o par pelo correspondente caractere de controle ^L |
| \e | substitui a seqüência de entrada pela seqüência vazia |
| \n | substitui o para pelo caractere "nova linha" |
| \r | substitui o para pelo caractere "retrocesso de carro" |
| \t | substitui o para pelo caractere "tabula" |
| \b | substitui o para pelo caractere "branco" |

- \\ substitui o para pelo caractere “\”
- \\x quando x for qualquer caractere diferente dos demais, substitui o par por x

2.6 Auxílio e aprendizado do uso

TALISMAN possui um sub-sistema de instrução programada. Este sub-sistema executa *lições* que explicam, dinamicamente, como realizar tarefas em TALISMAN. Lições são ativadas a partir do item **Treinamento** do **Menu Principal**.

Em adição, TALISMAN possui um amplo apoio de auxílio contextual. Este é ativado pela tecla **[F1]**, *Auxílio*, ou pelo botão **{DIR}**, *direita*, do “mouse”.

Detalhes sobre o uso de lições e do sistema de auxílio podem ser encontrados no capítulo **Auxílio e Treinamento**.

Página propositalmente deixada em branco

3. CONVENÇÕES

3.1 Sintaxe da descrição dos comandos

3.1.1 Convenções adotadas para descrever comandos

[cmd]	entre colchetes, é um comando fornecido pelo teclado. Os comandos podem ser formados por uma ou mais teclas.
{botão}	entre chaves, é um botão de “mouse”. A forma de selecionar cada um dos comandos por intermédio de “mouse” não será definida neste manual, uma vez que comandos são obtidos a partir de uma sucessão de escolhas óbvias. As escolhas de comandos por intermédio de “mouse” inicia sempre com o botão {DIR}. Já as demais ações efetuadas com “mouse”, tais como rolagem de vídeo, são descritas nos respectivos capítulos, bem como, no seu sentido geral, ao final do presente capítulo.
<tecla>	entre parênteses angulares, é o nome de uma das teclas contidas no teclado.

3.1.2 Sintaxe do texto do comando

[nome₁ nome₂ ... nome_n]

Quando redigido em maiúsculo: nome_i corresponde a uma das teclas existentes no teclado.

Quando redigido em minúsculo: nome_i representa o conjunto de teclas designado por nome_i. Por exemplo, o nome **setas** corresponde ao conjunto de teclas de dimensionamento <SOBE>, <DESCE>, <ESQ> e <DIR>.

3.1.3 Nomes das teclas de controle de “caixa” usados

^	control
!	alt
↑	shift (caixa alta)

3.1.4 Exemplos de descrição de comando

[PGUP]	comando acionado pela tecla <PGUP>.
[^Q A]	comando acionado ao teclar simultaneamente as teclas <CTRL> e <Q>, e, após, a tecla <A>. No Editor de Formulários corresponde ao comando Procura e Troca.
[!I]	comando acionado ao teclar simultaneamente as teclas <ALT> e <I>. No Editor de Formulário, quando posicionado em um campo Nome ou em um campo Relação, busca e seleciona nomes dos objetos contidos no dicionário correspondente ao campo sendo editado.

3.2 Padrão de “feedback” de estado no vídeo

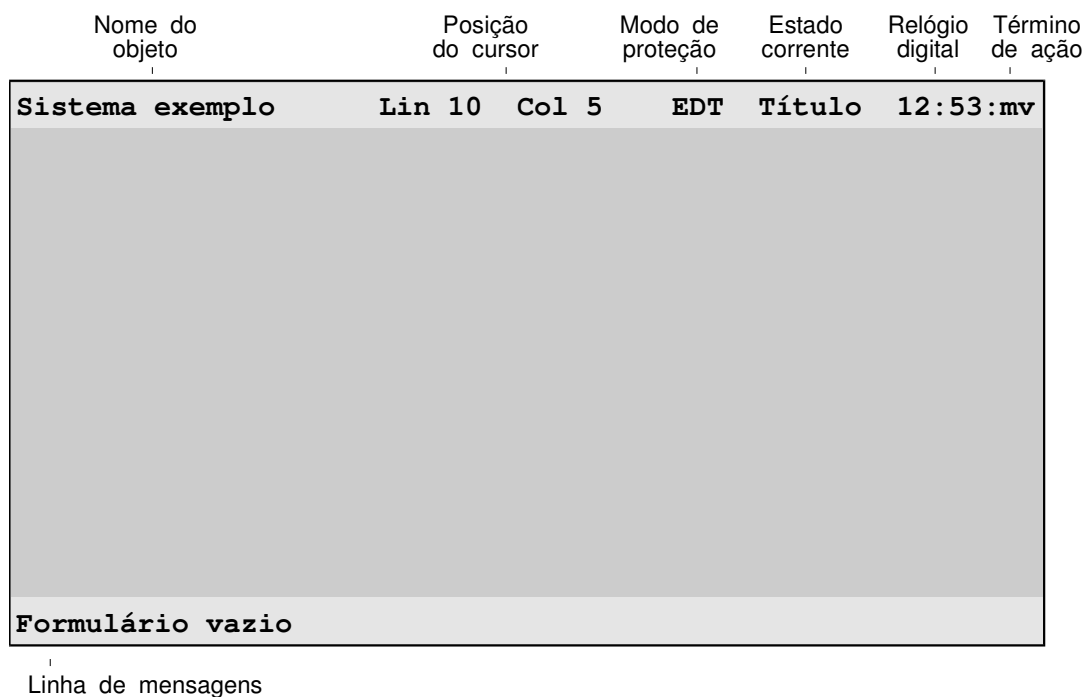


Figura 0.1. Padrão de tela

A figura 0.1 ilustra o padrão de tela (“desk top”) apresentado. A linha superior do vídeo informa o estado corrente do sistema. As seguintes informações são exibidas:

Nome do objeto	nome do elemento ou da folha sendo editada.
Linha e Coluna	posição do cursor de edição. Usado pelos editores de diagramas e de formulários.
Modo de proteção	EDT é permitido, PRT não é permitido alterar o conteúdo do campo
Estado corrente	identifica o que está sendo realizado no momento. O estado corrente informa a evolução da execução de comandos complexos definidos através de procedimentos.
Relógio digital	hora do dia.
Término de ação	mensagem transiente exibida sobre a parte de segundos do relógio, avisa o término de operações realizadas em multiprogramação. Estas operações são tipicamente operações de “auto-save” e de controle de integridade. Cada vez que aparecer sv , significa que todas as alterações até o momento foram registradas na base de software. Sempre que aparecer mv , foi completado um ciclo de validação da estrutura interna da memória virtual.

A linha inferior do vídeo exibe as mensagens de erro e advertência porventura geradas ao executar.

3.3 Comandos padronizados definidos globalmente

[F1]	inicia um diálogo para a obtenção de um comando por seleção via régua de comandos. É idêntico ao botão {DIR} do “mouse”.
[ENTER]	em menu ou diálogo de seleção: seleciona o elemento corrente. em edição de texto: vai para nova linha. em operações que requeiram ações múltiplas: (p.ex. Mover) confirma e termina a operação.
[ESC]	cancela a operação corrente, inclusive as que requeiram ações múltiplas.
[F2]	edita o conjunto de parâmetros que governam o editor corrente.
[!M]	(teclas <ALT> e <M> pressionadas simultaneamente, ver padrão de teclas “caixa”) ativa o menu de macros de teclado do editor corrente (ver capítulo Macros e Lições).
[!X tecla]	executa a macro de teclado de nome tecla do editor corrente. Não estando definida, ou sendo <ENTER>, ativa o menu de macros, comando [!M] acima.

3.4 Operação com “Mouse”

3.4.1 Cursor do “mouse”

O cursor do “mouse” difere dos demais cursores tanto no aspecto como na forma de operação. Ele opera solto, podendo ser movido sem que isto afete a edição em andamento. O cursor de edição será posicionado no local do cursor do “mouse” pressionando um dos botões {ESQ} ou {DIR}. O botão {ESQ} além de posicionar o cursor do editor, também ativa a régua de comandos permitindo a seleção de um comando.

3.4.2 Botões simples do “mouse”

Os botões simples realizam:

- | | |
|-------|---|
| {DIR} | botão da direita no mouse. Tem o sentido geral: <i>selecionar ação</i> relativa à posição corrente do cursor de “mouse”. <i>Ao processar menus</i> : seleciona a opção corrente. <i>Nos demais casos</i> : posiciona o cursor do editor sobre o cursor do “mouse” e inicia o diálogo de seleção do comando via régua de comandos. <i>Observação</i> : a tecla [F1] ativa a mesma régua de comandos. |
| {ESQ} | botão da esquerda no mouse. Tem o sentido geral: <i>marcar posição</i> . <i>Ao efetuar operações que requeiram ações múltiplas</i> : (p.ex. Mover elemento) posiciona o cursor do editor sobre o cursor do “mouse”, confirma e termina a operação. <i>Ao editar valores de um conjunto definido (diálogos)</i> : troca o valor corrente para o próximo da lista de valores permitidos. <i>Ao selecionar elementos de um conjunto</i> : marca o elemento como sendo um dos elementos do subconjunto sendo selecionado. <i>Nos demais casos</i> : posiciona o cursor do editor sobre o cursor do “mouse”. |
| {CAN} | ambos os botões esquerda e direita pressionados simultaneamente. Cancela a operação corrente. |

3.4.3 Botões com movimentação

Os comandos de “mouse” a seguir são operados:

- pressionar o botão da direita,
 - mover o “mouse” na direção desejada e, após,
 - soltar o botão.
- | | |
|-------------|--|
| {DIR esq} | <i>ao editar valores ou textos</i> : vai para a palavra ou campo à esquerda. <i>Ao editar diagramas</i> : posiciona a janela para a esquerda. O número de colunas reposicionado é definido pelo parâmetro <i>salta coluna janela</i> do conjunto de parâmetros do editor de diagramas, ver capítulo Editor de Diagramas. |
| {DIR dir} | <i>ao editar valores ou textos</i> : vai para a palavra ou campo à direita. <i>Ao editar diagramas</i> : posiciona a janela para a direita. O número de colunas reposicionado é definido pelo parâmetro <i>salta coluna janela</i> do conjunto de parâmetros do editor de diagramas, ver capítulo Editor de Diagramas. |
| {DIR sobe} | posiciona a janela para acima. |
| {DIR desce} | posiciona a janela para baixo. |

Página propositalmente deixada em branco

4. MENU PRINCIPAL

O menu principal coordena a operação do ambiente Talisman. A figura 3.1 mostra o seu conteúdo. Nas seções a seguir descreveremos os itens deste menu.

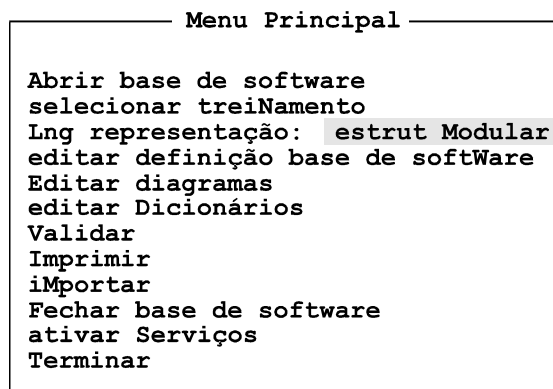


Figura 1.1 Menu Principal

4.1 Abrir base de software

Este item dispara a abertura base de software. Ao abrir a base serão solicitados:

- i. nome do arquivo da base.
- ii. nome do arquivo destino da cópia backup da base.
- iii. opcionalmente, nome do arquivo LOG associado à base.

Para fornecer o nome do arquivo, siga as instruções da seção **Selecionar Arquivo** mais adiante neste capítulo.

Antes de abrir a base, é verificado se o arquivo é uma base de software válida e consistente com o sistema de proteção. Caso uma das duas condições não se verifique, o processo de abertura da base será cancelado.

Caso o arquivo seja uma base de software válida, ele será aberto. Logo a seguir será solicitado o nome do arquivo destino de “backup”. O processo de seleção do arquivo “backup” é idêntico ao utilizado para abrir a base. Não sendo desejado fazer um “backup” da base, tecle [ESC].

Durante a edição e/ou exploração de uma base de software, pode-se saber o nome do arquivo correspondente, examinando a **Identificação da Base**, ver seção mais adiante.

ATENÇÃO: A base de software é um banco de dados e é atualizado diretamente. Durante o processamento podem ocorrer diversos acidentes capazes de destruir a base em edição. É conveniente, então, possuir cópias “backup” da base. Crie cópias “backup” com regularidade. Isto é uma proteção para o seu trabalho! Podem ser criadas cópias “backup” da base também durante o uso, veja a seção **Copiar Arquivos** do capítulo **Serviços**.

Caso o sub-sistema *Log* esteja ativo, ver **Serviço LOG**, será solicitado o nome do arquivo LOG.

4.2 Selecionar treinamento

Este item dispara um menu de seleção de lições. Lições destinam-se a auxiliar no aprendizado do uso de Talisman.

O capítulo **Auxílio e Treinamento** descreve em detalhe como operar lições.

4.3 Seleção da linguagem de representação corrente

O item **Lng representação** é utilizado para selecionar a linguagem de representação corrente. Esta poderá ser selecionada somente no **Menu Principal**. No entanto, por intermédio de navegações, pode-se navegar para elementos de outras linguagens de representação sem precisar selecionar explicitamente a nova linguagem de representação.

Teclando <ENTER> neste item, serão exibidas todas as linguagens suportadas pela instalação. Selecione a linguagem desejada e prossiga. Pode-se, também, mudar de uma linguagem para outra, teclando <SP>. A linguagem de representação corrente estará sempre exibida no **Menu Principal**.

4.4 Identificação da base de software

Este item dispara a edição do formulário de *identificação da base*. Este se destina a conter informações relativas à base de software como um todo. São exemplos: a lista de autores que editaram a base, a lista de alterações pendentes, o nome do arquivo da base de software, etc.

A edição da identificação da base é realizada por intermédio dos formulários definidos na linguagem **Identificação da Base** descrita no capítulo de mesmo nome. Para ativar a edição da identificação da base através deste item do menu principal, não é necessário mudar a linguagem de representação corrente.

A edição da identificação da base pode ser realizada, ainda, via **Editor de Dicionários**. Para isto é necessário selecionar a linguagem **Identificação da Base** e ativar o **Editor de Dicionários**.

4.5 Editar diagramas

Este item dispara a seleção, para posterior edição, dos diagramas definidos na linguagem de representação corrente.

A edição de diagramas está descrita no capítulo **Editor de Diagramas**. A edição de diagramas pode ser disparada, ainda, via o **Editor de Dicionários**.

4.6 Editar dicionários

Este item dispara o **Editor de Dicionários**, descrito no capítulo de mesmo nome.

O editor de Dicionários é a peça central para a edição e exploração da base de software. Através do editor de Dicionários pode-se:

- i. criar e excluir objetos.
- ii. editar formulários de especificação.
- iii. editar estruturas.
- iv. editar diagramas.
- v. validar o conteúdo da base de software.
- vi. gerar código através de linearizadores.
- vii. transformar representações em outras.
- viii. imprimir formulários.
- ix. exportar parte ou todo o conteúdo da base de software para outros sistemas.

4.7 Validar

Ativa a validação de elementos selecionados ou de Dicionários inteiros. Embora seja possível validar via o Editor de Dicionários, o uso deste item é mais indicado quando se deseja efetuar validações de partes significativas da base de software.

A validação será realizada utilizando o formulário *default* de cada classe de objeto sendo validada. O capítulo **Programação de Formulários** descreve como programar e configurar validadores.

4.8 Imprimir

Ativa a impressão de elementos selecionados, de Dicionários inteiros, ou de diagramas. Embora seja possível imprimir via o Editor de Dicionários, o uso deste item é mais indicado quando se deseja efetuar impressões de porções significativas da base de software.

A impressão será realizada utilizando o formulário *default* de cada classe de objeto sendo validada. O capítulo **Programação de Formulários** descreve como programar e configurar os geradores de relatórios.

Antes de imprimir, certifique-se que o sistema está corretamente configurado. A configuração de impressora, do arquivo *spool*, dos formatos de relatórios e do número da página inicial, são realizados através do menu **Serviços** descrito no capítulo de mesmo nome.

4.9 Importar

Ativa o sub-sistema de importação que torna possível receber dados exportados de outras base de software, ou gerados por outras ferramentas. Desta forma pode-se estabelecer e manter padrões, bem como trabalhar em equipe.

O sub-sistema de importação está descrito no capítulo **Importação e Exportação**.

4.10 Ativar Serviços

Ativa o sub-sistema de **Serviços**, que permite configurar o ambiente, configurar a estação de trabalho, efetuar serviços utilitários, etc.

O sub-sistema **Serviços** é descrito no capítulo de mesmo nome.

4.11 Seleção de arquivos

A seleção de arquivos segue sempre o mesmo padrão de interface. Inicialmente é executado um diálogo para a edição do nome do arquivo. Forneça o nome do arquivo desejado. Este nome pode conter caracteres “*” e/ou “?” no NomeArquivo e/ou no NomeExtensão, ver abaixo. Cabe frisar que os nomes de arquivos em Talisman devem ser sempre nomes completos, formados por:

Drive : \ Diretório \ NomeArquivo . NomeExtensão

Caso o nome editado contenha caracteres “*” ou “?”, será ativado um menu de seleção de arquivos. Este menu contém os nomes de todos os arquivos e subdiretórios que satisfazem o nome e que estão contidos no diretório definido. Os nomes de diretórios, caso existam, terminam com o caractere “\”.

Caso o nome do arquivo principie com o caractere ‘!’, este será substituído pela definição do diretório *default* para o tipo de arquivo. Após a substituição, o diálogo permite continuar editando. Para definir diretórios *default*, leia a seção: 9.2.2 *Configurar diretório*.

Pode-se repetir a busca do arquivo navegando sobre diretórios, utilizando o menu de seleção descrito acima. Caso seja selecionado um arquivo existente este será o arquivo a ser aberto. No entanto, antes de abrir, o arquivo é validado (por exemplo: é base de software, é versão de base compatível com o sistema, etc.). Não correspondendo a um arquivo válido, será emitida uma mensagem de erro, e continua-se no processo de abrir ou selecionar arquivo. Observação, para navegar o sistema de arquivos em busca de um nome de arquivo específico, forneça o nome completo do arquivo procurado, sucedendo-o pelo caractere “*”.

Este processo (edição de nome, seleção em menu de nomes) continuará até que, ou seja fornecido um nome de arquivo válido, ou a seleção de arquivo seja cancelada com um comando [ESC].

Página propositalmente deixada em branco

5. EDITOR DE DICIONÁRIOS

O editor de dicionários é utilizado para ativar *ações de usuário* com relação a objetos selecionados de dicionários. A figura 5.1 ilustra a interação do editor de dicionários com os demais editores. Ela ressalta o papel de controle do ambiente exercido pelo editor de dicionários.

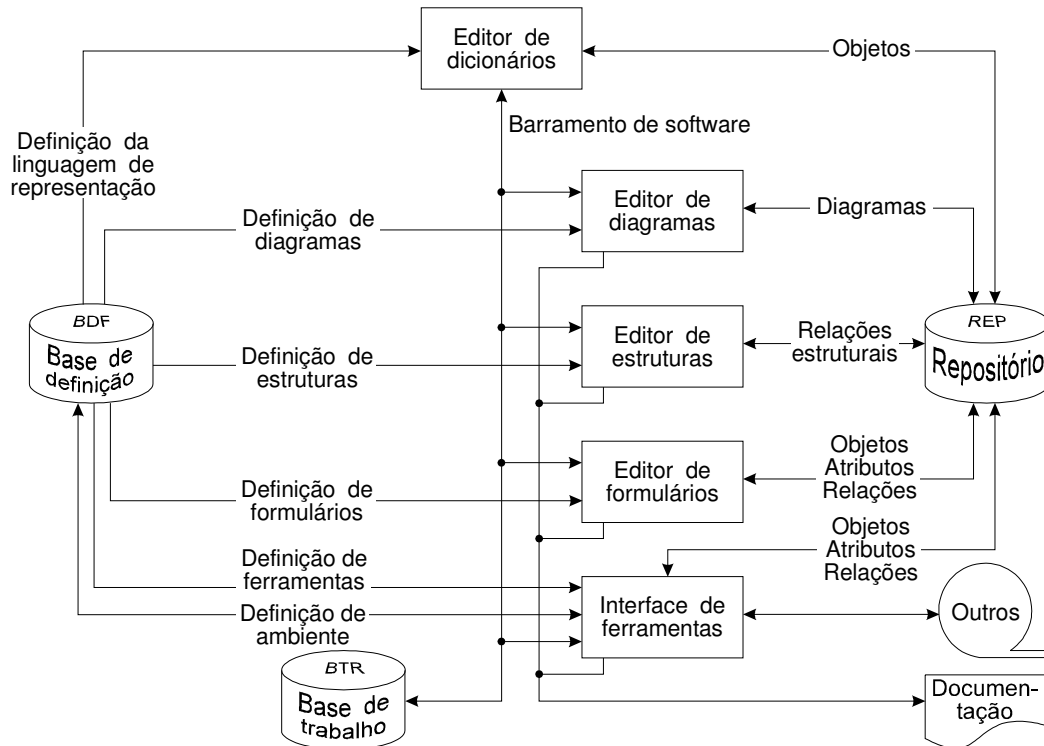


Figura 5.1. Interação do editor de dicionários com outros editores

5.1 Seleção de dicionário

A figura 5.2 ilustra o procedimento de seleção do conjunto de objetos e da ação sobre este conjunto. O editor de dicionários pode ser ativado a partir dos demais editores (comandos [D], e de navegação), bem como a partir do **Menu Principal**. Os dicionários selecionáveis em um dado momento dependem da linguagem de representação corrente. A linguagem de representação corrente é selecionada sempre no **Menu Principal**.

Ao iniciar o editor de dicionários, é ativado o menu de seleção das *classes de objetos* definidos na linguagem de representação corrente. Cada uma destas classes corresponde a um dicionário. Uma vez selecionado o dicionário, é ativado o menu de seleção dos objetos deste dicionário. As ações do usuário podem ser realizadas com relação a objetos individuais, ou, então, com relação a subconjuntos de objetos.

Para marcar os objetos de um subconjunto, utilize um dos comandos:

- | | |
|---------|---|
| [SP] | marca/desmarca o objeto corrente, ver capítulo Menus . Podem ser marcados quantos objetos se queira. O processamento dos objetos seguirá a ordem estabelecida pela marcação. |
| [ENTER] | marca o objeto corrente, termina a seleção do subconjunto, e ativa o menu de seleção da ação a executar. Note que, o objeto marcado com [ENTER] será o último elemento do subconjunto de elementos. |
| [F6] | |
| [ESC] | cancela a seleção independentemente do número de objetos marcados. |
| [F3] | |
| [^F3] | |

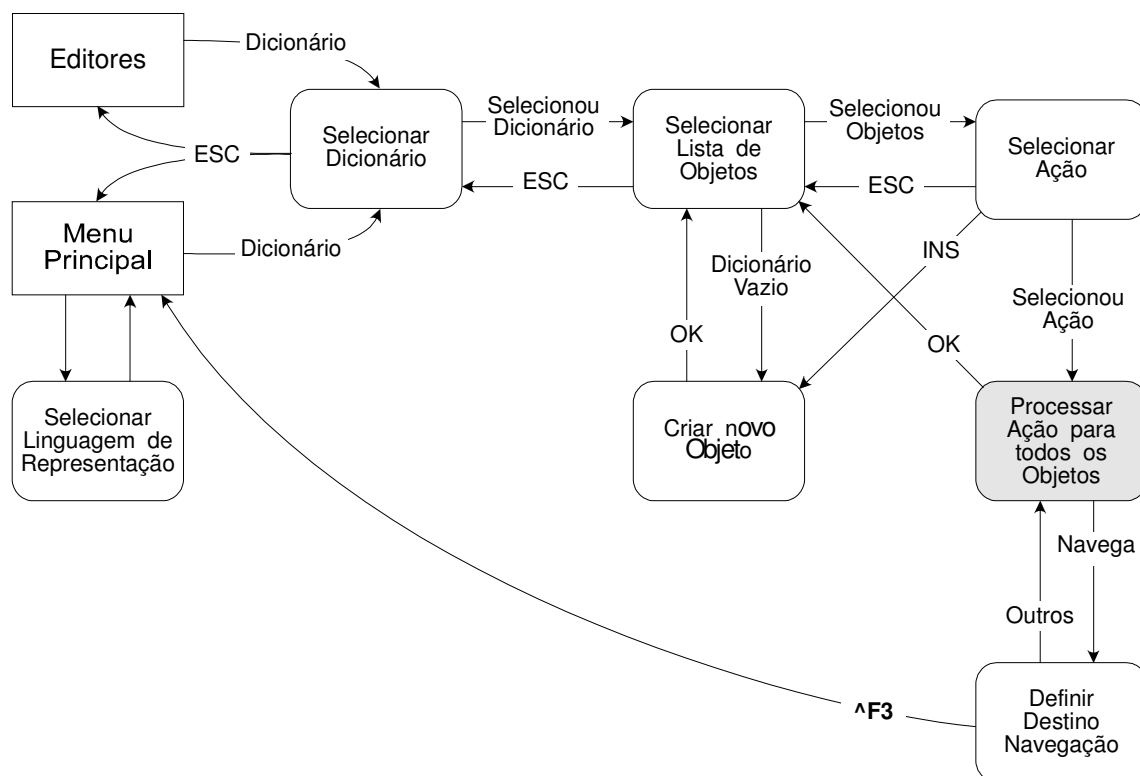


Figura 5.2. Procedimento de seleção de ação com o editor de dicionário

5.2 Ações sobre os objetos selecionados

Ações sobre objetos		
Editar	Operações	Imprimir
eStrutura Diagrama Especific. [F4] sel. esPecific.	Validar sel. vAlidar Linearizar sel. linearizar Transformar sel. transfOrm.	estRutura diaGrama espeCificação sel. iMpressão
Inserir [INS] edt. Nome [F5] eXcluir [DEL]	> exportar] sel. exportar	# serviços

Figura 5.3. Menu de ações sobre objetos selecionados

Cada um dos objetos do subconjunto selecionado pode ser submetido a uma, entre várias, ações. Estas ações são disparadas por botões do menu. Alguns dos botões estarão acessíveis diretamente, outros são acessíveis somente através do menu de botões. O menu de botões é ativado pelo botão [F6].

A figura 5.3 apresenta o menu das ações que podem ser aplicadas aos objetos de um conjunto de objetos selecionados em um determinado dicionário.

Inserir [INS] insere um novo objeto no dicionário. Para editar o nome, utiliza o Formulário associado à função **Criar/Editar nome** da correspondente classe de objeto. A programação de Formulários, bem como o procedimento de associação de Formulários *default* às ações, estão descritos no capítulo **Programação de Formulários**. Cabe salientar que a ação inserir pode ser disparada diretamente pelo botão [INS].

Editar nome [F5] edita o nome do objeto, utilizando o Formulário *default* associado à ação **Criar/editar nome** da correspondente classe de objeto.

Editar especificação [F4] edita a especificação do objeto, utilizando o formulário *default* associado à função **Editar** da correspondente classe de objeto.

- Editar especificação selecionada** edita a especificação do objeto, utilizando um formulário selecionado. Cabe salientar que formulários são tornados selecionáveis, caso sejam definidos com **InicExterno** ao invés de **InicFrm**, ver capítulo **Programação de Formulários**
- Excluir objeto [DEL]** confirma um a um os objetos marcados à medida que vão sendo processados. As respostas a estas confirmações podem ser:
- Não exclui** não realiza a exclusão do objeto selecionado
 - Só este** exclui somente o objeto, sem propagar a exclusão para objetos por ele referenciados.
 - Dependentes** exclui o objeto corrente. Exclui, ainda, objetos dependentes, caso somente possuam relacionamentos dependentes definidos na linguagem de representação corrente. São exemplos de relações dependentes *Decomposição*, *Corpo de pacote*, *Corpo de tipo*.
 - [ESC]** cancela o processamento do objeto corrente e do restante do subconjunto.
- Editar diagrama** apresenta a lista dos diagramas nos quais o objeto é instanciado. Esta lista exibir tantos elementos quantas forem as instâncias. Conseqüentemente, existirão elementos repetidos na lista sempre que o objeto em questão possuir mais do que uma instância em um diagrama.
- Editar estrutura** edita o objeto, utilizando o **Editor de Estrutura**, ver capítulo de mesmo nome.
- Validar** valida os objetos selecionados, utilizando o Formulário *default* associado à ação **Validar** da correspondente classe de objetos.
- Validar selecionado** valida os objetos selecionados, utilizando um Formulário também selecionado. Os Formulários selecionáveis são definidos no programa de Formulário usando a declaração **InicExterno**, ver capítulo **Programação de Formulários**.
- Linearizar** lineariza os objetos selecionados, utilizando o Formulário *default* associado à função **Linearizar** da correspondente classe de objetos. O texto linearizado poder ser dirigido para um arquivo, para o editor de Formulários ou para a impressora. A pergunta do destino da linearização é repetida para cada objeto a ser processado.
- Linearizar selecionado** lineariza os objetos selecionados, utilizando um Formulário selecionado pelo usuário. Os Formulários selecionáveis são definidos no programa de Formulário usando a declaração **InicExterno**, ver capítulo **Programação de Formulários**. O texto linearizado poder ser dirigido para um arquivo, para o editor de Formulários ou para a impressora. A pergunta do destino da linearização é repetida para cada objeto a ser processado.
- Transformar** transforma os objetos selecionados, utilizando o Formulário *default* associado à função **Transformar** da correspondente classe de objetos. A transformação processa e altera o conteúdo da base de software.
- Transformar selecionado** transforma os objetos selecionados, utilizando um Formulário selecionado pelo usuário. Os Formulários selecionáveis são definidos no programa de Formulário usando a declaração **InicExterno**, ver capítulo **Programação de Formulários**.
- Exportar** exporta os objetos selecionados, utilizando o Formulário *default* associado à função **Exportar** da correspondente classe de objetos. O texto exportado poder ser dirigido para um arquivo, para o editor de Formulários ou para a impressora. A pergunta do destino da exportação é repetida para cada objeto a ser processado. Detalhes sobre **Importação e Exportação** podem ser encontrados no capítulo de mesmo nome.
- Exportar selecionado** exporta os objetos selecionados, utilizando um Formulário selecionado pelo usuário.
- Imprimir estrutura** imprime o diagrama de estrutura em que figura o objeto selecionado. Deve-se assegurar que a configuração de impressora, relatório e número da página esteja correta antes de selecionar esta ação.
- Imprimir diagrama** imprime o diagrama em que o objeto selecionado está instanciado. Deve-se assegurar que a configuração de impressora, relatório e número da página esteja correta antes de selecionar esta ação.
- Imprimir especificação** imprime os objetos selecionados, utilizando o Formulário *default* associado à função **Imprimir** da correspondente classe de objetos. Deve-se assegurar que a configuração

ração de impressora, relatório e número da página esteja correta antes de selecionar esta ação.

Imprimir selecionado imprime os objetos selecionados, utilizando um Formulário selecionado pelo usuário. Os Formulários selecionáveis são definidos no programa de Formulário usando a declaração `InicExterno`, ver capítulo **Programação de Formulários**. Deve-se assegurar que a configuração de impressora, relatório e número da página esteja correta antes de selecionar esta ação.

Serviços ativa o menu **Serviços** descrito no capítulo de mesmo nome. Serviços são utilizados para configurar o ambiente e a estação de trabalho.

6. EDITOR DE FORMULÁRIOS

O editor de formulários é utilizado para as seguintes funções:

- i. *Criar/Editar* Nomes de objetos.
- ii. *Editar texto* de especificações de objetos.
- iii. *Editar texto* linearizado.
- iv. *Editar texto* exportado.
- v. *Editar texto* em geral, por exemplo: macros, janelas explanatórias, programas de formulário, arquivos sequenciais, etc.

Observação tendo em vista a difusão de editores que seguem o padrão de comandos estabelecido por WORDSTAR e TURBO, este padrão também poderá ser utilizado ao editar formulários. No entanto, o presente manual descreverá os comandos somente segundo a sintaxe estabelecida em Talisman.

6.1 Conceitos básicos

O editor de formulários é dirigido por programas de formulários, ver capítulo **Programação de Formulários**. Mais especificamente, o editor de formulários edita *formulários editáveis*. Um formulário editável é uma coletânea de *títulos* e *campos* dispostos em determinado *formato*. A composição e o formato de um formulário editável são definidos através de um *programa de formulário*. Os títulos são textos constantes (não editáveis) definidos no programa. Os campos são *atributos de objetos* ou *variáveis*. A figura 6.1 ilustra a composição de um programa de formulário com o conteúdo da base de software formando um formulário editável no vídeo.

Um formulário editável é uma *extração* (“view”) do conteúdo da base de software. A edição de formulários editáveis é semelhante à *edição de texto* convencional. Ela se diferencia pelo fato da edição de cada formulário ser dirigida por um programa específico.

Formulários editáveis são compostos por *campos* de diferentes *tipos*. Cada tipo de campo possui *regras* e *comandos de edição* específicos. As propriedades de cada tipo de campo serão descritas mais adiante neste capítulo.

Para cada classe de objeto das diversas linguagens de representação suportadas, e para cada ação de usuário, é definido um *formulário default*. O conjunto de formulários vinculados a determinada linguagem de representação são agrupados formando o *programa de formulários* desta linguagem de representação. O capítulo **Programação de Formulários** descreve como redigir os programas de formulários e como vincular as ações de usuários a formulários *default*. A associação de ações a formulários *default* é realizada pelo serviço Configurar formulários no menu **Serviços**.

A edição de um formulário é realizada em uma área de trabalho (arquivo TALISMAN.\$\$1). As edições realizadas somente serão registradas na base de software caso o formulário seja fechado normalmente (comandos [F3] *Terminar edição de formulário*, [F4] *Navegar para outro formulário*, [F6] *Navegar para editor de estrutura*, [^F6] *Navegar para editor de diagrama*, ou [^F3] *Retornar ao Menu Principal*). Caso o formulário seja cancelado (comando [ESC]), ou caso ocorra algum acidente, por exemplo falta de energia, nada será gravado na base de software.

O programa de formulário é executado enquanto o formulário está sendo editado a medida que forem sendo solicitadas as linhas a exibir no vídeo (modo corotinado). Conseqüentemente, o término ou cancelamento de um formulário antes de se atingir o fim do programa, deixar uma parte do código sem execução. Em princípio isto não traz problemas. Somente trará caso o programa de formulário execute tarefas de término, caso em que estas tarefas não serão executadas.

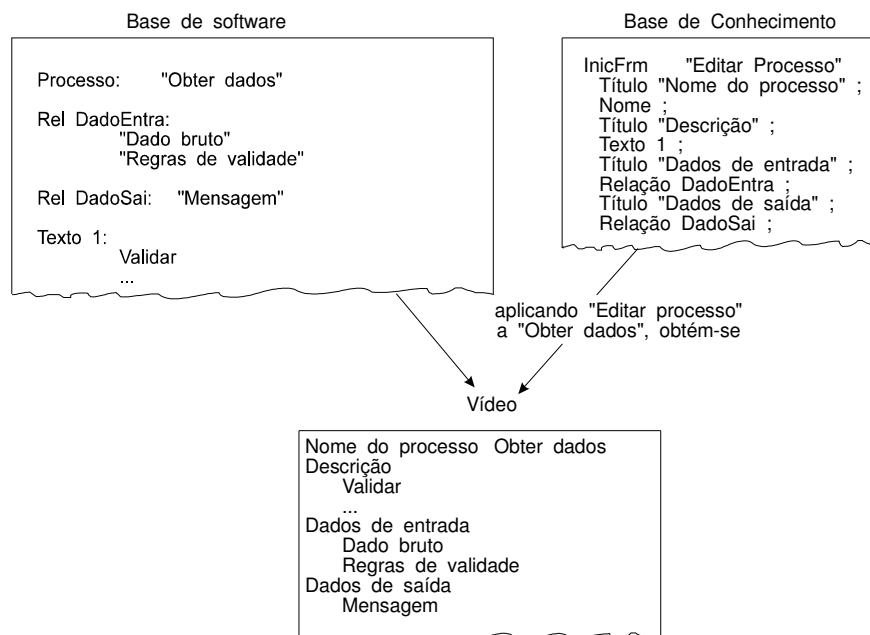


Figura 6.1. Ilustração da composição do formulário editável

6.2 Tipos de campos de formulários

O *estado* exibido na barra de “feedback” (primeira linha da tela, ver capítulo **Convenções**) indica o *tipo de campo* sendo processado pelo formulário. O resultado das operações de edição depende do tipo de campo do formulário. Acostume-se a olhar para o “feedback” de estado, isto eliminará muitas das dúvidas durante a edição.

Os estados durante a edição de formulários podem ser:

Título	o cursor se encontra sobre um campo <i>título</i> . Títulos são seqüências constantes (não editáveis) definidas no programa de formulário.
Nome	o cursor se encontra sobre um campo <i>nome de objeto</i> . Cada objeto possui um único nome. Os nomes dos objetos de uma determinada classe são agrupados em dicionários. Nomes de diferentes classes podem ser iguais. Nomes não podem ser vazios, ou seja, precisam ter pelo menos um caractere diferente de branco, podendo ter, no máximo 250 caracteres.
Aliás	o cursor se encontra sobre um campo <i>aliás</i> . Aliases podem ser vazios, podendo ter, no máximo, 250 caracteres.
Texto	o cursor se encontra sobre uma linha de um campo <i>fragmento de texto</i> . Fragmentos de texto podem ser vazios, podendo ter, no máximo, 32.000 linhas de até 250 caracteres cada uma.
Relação	o cursor se encontra sobre uma linha de um campo <i>relação</i> . Uma relação é um conjunto de referências (<i>relacionamentos</i>) a outros objetos contidos na base de software. Relações podem estabelecer referências cruzadas entre objetos, mesmo que pertençam a diferentes representações.
Var seqnc	o cursor se encontra sobre um campo <i>variável seqüencial</i> . Variáveis seqüenciais podem ser vazias, podendo ter, no máximo, 250 caracteres.
Var texto	o cursor se encontra sobre um campo <i>variável texto</i> . Variáveis texto podem ter de 0 a 32.000 linhas, cada qual com 0 a 250 caracteres.
Nome Mac	o cursor se encontra sobre um campo <i>nome de macro</i> . Cada macro de teclado possui um único nome. Nomes de macros não podem ser vazios, ou seja, precisam ter pelo menos um caractere diferente de branco, podendo ter, no máximo 250 caracteres.
Texto Mac	o cursor se encontra sobre uma linha de um fragmento de texto <i>corpo de macro</i> . Fragmentos de texto podem ser vazios, podendo ter, no máximo, 32.000 linhas de até 250 caracteres cada uma.
Vazio	o cursor se encontra sobre um campo sem valor.
Regist Blc	foi iniciada e está em progresso uma operação de registrar um bloco de texto.

- Delir Blc** foi iniciada e está em progresso uma operação de registrar um bloco de texto a ser excluído.
- Gravar Blc** foi iniciada e está em progresso uma operação de registrar um bloco de texto a ser gravado em um arquivo seqüencial.
- Adicnr Blc** foi iniciada e está em progresso uma operação de registrar um bloco de texto a ser adicionado ao final de um arquivo seqüencial.
- Imprim Blc** foi iniciada e está em progresso uma operação de registrar um bloco de texto a ser impresso.

6.3 Comandos e regras específicas

6.3.1 Regras para igualdade de nomes

Os nomes de objetos são considerados iguais de acordo com as regras a seguir:

- i. Letras maiúsculas e minúsculas, acentuadas ou não, são iguais.
- ii. Espaços antes e após o nome são descartados.
- iii. Um ou mais espaços entre os componentes de um nome são iguais a um único espaço.

6.3.2 Regras específicas para campos *NOME*

6.3.2.1 Regras de atualização de nomes

Ao *terminar* a edição de um formulário são observadas as seguintes regras para campos *NOME*:

- | | |
|---|--|
| <i>Nome vazio</i> | é emitido uma mensagem de erro. |
| <i>Nome de objeto novo diferente de todos os nomes conhecidos</i> | o objeto é criado com o nome editado. |
| <i>Nome de objeto novo igual ao nome de outro objeto existente</i> | o objeto é associado ao objeto já existente. |
| <i>Nome editado de objeto existente diferente de todos os nomes conhecidos</i> | o nome do objeto será trocado para o nome editado. |
| <i>Nome editado de objeto existente igual ao nome de outro objeto existente</i> | é emitido um erro de execução. |

6.3.2.2 Comandos específicos para campos *NOME*

- | | |
|----------|--|
| [!] | ativa um menu de seleção contendo os nomes de todos os objetos definidos no dicionário correspondente. Após selecionar um destes nomes, ele poderá ser editado. |
| [F4] | <i>em estado NOME</i> : termina o formulário sendo editado, registrando as edições na base de software. A seguir, inicia a edição do formulário do objeto cujo nome estava sendo editado. Se este for o nome do objeto sendo editado, corresponde a salvar e retomar a edição. |
| [F6] | <i>em estado NOME</i> : termina o formulário sendo editado, registrando as edições na base de software. A seguir, edita a estrutura da qual o objeto em questão é membro. |
| [^F6] | <i>em estado NOME</i> : termina o formulário sendo editado, registrando as edições na base de software. A seguir, edita um diagrama no qual o objeto em questão é instância. |
| [F8 DEL] | elimina todos os caracteres do nome, deixando-o vazio. Como nomes de objetos não podem ser vazios, este comando deve ser usado somente para posterior digitação de nome novo. |

6.3.3 Regras específicas para campos *RELAÇÃO*

6.3.3.1 Regras de atualização de relações

A edição de um campo *RELAÇÃO* é idêntica à edição de texto, sendo que cada linha de um campo relação é o nome de um objeto com o qual o objeto corrente possui um relacionamento.

Ao *terminar* a edição de um formulário são observadas as seguintes regras para campos *RELAÇÃO*:

cada linha do campo é um nome com o qual está ou será estabelecido um relacionamento.

*linha contém nome **editado** diferente dos nomes existentes no dicionário correspondente* o nome do objeto é ajustado para o nome editado.

*linha contém nome **editado** igual ao nome de um outro objeto existente no dicionário* é emitido uma mensagem de erro.

*linha contém nome **novo** diferente dos nomes existentes no dicionário correspondente* cria um novo objeto com este nome e estabelece um relacionamento com este objeto.

*linha contém nome **novo** igual a nome existente no dicionário correspondente* estabelece um relacionamento com o objeto designado por este nome.

linha vazia a linha é ignorada.

Cuidado caso seja digitado acidentalmente um texto corrido em um campo **RELAÇÃO**, cada linha deste texto será tratada como um nome de um objeto. Ao fechar o formulário os objetos correspondentes a estes nomes serão criados no respectivo dicionário. Para limpar os dicionários assim poluídos, ative o **Editor de Dicionários**. Neste editor selecione o dicionário que foi poluído e elimine todos os objetos indevidamente criados nele.

6.3.3.2 Comandos específicos para campos **RELAÇÃO**

[!] ativa um menu de seleção contendo os nomes de todos os objetos definidos no dicionário correspondente. Serão inseridos na relação todos os zero ou mais nomes selecionados. Os nomes inseridos são tratados como nomes novos na relação. Após a seleção, eles poderão ser editados.

[F4] *em estado **RELAÇÃO***: termina o formulário sendo editado, registrando as edições na base de software. A seguir, inicia a edição do formulário do objeto do relacionamento corrente ao dar o comando. Desta forma, consegue-se *explorar* a base de software numa modalidade parecida com a de hipertextos.

[F6] *em estado **RELAÇÃO***: termina o formulário sendo editado, registrando as edições na base de software. A seguir, edita a estrutura da qual o objeto destino do relacionamento corrente é membro.

[^F6] *em estado **RELAÇÃO***: termina o formulário sendo editado, registrando as edições na base de software. A seguir, edita um diagrama no qual o objeto destino do relacionamento corrente é uma instância.

6.4 Controles do editor de formulários

Os controles de edição podem ser alterados por comandos de teclado, descritos abaixo, ou ao editar os parâmetros de controle (comando [F2]).

6.4.1 Modo de inserção de caracteres

Inser introduz os caracteres digitados na posição do cursor, deslocando o resto da linha para a direita.

Sobre superimpõe os caracteres digitados na posição do cursor.

[INS] inverte o modo de inserção.

6.4.2 Modo de endentação do início de linhas

Endenta o primeiro caractere de nova linha de texto ser alinhado com o primeiro caractere não branco na linha de texto acima.

Não endenta o primeiro caractere de nova linha de texto ser alinhado na primeira coluna do campo.

[!E] inverte o modo de endentação.

6.4.3 Modo de acentuação

Acentua interpreta as regras de acentuação gerando caracteres acentuados.

Não acentua não interpreta as regras de acentuação.

[!A] inverte o modo de acentuação.

6.4.3.1 Regras para a acentuação

- para digitar uma letra acentuada, primeiro digite o acento e, depois, digite a letra a acentuar.
- para acentuar uma letra já digitada, acentuada ou não, posicione o cursor no caractere e digite o acento.
- para remover o acento, remova a letra acentuada e introduza a letra não acentuada.

Os caracteres acentuados aceitos pelos editores TALISMAN são:

' (agudo)	com a, e, i, o, u, c, E, C gera á, é, í, ó, ú, ç, É, Ç
~	com a, o gera ã, õ
^	com a, e, o gera â, ê, ô
` (crase)	com a gera à
"	com u gera ü

6.5 Posicionamento do cursor e da janela

[setas]	movimentam o cursor na direção desejada.
[^ESQ]	move o cursor para o início da palavra à esquerda. Esta pode estar na linha ou no campo anterior.
[^DIR]	move o cursor para o início da palavra à direita. Esta pode estar na linha ou no campo a seguir.
[HOME]	posiciona o cursor no primeiro caractere da linha (ou campo), rolando horizontalmente a janela.
[END]	posiciona o cursor imediatamente após o último caractere não branco da linha (ou campo), rolando a janela horizontalmente. Caso o tamanho do valor do campo seja maior do que o tamanho do campo, ocorrerá rolagem somente na região visível do campo.
[PGUP]	reposiciona a janela <i>rola janela</i> linhas para cima. Este parâmetro é definido no conjunto de parâmetros do editor de formulários (comando [F2]).
[PGDN]	reposiciona a janela <i>rola janela</i> linhas para baixo.
[^PGUP]	posiciona o cursor no início do primeiro campo do formulário.
[^PGDN]	posiciona o cursor após o último caractere da última linha do último campo do formulário.
[↑ TAB]	tabula para a esquerda parando no início da linha ou no início da primeira palavra à esquerda, considerando-se a linha acima.
[TAB]	tabula para a direita posicionando-se início da primeira palavra à direita considerando a linha acima. Em modo inserção, desloca o texto do cursor para a direita, preenchendo o espaço com caracteres brancos.
{ESQ}	posiciona o cursor de edição sobre o cursor do “mouse”.
{DIR}	posiciona o cursor de edição sobre o cursor do “mouse”. a seguir ativa a régua de comandos.

6.6 Inserir linhas

[ENTER] *se modo for insere e campo for TEXTO editável:* cria uma linha a seguir da linha onde está o cursor. Caso o cursor esteja no meio da linha, quebra a linha.
demais casos: vai para a linha a seguir.

Observação Campos editáveis compostos por diversas linhas (TEXTO ou RELAÇÃO) são terminados pelo marcador ◆. Quando estes campos estiverem vazios (existe somente o marcador), o primeiro comando de edição deverá ser [ENTER] para que seja criada a primeira linha.

6.7 Excluir linhas ou caracteres

- [BKSP] elimina o caractere à esquerda do cursor.
- [DEL] elimina o caractere sob o cursor.
- [F8 P] *estando sobre caracteres delimitadores* elimina todos os caracteres delimitadores de palavras até o próximo caractere não delimitador. Os caracteres delimitadores são definidos pelos parâmetros do editor de formulários (comando [F2]). *Estando sobre caracteres não delimitadores* elimina a palavra à direita do cursor. *Estando no fim da linha* concatena a linha a seguir ao final da linha do cursor.
- [F8 DEL] elimina todos os caracteres do cursor, inclusive, até o fim da linha ou campo, sem eliminar a linha.
- [F8 L] *em campos NOME, ALIÁS, VAR SEQ ou NOME MAC* elimina todos os caracteres do campo. Lembre-se que nomes não podem ficar vazios, porém aliases podem. Um campo vazio é identificado pelo marcador ♦. *Demais campos* elimina toda a linha do cursor, inclusive os caracteres anteriores ao cursor.

6.8 Operações de troca e procura

As operações de troca e procura percorrem o formulário procurando seqüências iguais à fornecida. A pesquisa sempre inicia no local onde se encontra o cursor e prossegue em direção do final do formulário.

A comparação poderá ser *sensível* ou *insensível* à caixa e acentuação. Caso a opção escolhida seja insensível, letras maiúsculas e minúsculas, acentuadas ou não, serão consideradas iguais. Ao definir a seqüência a ser procurada, pode-se escolher as classes de campos a serem examinadas.

- [F7 P] procura uma seqüência de caracteres no formulário. Pára ao final da seqüência encontrada, ou no final do formulário.
- [F7 T] procura uma seqüência. Caso seja encontrada, troca-a por outra. Esta operação pode ser efetuada em modo repetitivo.
- [F7 C] continua a última operação de troca ou procura definida. A continuação pode ser realizada mesmo que se tenha mudado de formulário editável, de base de software ou de arquivo seqüencial em edição.

6.9 Operações com blocos de texto

6.9.1 Regras gerais para blocos de texto

Talisman opera com um *bloco corrente*. O bloco corrente é semelhante a um *clip board* e é composto por zero ou mais linhas e estará registrado em uma área de trabalho específica. As linhas do bloco corrente são tratadas como linhas de texto normal. O bloco corrente é sempre uma cópia de um pedaço de algum formulário editável.

As operações de *registro* e *eliminação* de bloco criam o bloco corrente descartando o bloco corrente porventura existente. Uma vez registrado, o bloco corrente continuará existindo até que um novo bloco corrente seja registrado, mesmo que seja iniciada a edição de um outro formulário, ou até de outra base de software. A operação inserção de bloco insere o bloco corrente no formulário sendo editado. Este mecanismo serve para copiar textos entre formulários, bases e arquivos seqüenciais sendo editados.

6.9.2 Marcação de bloco

O *bloco marcado* é uma porção do formulário sendo editado contido entre *delimitadores*. Qualquer campo, inclusive TÍTULO e VAZIO, poderá servir como delimitador de início ou fim de bloco. Em campos com várias linhas (TEXTO ou RELAÇÃO), qualquer linha pode servir como delimitador de bloco.

Os campos são tratados como unidades completas, independentemente da posição do cursor no campo. Caso uma linha na tela contenha vários campos, a marcação levará em conta somente o campo onde se encontra o cursor. Caso um campo contenha várias linhas (TEXTO ou RELAÇÃO) a marcação levará em conta a linha em que se encontra o cursor.

Delimitador inicial é o campo de menor índice do bloco. Ele sempre participa do bloco marcado.

Delimitador final é o campo de maior índice do bloco. Ele nunca participa do bloco marcado, servido somente como delimitador.

Um *bloco marcado* inicia no delimitador inicial inclusive e termina na linha ou campo imediatamente anterior ao delimitador final. A marcação poderá ocorrer em qualquer sentido, seja do delimitador inicial para o final, seja no sentido inverso.

6.9.3 Término da marcação de bloco

Ao ativar um comando que requeira um bloco marcado, o campo ou linha onde se encontra o cursor é um dos delimitadores. Para determinar o outro delimitador, posicione o cursor nele, utilizando para isto qualquer comando de posicionamento. Termine a marcação com:

[ENTER] marca a posição do segundo delimitador.

[F9 END] marca a posição do segundo delimitador, mesmo que acima.

[ESC] cancela a marcação do bloco e a operação com blocos.

6.9.4 Operações com bloco corrente

[F9 ENTER] inicia a marcação do bloco. Uma vez o bloco marcado, copia-o para a área de bloco corrente. O texto do formulário não é afetado. Quaisquer edições realizadas no formulário após o registro do bloco, não serão registradas no bloco corrente. Utilize este comando para copiar e colar.

[F9 DEL] inicia a marcação do bloco. Terminada a marcação, exclui o bloco. O bloco a ser excluído pode conter somente linhas de um único campo *TEXTO* ou *RELAÇÃO*. O bloco excluído passa a ser o bloco corrente. Utilize este comando para cortar e colar.

[F9 INS] *posicionado em uma linha de campo TEXTO ou RELAÇÃO*: insere o bloco corrente *antes* desta linha. *Posicionado em uma linha de campo NOME ou ALIÁS*: insere somente a primeira linha do bloco corrente no campo corrente.

Para mover blocos (cortar e colar), efetue o procedimento:

- registre o bloco corrente com o comando [F9 DEL].
- posicione-se no local de inserção e insira o bloco corrente com o comando [F9 INS].

Para copiar blocos (copiar e colar), efetue o procedimento:

- registre o bloco corrente com o comando [F9 ENTER].
- posicione-se no local de inserção e insira o bloco corrente com o comando [F9 INS].

Pode-se inserir o mesmo bloco tantas vezes quantas se queira.

É permitido que blocos copiados de uma representação, base de software, ou arquivo, sejam incluídos em outra representação, base de software, ou arquivo.

6.9.5 Ler, gravar e imprimir blocos

[F9 L] solicita o nome de um arquivo de texto ASCII (diretório *default Texto*, ver *Configuração de diretórios em Serviços*) e lê o conteúdo deste arquivo para a linha anterior à do cursor. O bloco corrente não é afetado.

[F9 G] inicia a marcação do bloco. Uma vez marcado, solicita o nome de um arquivo de texto ASCII (diretório *default Texto*), gravando nele o conteúdo do bloco corrente. O arquivo anterior, caso exista, terá sua extensão renomeada para *.BAK*.

[F9 A] o mesmo que *Gravar bloco*, sendo que o bloco será adicionado ao final do arquivo.

[F9 I] inicia a marcação do bloco. Uma vez marcado, imprime o bloco marcado utilizando a impressora e o descritor de relatório correntes (ver *Serviços*).

6.10 Terminar a edição de formulário

- [F3] fecha o formulário, registrando todas as alterações na base de software. A edição de um formulário se dá em uma área de trabalho. As alterações realizadas somente serão registradas na base de software se o formulário for fechado (comandos [F3], [^F3], [F4], [F6], ou [^F6]). Os comandos [F4], [F6] e [^F6] navegam para outro editor ou formulário. Estes comandos estão descritos nas seções que tratam de campos *NOME* e *RELAÇÃO* no presente capítulo.
- [^F3] termina o formulário sendo editado, registrando as edições na base de software. A seguir, termina todas as edições em andamento, retornando ao *menu principal*.
- [F4] navega para o formulário de especificação gerado a partir do objeto contendo o campo corrente.
- [F6] navega para a estrutura contendo o bloco de estrutura correspondente ao objeto contendo o campo corrente.
- [^F6] navega para um diagrama contendo uma instância do objeto que contém o campo corrente. Esta navegação somente poderá acontecer se a linguagem que contém o objeto destino for uma linguagem gráfica e este objeto já possui pelo menos uma instância.
- [ESC] após confirmação, cancela o formulário, descartando todas as alterações realizadas.

6.11 Comandos diversos

6.11.1 Caracteres especiais

- [^F2] ativa um menu para a seleção de caracteres especiais. Os caracteres especiais são os caracteres que não podem ser gerados diretamente pelo teclado. O caractere escolhido será inserido no texto sendo editado. Antes de imprimir, verifique se a impressora é capaz de imprimir os caracteres especiais inseridos.

6.11.2 Parâmetros de controle do editor de formulários

- [F2] ativa o menu de parâmetros do editor de formulários. Os parâmetros editáveis são:
- Linha, Coluna redefinem a linha e a coluna onde deverá ficar o cursor. A linha poderá ser definida nos formatos:
- +nnn avança nnn linhas.
 - nnn recua nnn linhas.
 - nnn posiciona na linha nnn.
- Margem direita lógica redefine a coluna direita limite de textos sendo editados. Em campos tipo TEXTO, ao atingir a margem direita lógica, é criada uma nova linha e a palavra corrente é movida para esta linha. Não é realizado o alinhamento da margem direita.
- Linhas rola define o número de linhas que serão roladas ao efetuar os comandos [PGUP] ou [PGDN].
- Caracteres delimitadores é o conjunto de caracteres que determinam início e fim de palavras para os comandos de salto sobre palavras, eliminação de palavras, tabulação e quebra de linha em margem direita lógica.
- Acentuação, Inserção, Endentação valores seleccionáveis definindo os modos de acentuação inserção e endentação (troca com [SP]). Estes controles podem ser trocados (invertidos), respectivamente, com os comandos: [!A], [INS] ou [!E].

Página propositalmente deixada em branco

7. EDITOR DE ESTRUTURAS

O editor de estruturas edita *estruturas hierárquicas*. Estruturas hierárquicas são compostas por objetos de uma mesma classe, onde estes objetos são relacionados através de *decomposição* e *composição* (relações **Decomp** e **Comp**). Estruturas hierárquicas são muito comuns, em particular ao editar programas, definições de dados, documentos, objetos que admitem decomposição, etc.

O editor de estruturas de Talisman é um editor genérico. Ele é capaz de editar estruturas de diversas naturezas. Pode-se editar, entre outras:

Estruturas de programas usando a linguagem de representação **Estrutura Modular**. As estruturas geradas nestas linguagem podem ser linearizadas, produzindo programas compiláveis. Por exemplo, o linearizador CLIN, descrito no capítulo Linguagem **Estrutura Modular**, gera programas C a partir de estruturas modulares.

Estruturas de atividades usando a linguagem de representação **Gerência de Projetos**.

Estruturas de documentos usando a linguagem de representação **Documentação**.

O editor de estruturas pode ser utilizado, ainda, para editar hierarquias genéricas. Por exemplo, um organograma pode ser entendido como sendo uma estrutura formada por *entidades externas*. Assim, utilizando a linguagem de representação **Fluxo de Dados**, pode-se criar organogramas, editando uma composição de entidades externas. Os organogramas, por sua vez, podem ser vinculados a diagramas de fluxos de dados em variados níveis de decomposição. A figura 7.1 ilustra esta interação de estruturas genéricas, formadas por diversos diagramas.

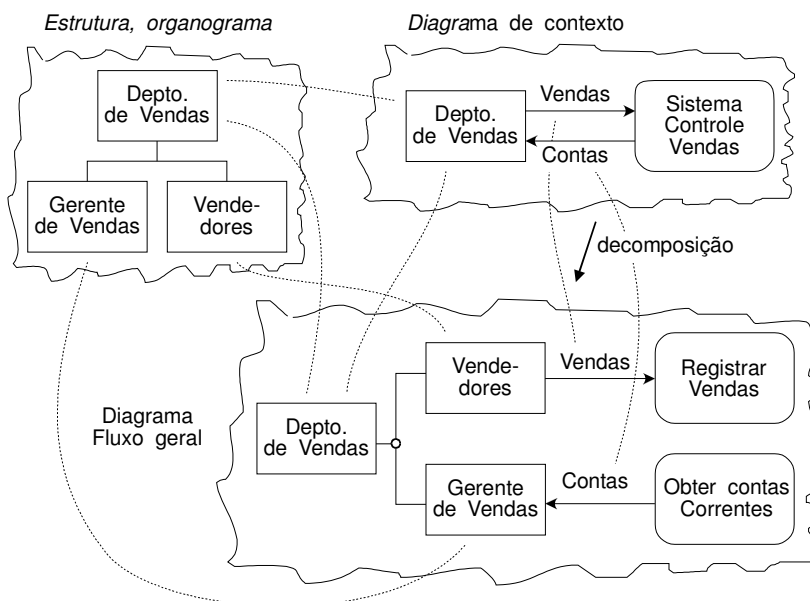


Figura 7.1. Ilustração do uso de estruturas com outros diagramas

7.1 Conceitos básicos

Nesta seção apresentamos os elementos manipulados pelo editor de estruturas. A apresentação é feita de forma genérica, uma vez que o editor de estruturas é um editor genérico (meta-editor). A correspondência entre os elementos de estrutura com elementos específicos das linguagens de representação, é estabelecida nos capítulos que tratam destas linguagens. Por exemplo, a vinculação dos elementos genéricos do editor de estruturas aos elementos específicos de um projeto modular de programas é descrito no capítulo **Linguagem Estrutura Modular**. Cabe salientar que a terminologia utilizada corresponde à terminologia de projetos estruturado de programas, uma vez que esta é a forma mais comum do uso do editor de estruturas. No entanto, os termos utilizados aqui são termos genéricos e sua especificidade depende da linguagem de representação sendo editada. Por exemplo, na linguagem de representação **Documentação**, o elemento genérico *Módulo* corresponde ao elemento específico *Documento*.

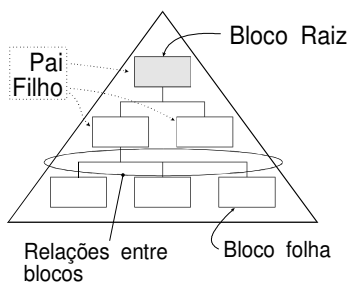


Figura 7.2. Ilustração de estrutura

Estrutura	é uma hierarquia definida pelas relações <i>composição</i> (relação Comp) e <i>decomposição</i> (relação Decomp), envolvendo elementos de uma mesma classe. A figura 7.2 ilustra uma estrutura e a nomenclatura correspondente
Bloco	é uma unidade da estrutura e que corresponde a um único objeto da classe sendo editada.
Bloco raiz	é um bloco origem de uma estrutura. Usualmente, embora não necessariamente, uma estrutura é uma árvore, isto é, uma estrutura hierárquica em que cada nó tem no máximo um pai. O bloco raiz é a raiz desta árvore.
Pacote	é um elemento que define a <i>interface</i> de uma estrutura com outros objetos. Por exemplo, em estruturas de código, um pacote corresponde a um procedimento ou sub-rotina. Em estruturas de dados um pacote corresponde a um tipo de dados. Em estruturas de par grafos, um pacote corresponde a um capítulo. A figura 7.3 ilustra uma estrutura de pacotes.

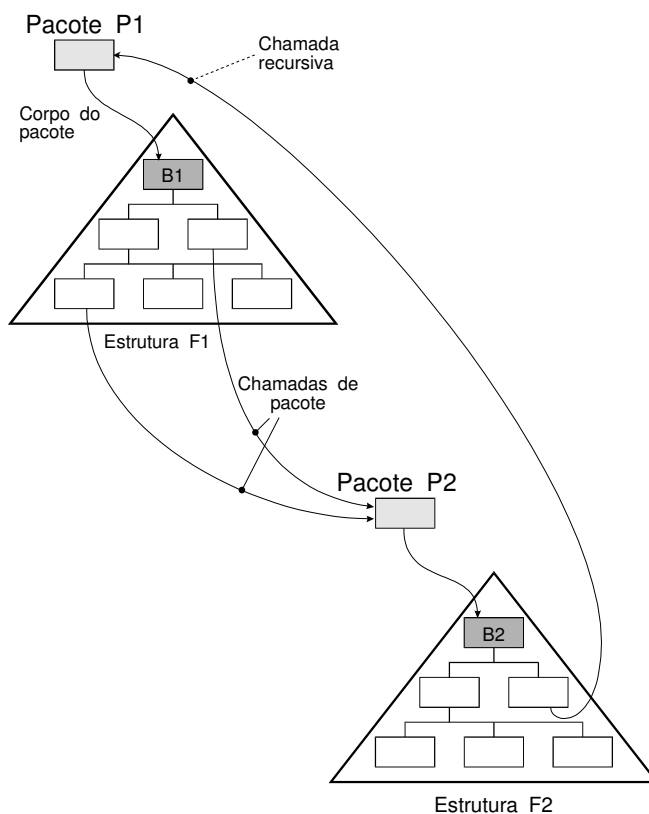


Figura 7.3. Estrutura de pacotes

Corpo de pacote	é uma estrutura vinculada a um pacote. O corpo do pacote define uma implementação do pacote.
Composição de pacote	é uma estrutura formada por objetos da classe correspondente ao pacotes do editor de estruturas. Esta estrutura tem por finalidade descrever a interdependência existente tais objetos.

Bloco chama	é um bloco que referencia um pacote. Por exemplo: em estruturas de código de programas, o bloco chama corresponde a uma chamada de sub-rotina, onde esta sub-rotina é definida pela estrutura do pacote chamado. Em estruturas de dados, um bloco chama corresponde a uma referência de uso do tipo de dados chamado. Em uma estrutura de atividades (linguagem Gerência de Projetos), uma chamada relaciona uma atividade a um projeto. Este relacionamento corresponde à necessidade da conclusão do projeto chamado, para que a atividade possa ser realizada.
Módulo	é uma coletânea de pacotes e estruturas, formando uma unidade bem delimitada e independente. Por exemplo, um composto formado por diversos pacotes programa, tipos de dados e declarações de dados, corresponde a um módulo de programa.
Composição de módulo	é uma estrutura formada estritamente por objetos da classe correspondente aos elementos módulo da estrutura. Esta estrutura tem por finalidade descrever a interdependência existente entre tais objetos.

O significado específico de cada um dos conceitos acima descritos depende da linguagem de representação utilizada. A terminologia está calcada nos conceitos de projeto estruturado de programas, por ser esta a forma mais comum de estrutura. A título de ilustração, apresentamos, a seguir algumas correspondências entre os componentes de uma estrutura e os componentes de algumas linguagens de representação.

Linguagem Estrutura Modular	Nesta linguagem um módulo corresponde a um programa compilável independentemente. Um pacote corresponde ou a uma função (“procedure”, “subroutine”, “function”, etc.), ou a um tipo de dados (“struct”, “typedef”, “record”, etc.). Finalmente, um bloco corresponde ou a um bloco programa (fragmento de código) ou a uma estrutura de dados (declaração elementar).
Linguagem Documentação	Nesta linguagem um módulo corresponde a um documento. Um pacote corresponde a um capítulo e um bloco corresponde a um parágrafo.
Linguagem Gerência de Projeto	Nesta linguagem um pacote corresponde a um projeto e um bloco corresponde a uma atividade. Nesta linguagem não existe classe de objetos que corresponda a módulos.

Nos capítulos que tratam das linguagens de representação, serão detalhadas as correspondências das diversas classes de objeto com os elementos de estrutura do editor de estruturas.

7.2 Ativação do editor

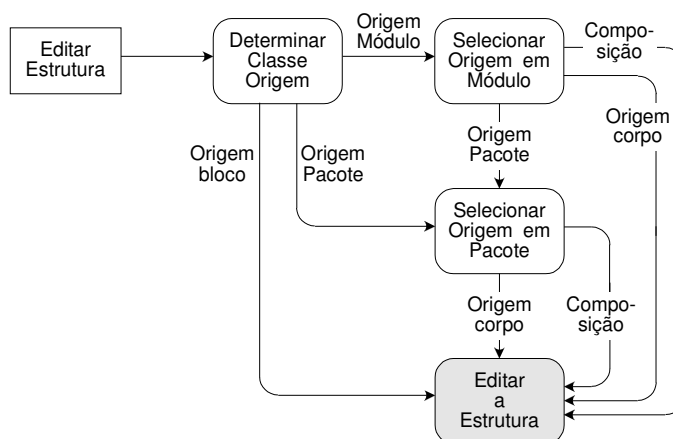


Figura 7.4. Procedimento para seleção da estrutura a editar

O editor de estruturas é ativado pelo **Editor de Dicionário**. A figura 7.4 ilustra o procedimento efetuado do momento em que é ativado o editor de estruturas, até o momento em que a estrutura se torna disponível para edição. A descrição do procedimento baseia-se na linguagem de representação **Estrutura modular**. Nas demais linguagens de representação o procedimento segue linhas análogas. Note que é possível iniciar a edição de estrutura em qualquer classe de objeto. Mais especificamente, é possível iniciar a edição de estruturas nas classes correspondentes a módulos, a pacotes, ou a blocos.

Se a classe de elemento corrente corresponde a um *módulo*, poderão ser escolhidos:

Pacotes	seleciona e edita um dos pacotes de programa conhecidos pelo módulo.
Corpo	seleciona e edita o corpo do módulo. Em programas Pascal, o corpo corresponde ao código do programa principal. Já em programas C, o corpo de módulos não possui significado.
Tipo	seleciona e edita um dos tipos de dado conhecidos pelo módulo. Em programas, estes são pacotes interpretados como estruturas de <i>dados</i> .
Dados internos	seleciona e edita as declarações de dados <i>encapsulados</i> no módulo
Interface	seleciona e edita as declarações que fazem parte da interface do módulo.
Composição	edita uma estrutura de módulos. A estrutura de composição permite definir estruturas de interdependência dos módulos.

Se a classe de elemento corrente corresponde a um pacote (pacote de programa ou tipo de dados) na linguagem de representação corrente, poderão ser escolhidos:

Corpo	edita a estrutura associada ao pacote. Se o pacote for um <i>pacote programa</i> os blocos serão interpretados como <i>programa</i> . Se o pacote for um <i>tipo</i> , os blocos serão interpretados como <i>dados</i> .
Composição	edita a estrutura de pacotes. A estrutura de composição permite definir estruturas de interdependência dos pacotes.

Se a classe de elemento NÃO corresponder nem a um módulo nem a um pacote na linguagem de representação corrente, é ativado um editor simples de estruturas de composição e decomposição.

7.3 Foco de atenção

O foco de atenção do editor de estruturas é definido por:

Posição da janela sobre a estrutura	é sempre posicionada e diagramada automaticamente.
Bloco corrente	é o bloco exibido na janela que está com a moldura realçada. As operações do editor de estrutura são sempre relativas ao bloco corrente.
Tipo do bloco sendo editado	identifica a classe de objetos na qual iniciou a edição. Note que esta classe pode ser diferente da classe dos blocos que formam a estrutura sendo editada. Por exemplo, ao editar blocos programa, tendo iniciado em um objeto da classe módulo, o tipo do bloco apresentado será módulo. É exibido à esquerda na barra de “feedback” do vídeo.
Interpretação	identifica como está sendo interpretada a estrutura sendo editada (<i>programa, dados, documento, composição, etc.</i>). É exibido na barra de “feedback” à direita do tipo do bloco.

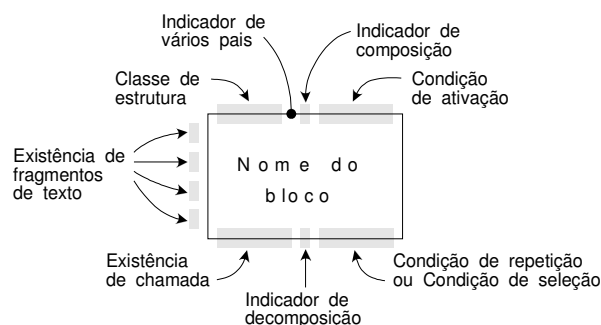


Figura 7.5. Informações nos blocos

Os blocos exibidos na tela apresentam uma série de informações complementares. A figura 7.5 ilustra os campos de informação de um bloco programa. Exceto as informações estruturais, todas as demais dependem da linguagem de representação utilizada. O indicador de composição, informa se existe ou não pai para o bloco. Quando a linha superior do bloco for desenhada com linha dupla (indicador de vários pais), significa que o bloco possui mais de um pai. O indicador de decomposição informa se existem ou não filhos para o bloco em questão. O editor desenha uma subconjunto da estrutura. A escolha deste subconjunto leva em conta os blocos mais recente-

mente referenciados. Conseqüentemente, blocos que não tenham sido referenciados, indicam somente que possuem filhos.

7.4 Navegação na estrutura

- [setas] torna corrente o bloco adjacente na direção da seta. Caso se use a seta [SOBE] e o bloco possui mais do que um pai: é ativado um menu de seleção para determinar qual pai deve ser escolhido como destino. Utilizando os botões neste menu pode-se selecionar (caminhar), adicionar, excluir, bem como examinar ou editar especificações dos blocos pai. Um bloco com vários pais é sinalizado por uma linha dupla na moldura superior do bloco. Estabelecendo vários pais para um mesmo bloco, pode-se criar estruturas não arborescentes quaisquer (grafos). No entanto, na tela serão exibidos somente estruturas arborescentes.
- [END setas] torna corrente o bloco no final da estrutura na direção dada.
- [S] seleciona e torna corrente um bloco consistente com o tipo de estrutura sendo editado. Por exemplo, ao editar código executável (opções de componentes de módulo **Pacote** e **Corpo**), pode-se navegar para blocos programa, pacotes ou corpos de módulos. Já quando se está editando declarações de dados (opções **Tipos**, **Interface** e **Dados internos**), pode-se navegar para blocos de dados, tipos e módulos. Finalmente, ao editar uma estrutura de composição de módulos (opção **Composição**), pode-se navegar somente para outros módulos.

7.5 Criação e exclusão de blocos

- [INS seta] insere um bloco novo na direção da seta. O nome deste bloco será editado com o formulário associado à função **Criar/Editar Nome** da classe correspondente.
- [DEL B] exclui somente o bloco corrente. Os filhos do bloco corrente serão movidos para o local onde se encontrava o bloco corrente.
- [DEL E] exclui o bloco corrente e toda a subestrutura que dele emana.

Em ambos os casos de exclusão a confirmação da exclusão oferece as alternativas:

- Excluir dependentes* o objeto corrente será excluído do dicionário e de todas as relações em que figure. Objetos dependentes relacionados ao objeto corrente serão excluídos também caso não se relacionem a outros objetos.
- Manter* o objeto corrente será excluído da estrutura sendo editada, mas permanece no dicionário (base de software). Para facilitar a localização dos objetos mantidos, eles serão tornados filhos de um objeto da mesma classe com o nome **!Limbo**. Caso este objeto não exista, ele será criado automaticamente. A vinculação de objetos como decomposições do objeto **!Limbo** permite a realização de operações de edição e movimentação complexas.
- Não excluir* cancela o comando.

7.6 Cópia e movimentação de blocos

- [C] marca o bloco corrente como *estrutura a copiar* e entra no estado **COPIAR EST**. Neste estado pode-se executar todos os comandos de navegação na estrutura. O bloco destino é identificado pelo comando [ENTER]. Neste momento é ativado um menu para a seleção de onde será realizada a inserção da cópia, que poderá ser à direita, à esquerda ou abaixo do bloco marcado. Selecionada a posição da inserção, a estrutura a copiar é copiada, sendo que os nomes dos blocos serão precedidos do caractere '#', evitando, desta maneira, a duplicidade de nomes no dicionário. Todos os fragmentos de texto, aliases e relações copiáveis serão copiados conforme a linguagem de representação corrente. Deve-se tomar o cuidado de não copiar para dentro da estrutura sendo copiada. Caso isto seja feito serão criadas repetidas cópias, até esgotar a estrutura interna de controle da cópia.
- [M] marca o bloco corrente como *estrutura a mover* e entra no estado **MOVER EST**. Neste estado pode-se executar todos os comandos de navegação na estrutura. O bloco destino é identificado pelo comando [ENTER]. Neste momento é ativado um menu para a seleção de onde será realizada a inserção da estrutura a mover, que poderá ser à direita, à esquerda ou abaixo. Seleciona-

da a posição da inserção, a estrutura a mover é retirada da posição onde se encontrava e inserida na nova posição. Movimentações de subestruturas podem ser conseguidas também via **Editor de Formulário** ao editar diretamente as relações composição/decomposição (pai/filho) nos respectivos formulários.

É permitido copiar ou mover de qualquer objeto para qualquer outro objeto da mesma classe, independentemente do fato de pertencerem ao mesmo diagrama ou não.

7.7 Estabelecer chamadas de pacotes

Blocos de estrutura contidos em corpos de pacotes ou de módulos, podem *chamar* pacotes. Caso a estrutura represente um programa a chamada corresponde a um “call” de sub-rotina. Como o editor de estrutura é genérico, um pacote pode representar, entre outros, um tipo de dados. Neste caso a *chamada* corresponde a uma referência a um tipo utilizada em uma declaração de dados.

7.7.1 Estabelecer: *bloco chama pacote*

[PGDN] navega, ou estabelece uma chamada do bloco corrente para um pacote. É ativado um menu das chamadas do bloco corrente a pacotes.

O menu possui os seguintes botões:

[ENTER] caminha para o bloco raiz do corpo do pacote chamado selecionado.

[INS] insere um pacote na lista de pacotes chamados pelo bloco corrente, mesmo que esta esteja vazia, e torna corrente a raiz do corpo deste pacote. A inserção é realizada pelo editor de formulários.

[F4] edita o formulário *default* **Editar** do pacote chamado.

[F5] edita o formulário *default* **Criar/Editar Nome** do pacote chamado.

[DEL] exclui o pacote da lista de pacotes chamados.

7.7.2 Estabelecer: *Pacote é chamado por bloco*

[PGUP] *em bloco raiz de corpo de pacote:* ativa o menu dos blocos que chamam o pacote correspondente ao bloco raiz corrente. *Nos demais casos:* indica erro de operação. Cabe salientar que ao estabelecer uma relação *Pacote é chamado por bloco* está-se automaticamente criando também a relação *Bloco chama pacote*.

O menu possui os seguintes botões:

[ENTER] caminha para o bloco selecionado que chama o pacote corrente.

[INS] insere um bloco que chama o pacote corrente e caminha para ele.

[F4] edita o formulário **Especificar** da classe de objeto que chama o pacote corrente.

[F5] edita o formulário **Criar/Editar Nome** da classe de objeto que chama o pacote corrente.

[DEL] exclui o bloco da lista dos blocos que chamam o pacote corrente.

7.8 Empacotar, desempacotar

[E] *não existindo chamada de pacote no bloco corrente:* cria um bloco pacote e vincula-o ao bloco corrente (*empacota a subestrutura*). No local em que se encontrava o bloco corrente, será criado um bloco que chama o pacote criado. Para criar este bloco é ativado o editor de formulário. Sugere-se que antes de empacotar, o nome do bloco raiz seja ajustado para identificar o pacote a ser criado. O bloco chamada que será criado ao empacotar deverá ter um nome que evidencie o uso que se faz do pacote sendo gerado. *Caso o bloco corrente já chame pacote:* comporta-se como o comando **[PGDN] navegar para pacote chamado** descrito acima.

[R] *caso o bloco corrente seja um bloco raiz de um pacote e exista exatamente uma chamada para este pacote:* elimina o vínculo do bloco raiz com o pacote, tornando o bloco raiz em um bloco filho do bloco que chamava o pacote (*desempacota*). *Caso o bloco corrente seja um bloco*

chamada: seleciona qual das chamadas deve ser recomposta e, após, opera de modo semelhante ao descrito acima.

Observação Um pacote somente poderá ser desempacotado se ele for chamado a partir de um único bloco. Um pacote com mais de uma chamada é sinalizado por uma linha dupla na moldura superior do bloco raiz do corpo do pacote.

7.9 Navegação para outros editores

- [F4] edita o formulário de especificação do bloco corrente.
- [L] edita o formulário de linearização gerado a partir do bloco corrente.
- [D] ativa o **Editor de Dicionários**. No editor de dicionários pode-se ativar, recursivamente, o editor de estrutura ou outro editor qualquer. Ao retornar deste editor, volta-se a editar a estrutura corrente.
- [F6] ativa o editor de estruturas de modo a editar a estrutura contendo o bloco corrente.
- [^F6] ativa o editor de diagramas de modo a editar um dos diagramas em que o bloco corrente figure como instância.

7.10 Terminar a edição de estruturas

- [F3] termina o editor de estrutura. Como o editor de estrutura pode ser ativado recursivamente, pode ocorrer que se retorne a uma estrutura que estava em edição antes.
- [^F3] termina todos os editores ativos, retornando ao **Menu Principal**. O retorno ao menu principal se dá independentemente do nível de recursão em que o sistema se encontre.

7.11 Outros comandos

- [D] ativa o **Editor de Dicionários**. No editor de dicionários pode-se ativar recursivamente qualquer outro editor, inclusive o editor de estrutura.
- [L] edita o texto linearizado a partir do bloco corrente.
- [I] imprime uma seleção formada por formulários e/ou diagramas de estrutura
- [O] ativa o menu **Serviços**.
- [*] redesenha a janela corrente.

Página propositalmente deixada em branco

8. EDITOR DE DIAGRAMAS

O editor de diagramas é utilizado para editar representações gráficas nas diferentes linguagens suportadas. O editor é genérico, independentemente da linguagem específica a ser editada. Com o editor de diagramas pode-se editar, por exemplo, diagramas de fluxo de dados, diagramas de entidades e relacionamentos, máquinas de estado entre outros.

8.1 Conceitos básicos

- Folha (folha eletrônica)** é uma superfície imaginária sobre a qual é desenhado um diagrama. As informações necessárias para se desenhar uma folha eletrônica estão contidas na base de software. Parte da folha eletrônica é visualizada na *janela de edição*. A folha eletrônica poderá ser impressa, gerando uma folha de papel contendo o diagrama. Folhas eletrônicas em Talisman possuem dimensões virtualmente ilimitadas (cerca de 80 por 80 metros).
- Objeto** é um elemento representativo de alguma entidade permanente. São exemplos: *processo*, *depósito de dados*, *ponto de confluência*, *estrutura de dados*, *comentários* etc. Alguns objetos podem aparecer em diagramas, por exemplo *processos* e *relacionamentos*. Outros objetos não figuram em diagramas. Por exemplo, *estrutura de dados* não aparecem em diagramas de fluxo de dados, no entanto, podem ser referenciadas pelos fluxos e pelos depósitos de dados.
- Ligação** é um relacionamento que determina a existência de uma interface entre duas instâncias de objeto.
- Ponto de passagem** é o ponto inicial ou final de uma ligação, ou um ponto de quebra da ligação.
- Aresta de ligação** é um segmento de reta contido entre dois pontos de passagem.
- Rótulo** é um elemento representativo de alguma entidade associada a uma ligação. O rótulo define a natureza da interface estabelecida pela ligação a que estiver associado. São exemplos: *fluxos de dados* definem o que flui; *cardinalidades* definem o número de entidades vinculáveis através de uma relação.
- Elemento** é um objeto ou um rótulo. Cada elemento estará registrado no dicionário correspondente à classe do elemento.
- Instância de elemento** é a ocorrência de uma referência a um elemento em um diagrama. A instância de um elemento está vinculada à especificação do elemento contida no respectivo dicionário. Em um mesmo diagrama podem existir diversas instâncias de elemento. Por exemplo, pode-se representar um mesmo depósito de dados várias vezes em um mesmo diagrama. Instâncias de elemento podem aparecer, ainda, em diversos diagramas. Ao editar um diagrama, as instâncias dos elementos contidas nos diagramas podem ser utilizadas para se ter acesso às especificações destes elementos.
- Imagem de ligação** é um vínculo definido em uma folha, a *folha filho*, referenciando uma das ligações que atinge uma *instância pai* desta folha. Na atual versão, imagens de ligação somente ocorrem em diagramas de fluxos de dados. Nestes diagramas, um processo pode ser explodido em um diagrama mais detalhado. Neste caso a instância do processo é a instância pai e o diagrama é a folha filho. Por definição da linguagem fluxo de dados, cada fluxo (ligação mais rótulo) que atinge um processo, deve ocorrer também na folha filho deste processo. Estes fluxos são as imagens de ligação.
- Instância de imagem** é a ocorrência de uma referência a uma imagem de ligação. Da mesma forma como os objetos estão contidos em dicionários e são referenciados em diagramas através de instâncias de objeto, imagens de ligação ocorrem em dicionários especiais associados às folhas e são referenciadas na folha através de uma instância de imagem. Instâncias de imagem são formadas sempre por um *objeto imagem*, uma *ligação* e um *rótulo imagem*. Como já foi dito, uma imagem de ligação corresponde a uma ligação que atinge a instância pai da folha em que está definida a imagem de ligação. O objeto no outro extremo desta ligação bem como o rótulo da ligação serão copiados para a folha filho sob a forma de um objeto imagem.

8.2 Instâncias de elementos

Para evitar trabalho repetido e o surgimento de inconsistências devido a duplicidades, Talisman mantém uma única cópia para cada elemento do sistema. Os diversos diagramas e formulários estabelecem inter-relacionamentos entre estes elementos. Através de poderosos mecanismos de exploração, pode-se examinar visualmente estas inter-relações, ou, então, produzir regras de validação para elas.

Ao inserir um elemento em um diagrama, está-se criando uma *instância* do elemento neste diagrama. Esta instância é, na realidade, uma referência (relacionamento) à especificação do elemento correspondente. Isto possibilita a utilização de um mesmo elemento em diversos lugares num mesmo diagrama, em diagramas diferentes, ou, mesmo, em linguagens de representação diferentes. Possibilita, ainda, ter acesso direto à especificação correspondente à instância em que se esteja em um dado momento.

A instanciação cria *referências cruzadas*. Em outras palavras, é a instanciação que permite determinar todos os lugares onde um dado elemento é utilizado. Esta informação é vital para o controle da qualidade e para a manutenção do sistema.

Para assegurar a unicidade de especificações, é fundamental instanciar elementos existentes, ao invés de criar elementos novos. Desta forma evita-se a criação de um sem número de elementos diferentes com nomes parecidos que, na realidade, correspondem a um mesmo elemento do sistema sendo especificado. Esta multiplicidade de definições, além de confundir, torna inviável qualquer tentativa de controle de qualidade, mesmo quando realizada sem o apoio de Talisman.

As ligações que atingem uma determinada instância de um objeto definem uma parte das relações deste objeto com outros objetos. O conjunto formado por todas as ligações que atingem todas as instâncias do objeto definem a relação completa deste objeto com os objetos ligados.

8.3 Estruturas de diagramas

Através da decomposição de instâncias de objetos pode-se criar *estruturas de decomposição*. Por exemplo, em diagramas de fluxo de dados, podem-se decompor, sucessivamente, os processos que figuram nestes diagramas. Já, em diagramas de estados e transições pode-se decompor transições. Finalmente, em diagramas de entidades e relacionamentos, podem-se decompor as entidades.

As estruturas de decomposição formam estruturas arborescentes. Nestas estruturas cada diagrama fruto de uma decomposição, especifica propriedades da instância de objeto decomposta em um nível de abstração menor. Finalmente, estruturas de decomposição formam um todo interdependente.

Em diversas ocasiões pode-se desejar criar estruturas de decomposição independentes uma da outra. Para isto deve-se definir, no diagrama de mais alto nível, tantas instâncias de objeto quantas serão as estruturas de decomposição. Um diagrama de mais alto nível é, por exemplo, o diagrama de contexto em fluxos de dados. Em outras palavras, é o primeiro diagrama criado ao editar uma determinada linguagem de representação. As instâncias de objeto criadas no diagrama de mais alto nível, são as raízes de cada uma das estruturas de decomposição. A decomposição destas instâncias resulta nas diferentes estruturas de decomposição.

CUIDADO

na atual versão, somente podem ser decompostas instâncias únicas de objetos. Ou seja, caso um determinado objeto possua duas ou mais instâncias, ele NÃO poderá ser decomposto (explodido). Em adição, se um objeto possui uma instância decomposta, NÃO se pode criar outra instância para este objeto.

8.4 Foco de atenção

O foco de atenção do editor de diagramas é definido por:

Posição da janela sobre a folha posicionada automaticamente, ou por controle do usuário ([S] seleção direta, [J setas] movimentação de janela)

Escala define a dimensão dos elementos exibidos.

Posição do cursor de edição sobre a folha visualizado através de um retângulo cheio.

Elemento corrente é a instância de elemento em cujo domínio se encontra o cursor. Somente existe elemento corrente caso o cursor se encontre no domínio de alguma instância. Se o elemento corrente for um objeto, a sua moldura estará realçada. Se o elemento corrente for um rótulo, a ligação correspondente estará realçada.

Ligação corrente quando estiver definida, é a ligação realçada.

Folha sendo editada identificada na barra de “feedback”.

Coordenadas da posição do cursor sobre a folha identificadas na barra de “feedback”.

8.5 Posicionar o foco de atenção

8.5.1 Posicionar o cursor e a janela

- [setas] movimentam o cursor gráfico.
- [↑setas] reposicionam o cursor segundo os parâmetros *salta linha* ou *salta coluna*, ver comando [F2] parâmetros do editor de diagramas. *Observação:* estes comandos correspondem, na realidade, aos dígitos vinculados às teclas de seta.
- [J setas] reposicionam a janela segundo os parâmetros *janela linha* ou *janela coluna*, ver comando [F2] parâmetros do editor de diagramas.
- [END setas] posicionam o cursor na coordenada extrema (*borda*) do diagrama criado até o momento.
- [F2] é ativado o menu de parâmetros do editor. Neste menu pode-se, entre outros, editar a posição do cursor e/ou da janela.

Se o cursor de edição for movimentado para fora da janela, esta será reposicionada segundo parâmetros *rola linha* e *rola coluna*, ver comando [F2].

8.5.2 Posicionar por seleção

- [E] *caso a instância de objeto corrente possua folha filho:* passa a editar esta folha. *Caso a instância de objeto possa ser decomposta e ainda não possua folha filho:* cria uma nova folha e passa a editá-la.
- [R] retorna à folha que contém a instância pai. Não será desfeito qualquer vínculo ou relação.
- [S] É ativado um menu de seleção da classe de elemento (rótulo, objeto ou folha). Utilizando a opção **folha**, seleciona uma entre todas as folhas definidas na linguagem de representação corrente. As demais opções selecionam a instância de elemento a ser tornada corrente. É ativado um menu de seleção da classe de elemento (rótulo, objeto ou folha). Selecionada a classe, é ativado outro menu de seleção contendo as instâncias desta classe definidas na folha corrente. Selecionada a instância, o cursor será posicionado nesta instância.
- [L T] *estando sobre uma aresta de uma ligação:* posiciona no rótulo associado a esta ligação.

Para selecionar uma ligação, utilize uma das seguintes formas:

- Posicione o cursor sobre uma aresta. Quando o cursor se encontra sobre uma das arestas de uma ligação, esta será a ligação corrente para todos os comandos que requerem uma ligação corrente.
- Posicione o cursor no domínio de um rótulo, tornando o rótulo o elemento corrente. Quando o cursor estiver no domínio de um rótulo, a ligação correspondente é a ligação corrente e estará realçada.

8.6 Inserir elementos no diagrama

8.6.1 Inserir objetos

Para inserir um objeto, efetue o procedimento:

- posicione o cursor onde dever ficar o canto esquerdo superior do objeto a ser instanciado.
- inicie o processo de inserção de instância, comando [INS].
- *se for instância nova:* edite o nome do objeto.
- *se for instância de objeto existente:* selecione o objeto a ser instanciado.

- [INS] ativa o menu de seleção da classe do objeto a inserir e do modo da inserção. Os modos são:
Objeto novo: ativa um formulário para a edição do nome deste novo objeto, ver abaixo.
Instanciar objeto conhecido: ativa o menu de seleção dos elementos contidos no dicionário correspondente. Após a seleção insere no diagrama uma nova instância do objeto selecionado.

8.6.1.1 Regras para nomes de objetos novos

O nome, a sigla e o identificador do objeto são editados utilizando o formulário associado à ação do usuário Criar/Editar Nome, ver **Serviços e Programação de Formulários**. O resultado desta edição pode ser um objeto novo criado ou um objeto existente instanciado:

Será criado e instanciado um novo objeto: se o nome for diferente de todos os nomes contidos no respectivo dicionário.

Será instanciado um objeto conhecido: se o nome for igual a um dos nomes contidos no respectivo dicionário. Pode-se selecionar um nome do dicionário de objetos da respectiva classe por intermédio do comando **[!]** selecionar de dicionário. Evite digitar o nome de um objeto existente, procure sempre selecioná-lo usando o comando **[!]**. Caso o nome seja digitado errado será criado um novo objeto no dicionário corrente, não sendo possível converter este novo objeto em um objeto já existente.

8.6.1.2 Critérios para igualdade de nome

Nomes são comparados de acordo com as regras a seguir:

- Letras maiúsculas e minúsculas, acentuadas ou não, são tratadas como letras iguais.
- Espaços antes e após o nome são descartados.
- Mais de um espaço entre componentes de um nome são tratados como um único espaço.

8.6.1.3 Comandos especiais para nomes

Ao editar o formulário **Criar/Editar Nome** valem, entre outros, os comandos:

- [!]** *em estado NOME:* seleciona um entre os nomes contidos no respectivo dicionário. O nome selecionando será copiado para o formulário. Este nome poderá ser editado. Ao terminar a edição do formulário será decidido se o nome é efetivamente novo (igual a um nome já existente) ou não, tal como descrito acima.
- [F3]** termina a edição do formulário de edição do nome e atualiza o diagrama.
- [ESC]** cancela a edição do formulário, cancelando também o comando de inserção.

Observação pode-se criar uma instância de objeto existente ou através da seleção de uma ação *criar instância*, ou através de uma ação *criar novo* utilizando um nome já definido.

8.6.2 Inserir instância de imagem de ligação

As imagens de ligação são vínculos da folha filho com a instância de objeto pai. A referência a uma imagem se dá através de uma instância de imagem, de modo semelhante à instanciação de objetos.

Podem ser definidas diversas instâncias para uma mesma imagem de ligação. Para criar uma instância de imagem, efetue o procedimento:

- antes de introduzir uma instância de imagem, assegure que já esteja definida a instância de objeto à qual será ligada a instância de imagem.
- posicione o cursor onde será inserida a instância do objeto imagem.
- inicie o processo de criação da instância de imagem, comando **[INS]** com a opção *Imagem de ligação*.
- será apresentada a relação de todas as imagens de ligação definidas na folha sendo editada. Selecione a imagem de ligação que deseja instanciar. As imagens são identificadas pelo nome do rótulo da ligação na instância pai.
- após a seleção, o sistema exibe a instância do objeto imagem correspondente e entra no estado **LIGAR** ou **LIGAR PTO**, conforme o modo de posicionamento de ligações corrente (explicado a seguir).

- ligue o objeto imagem à instância identificada no início deste plano.

8.6.3 Inserir ligação entre instâncias de objetos

8.6.3.1 Modo de criação dos pontos de passagem das ligações

A criação dos pontos de passagem de uma ligação poderá ser realizado:

- ou *automaticamente* (modo padrão) a partir da escolha de um formato padronizado. Ao ligar no modo padrão, o “feedback” de estado indicará **LIGAR**.
- ou *pelo usuário* marcando os pontos de passagem ao estabelecer a ligação (modo livre). Ao ligar no modo livre, o “feedback” de estado indicará **LIGAR PTO**.
- ou pelo usuário *através dos comandos* [P INS] inserir ponto, [P M] mover ponto ou [P DEL] excluir ponto.

O modo de criar ligações pode ser estabelecido globalmente para todas as ligações, editando-se o parâmetro *modo de ligação* do editor de diagramas (comando [F2]).

8.6.3.2 Procedimento para criar ligações

Para criar uma ligação, efetue o seguinte procedimento:

- posicione-se na instância origem.
- *no modo ligar padrão (estado **LIGAR**)*: a direção é selecionada no menu de seleção do formato da ligação. O cursor poderá estar em qualquer lugar no domínio da instância origem. Os pontos de passagem inicial, final e internos serão calculados automaticamente.
- *no modo ligar livre (estado **LIGAR PTO**)*: a direção selecionada é da instância de objeto origem para a instância de objeto destino. Posteriormente, a direção pode ser trocada com o comando [L L] *editar ligação*. Ao iniciar, o cursor deve estar no ponto de passagem início da ligação.
- inicie o processo inserir ligação, comando [L INS].
- caminhe para a instância destino e confirme a seleção da instância destino com [ENTER] ou {ESQ}.
- *em modo **LIGAR PTO***: marque os pontos de passagem durante o caminhamento para a instância destino, utilizando [ENTER] ou o botão {ESQ} do “mouse”. Tome o cuidado de marcar os pontos de passagem fora do domínio de instâncias.
- *se uma das instâncias ligadas for uma instância de objeto imagem*: é ativado o menu de seleção das imagens de ligação vinculadas a esta instância de objeto imagem.
- *em modo **LIGAR***: é ativado um menu para a definição das características da ligação, selecione a direção, o modo de posicionar o rótulo (**manual** ou **automático**) e selecione o formato da ligação. Os pontos de passagem serão calculados automaticamente.
- *se o formato da ligação escolhido for arco*: marque a posição do terceiro ponto (estado **MARCAR PTO**).
- é ativado o formulário de edição/criação de nome do rótulo. Edite o nome do rótulo. As regras de edição do rótulo são idênticas às da edição de nome de objeto vistas acima. Caso a ligação deva ficar sem rótulo, cancele a edição com [ESC] ou {CAN}. Neste caso a ligação continua existindo, porém não terá rótulo. Ao terminar a edição com [F3] e se o nome for vazio, a ligação também ficará sem rótulo.
- *se o rótulo deve ser posicionado manualmente*: marque a posição do rótulo (estado **MARCAR ROT**).

[L INS] *estando na instância origem*: entra no estado **LIGAR** ou **LIGAR PTO** conforme o parâmetro de modo de ligação definido (comando [F2]. *Em modo **LIGAR***: será escolhido um dos formatos padronizados para a ligação. *Em modo **LIGAR PTO***: os pontos de passagem (pontos de quebra) devem ser marcados ao se caminhar para o objeto destino. A ligação é criada entre o objeto corrente ao iniciar a criação (instância origem) e o objeto corrente ao terminar a criação (instância destino) com o comando [ENTER] ou {ESQ}.

[ESC] *durante a criação da ligação*: cancela a criação da ligação.

8.6.4 Inserir rótulo em ligação existente

- [L R] *existindo rótulo para a ligação corrente:* edita este rótulo. *Não existindo rótulo para a ligação corrente:* cria um rótulo para a ligação. Caso o rótulo deva ser posicionado manualmente, posicione o rótulo, estado **MARCAR ROT**.

8.6.5 Inserir ponto de passagem em ligação corrente

- [P INS] *estando sobre uma aresta de uma ligação:* cria um ponto de passagem na ligação corrente no lugar onde se encontra o cursor.

A inserção de pontos de passagem, junto com a movimentação [P M] e a exclusão de pontos de passagem [P DEL], permite organizar o traçado de uma ligação de geometria “difícil”.

8.6.6 Criar folha filho para um objeto

- [E] *estando em uma instância corrente:* muda para a folha filho do objeto correspondente à instância. Caso o objeto não possua folha filho, será criada uma nova folha e inicia-se a sua edição.

8.7 Editar

8.7.1 Editar especificações

- [F5] *estando em instância corrente:* edita nome, sigla e identificador do elemento correspondente a esta instância. O nome é editado com o formulário associado à ação **Criar/Editar Nome**.
- [F4] *estando em instância corrente:* edita a especificação do elemento correspondente. Utiliza o formulário associado à ação **Editar**.
- [L ENTER] *estando em instância corrente:* edita o texto linearizado a partir do objeto instanciado. Utiliza o formulário associado à ação **Linearizar**.
- [D] ativa o **Editor de Dicionários**.

8.7.2 Editar ligação

- [L L] *estando em ligação corrente:* ativa um menu que permite a troca da direção, a troca de formato e a seleção de posicionamento manual ou automático para o rótulo.

8.8 Mover

8.8.1 Mover objetos e rótulos

- [M] *estando em instância corrente:* entra no estado **MOVER ELM** e move a instância o objeto ou rótulo corrente. No estado **MOVER ELM** valem somente os comandos de posicionamento e seleção. [ENTER] ou {ESQ} confirmam a nova posição. Após mover um objeto, todas as ligações que atingem este objeto serão reposicionadas. Após mover um rótulo, este estará marcado para posicionamento manual. Para voltar ao estado de posicionamento automático, edite a ligação, comando [L L].

8.8.2 Mover pontos de passagem de ligação

- [P M] *estando em um ponto de passagem:* entra no estado **MOVER PTO** e movimenta o ponto de passagem da ligação. Os incrementos são de caractere. [ENTER] ou {ESQ} confirmam a nova posição do ponto.

8.8.3 Procedimento para reorganizar diagramas

Para reorganizar diagramas, efetue o procedimento:

- crie o diagrama preocupando-se com o conteúdo e não com o aspecto.

- assegure que todos os nomes e siglas escolhidos representem precisamente o significado do elemento. Lembre-se que o nome pode ter até 250 caracteres. Use a sigla para abreviar e formatar nomes.
- efetue a validação do diagrama. Somente prossiga após ter eliminado todos os erros (ver validação da especificação).
- mova primeiro as instâncias de objeto para as posições finais. Utilize a escala 1:2 para facilitar o trabalho. Procure construir um diagrama “apertado”. Desta forma o diagrama impresso ficará pequeno, e, além disso, ver-se-á mais elementos na tela.
- edite o formato das ligações. Em casos mais complicados, crie, mova ou elimine pontos de passagem das ligações.
- por último, mova os rótulos das ligações.

8.8.4 Exclusão

- [DEL] *estando em instância corrente:* exclui esta instância, após confirmação. Caso a instância seja pai de folha filho, a folha filho, e todas as suas dependências serão eliminadas.
- [L DEL] *existindo ligação corrente:* exclui a ligação corrente após confirmação. São eliminadas todas as imagens de ligação dependentes desta ligação.
- [P DEL] *estando em ponto de passagem intermediário:* elimina este ponto de passagem.

8.9 Navegar e/ou terminar a edição de diagramas

- [F3] termina a edição do diagrama corrente.
- [^F3] termina a edição do diagrama corrente e quaisquer outras edições em progresso, retornando ao **Menu Principal**.
- [F6] *estando sobre alguma instância de objeto:* inicia a edição da estrutura contendo o objeto.
- [^F6] *estando sobre alguma instância de objeto:* navega para um dos diagramas em que o objeto em questão é instanciado.
- [D] ativa o **Editor de Dicionários**. Pode-se agora ativar, recursivamente, qualquer editor inclusive o editor de diagramas.

8.10 Comandos diversos

- [<] muda para a escala 1:1. Esta escala é equivalente à utilizada para impressão.
- [>] muda para escala 1:2. Esta escala exibe mais elementos na janela. Todas as operações continuam valendo. O texto continua na escala 1:1, sendo apresentado de forma reduzida.
- [V] valida a instância de elemento corrente. Utiliza o formulário associado à ação **Validar**.
- [I] imprime a folha corrente ou a especificação da instância corrente.
- [O] ativa o menu **Serviços**.
- [*] redesenha a janela. Este comando assegura que o que é visível na janela corresponde exatamente ao que está contido na base de software.
- [!B] recalcula as ligações automáticas de todo o diagrama e, depois, redesenha a janela.

8.11 Parâmetros do editor de diagramas

- [F2] edita os parâmetros de controle do editor de diagramas, descritos a seguir:
- Coordenadas cursor definem a posição do cursor sobre a folha eletrônica.
- Coordenadas janela definem a posição do canto esquerdo superior da janela sobre a folha eletrônica.

Rola linha e rola coluna definem os incrementos de rolagem automática da janela. São utilizados quando o cursor ultrapassa a margem da janela (somente ao utilizar teclado).

Salta linha e salta coluna definem os incrementos a serem utilizados pelos comandos [**↑setas**].

Salta janela linha e salta janela coluna definem os incrementos a serem utilizados pelos comandos [**J setas**].

Modo de ligação determina se os pontos de passagem devem ser calculados automaticamente (modo **LIGAR**), ou se estes serão determinados pelo usuário ao se definir a ligação (modo **LIGAR PTO**).

Balanceamento de ligações determina se os pontos de passagem de uma ligação devem ser calculados imediatamente - *Automático* - ou somente quando for acionado o comando rebalancear ligações [**!B**] - *Manual*. O uso de rebalanceamento manual acelera o redesenho de janelas ao editar diagramas, em particular, ao mover instâncias de objetos.

Tipo de extração determina como devem ser extraídas as arestas que compõem as ligações. Em modo *Parcial* somente são extraídas as arestas que estão ligadas a alguma instância de objeto pelo menos parcialmente exibida na janela. Em modo *Total*, todas as arestas do diagrama são examinadas, sendo exibidas todas as arestas que interceptem a janela, mesmo que não estejam ligadas a instâncias de objeto visíveis. O uso de extração parcial acelera o redesenho de janelas quando o diagrama contiver muitas ligações que não aparecem na janela.

9. MENU SERVIÇOS

O menu **Serviços** oferece:

- diversos *serviços de configuração do ambiente de desenvolvimento*, tais como Instalação de programas de formulários e Configuração de formulários.
- diversos *serviços de configuração da estação de trabalho*, tais como Configuração de impressora e Configuração de diretórios.
- configuração de diversos *modos de operação*, tais como Modo “auto-save” e Modo LOG.
- execução de *operações utilitárias*, tais como Copiar arquivos e Edição de arquivos quaisquer.

A figura 0.1 ilustra o menu de serviços. Nas seções a seguir descreveremos cada um dos itens deste menu.

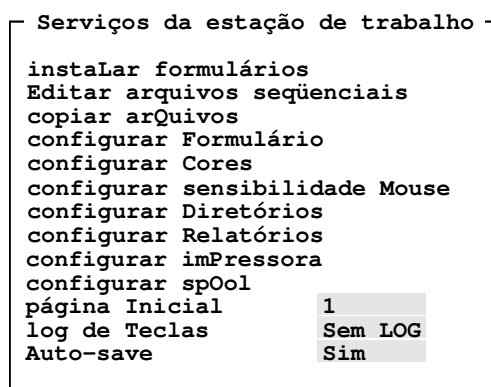


Figura 0.1. Menu de serviços complementares

9.1 Configuração do ambiente de desenvolvimento

9.1.1 Instalação de programas de formulário fornecidos

Talisman apoia diversas linguagens de representação. No **Menu Principal** o item Linguagem de Representação permite examinar e escolher as linguagens que estão disponíveis no ambiente em uso. A linguagem corrente estará sempre identificada no menu principal.

Para poder utilizar uma determinada linguagem de representação é necessário que pelo menos o programa de formulários de *especificação* correspondente a esta linguagem esteja instalado.

O ambiente Talisman é fornecido com programas de formulários já instalados. No entanto, existem diversos programas de formulário alternativos para os que se encontram pré-instalados, além de ser possível desenvolver novos. O item Instalar formulários permite reinstalar ou instalar alternativas dos programas de formulário fornecidos. Por exemplo, o ambiente é fornecido com o linearizador C instalado (ver capítulo **Linguagem Estruturada Modular**, seção Linearizador CLIN). Está disponível, ainda, o linearizador C++ (seção Linearizador CPPLIN). Como os dois programas são mutuamente exclusivos, é necessário instalar especificamente este linearizador, se for desejado gerar programas em C++.

Selecionando o item Instalar formulários, é ativado um menu com todas opções de instalação de programas de formulário fornecidos junto com o ambiente Talisman (arquivos .FRM). Escolhendo a opção desejada é executada a *macro de instalação* correspondente.

Caso, por algum motivo, ocorra um erro durante a execução da macro de instalação, interrompa a execução da macro (qualquer tecla, menos <ESC>) e utilize o item Configurar formulário, descrito a seguir neste capítulo, para efetuar a instalação.

9.1.2 Instalar outros programas de formulário

Para instalar um programa de formulário criado pelo usuário, execute os passos:

- selecione a linguagem de representação no **Menu Principal**.

- ative o serviço **Configurar formulários** e, neste serviço, selecione a classe de programa a ser instalado. O capítulo **Programação de Formulários** descreve cada uma das classes de programa de formulário.
- será solicitado o nome do arquivo contendo o código fonte do programa. Forneça o nome do arquivo, conforme descrito na seção **Selecionar Arquivos** mais adiante neste capítulo.
- O texto fonte do programa estará disponível para edição. Para compilar o programa de formulários, digite [F3]. Caso ocorra algum erro durante a compilação, siga as instruções contidas no capítulo **Programação de Formulários**.

Formulários podem ser instalados também através do serviço `instalar formulários`. Este serviço permite a instalação dos conjuntos de formulários identificados. Siga os seguintes passos:

- selecione a linguagem de representação no **Menu Principal**.
- utilizando o serviço *9.2.2 Configurar diretório*, defina o diretório *default* onde se encontram os programas de formulários.
- utilizando o serviço `instalar formulários`, selecione os formulários a instalar.
- a seleção `instalar geral` poderá ser utilizada somente se a base de conhecimento tiver nenhum programa de formulários instalado.

9.1.3 Edição de programas de formulários

O editor de formulários é utilizado para as seguintes *ações de usuário*:

- Criar objetos.
- Editar a especificação de objetos.
- Imprimir objetos.
- Validar objetos.
- Linearizar objetos.
- Transformar objetos.
- Exportar objetos.

Para cada uma das classes de objetos de cada uma das linguagens de representação utilizadas, deve ser associado um formulário *default* a cada uma destas funções. Isto também é realizado pelo serviço **Configurar formulários**. Detalhes quanto ao processo de busca do formulário origem a executar em função de uma ação do usuário, estão descritos no capítulo **Programação de Formulários**.

Ao ativar o serviço **Configurar formulário**, será ativado um menu de seleção das ações do usuário e dos programas de formulário da linguagem corrente. Ao selecionar uma ação, será ativado outro menu, no qual poderão ser definidos os formulários *default* para a classe de objetos e a ação de usuário selecionada. Ao selecionar uma classe de programa, será solicitado o nome do arquivo que contém o programa fonte. A linguagem de representação corrente pode ser escolhida somente no **Menu Principal**. A edição do arquivo fonte é realizada conforme descrito no capítulo **Editor de Formulários**. A organização e composição do texto fonte de um programa está descrita **Programação de Formulários**.

9.2 Configuração da estação de trabalho

9.2.1 Configurar cores

Para poder dar suporte à variedade de sistemas de vídeo, bem como às variadas características do ambiente Talisman, o sistema de configuração de cores utiliza os seguintes conceitos:

Esquema: cada esquema define uma classe de controladores de vídeo. Para cada esquema é definida uma coletânea completa de parâmetros de controle de cor. Estes parâmetros podem ser editados individualmente. Dependendo do “driver” de vídeo em uso, o conjunto de esquemas selecionáveis poderá variar. Pode-se mudar de esquema a qualquer momento,

sem que seja necessário sair de **Talisman**. Isto permite estabelecer diversos padrões de interface dinamicamente selecionáveis. Existem três esquemas:

<i>Monocromático</i>	utilizado tipicamente por vídeos Hércules. É utilizado, ainda, por vídeos CGA ao editar gráficos. Vídeos monocromáticos admitem somente uma cor e algumas poucas características para os caracteres exibidos (por exemplo: enfatizado, normal, piscante, reverso, etc.). Ao editar diagramas utilizando vídeos CGA e Hércules, as “cores” corresponderão a padrões de vazamento (“dithering”) ao invés de cores ou tonalidades, uma vez que, nestas classes de vídeo, diagramas são monocromáticos.
<i>Tons de cinza</i>	utilizado por vídeos não coloridos que admitam várias gradações de intensidade de frente e fundo. Por exemplo, diversos vídeos admitem gradações de intensidade de verde (vídeos CGA não coloridos em modo texto), outros admitem gradações de cor cinza (vídeos EGA não coloridos).
<i>Colorido</i>	utilizado por vídeos coloridos.
Modo	define uma forma de uso do controlador de vídeo. Em vídeos CGA e Hércules existem diferenças entre a exibição em modo <i>texto</i> e em modo <i>gráfico</i> . Assim, sendo um destes o vídeo instalado, ser solicitado o modo de exibição.
Classes de cores	correspondem aos diversos processadores disponíveis no ambiente Talisman . Cada classe de cores possui o seu conjunto de parâmetros de configuração de cores, nos diversos esquemas e modos.

Ao ativar o item **configurar Cores** é ativado um menu para a escolha do esquema de cores, ou para ativar a configuração, item **conFigurar**.

Selecionando o item **conFigurar** é ativado um menu para a escolha da classe de cores. Após é escolhido o esquema e, se necessário, o modo da classe de cores a configurar. Finalmente, é ativado o configurador de cores. A operação do configurador de cores é óbvia e está descrita na própria janela de configuração. Para terminar a configuração pode-se utilizar um dos seguintes comandos:

[ENTER]	confirma a seleção de cores e termina.
[ESC]	cancela e termina a seleção de cores sem alterar a configuração.
[F6]	restaura a seleção de cores original “de fábrica”.

9.2.2 Configurar diretório

Diretórios são nomes padronizados propostos ao abrir, ou selecionar, arquivos. Cada uma das diferentes categorias de arquivos utilizados por **Talisman** possui o seu conjunto de parâmetros de diretório.

O serviço **Configurar Diretórios** ativa um menu no qual podem ser editados o *nome do diretório* o *nome do arquivo* e o *nome da extensão* a serem propostos sempre que for solicitado um arquivo da respectiva categoria.

O *nome de diretório* pode ser um de:

“path name” completo	o nome proposto será o “path name”. Este nome deve ser fornecido no formato: “drive”:\diretório. O “path name” não deve conter caracteres “*”, “!” ou “?”
!	o nome proposto é o nome do diretório default definido para o tipo de arquivo.
*	o nome proposto é o nome do diretório corrente ao iniciar a execução.
*subdir	o nome proposto é o nome do diretório corrente ao iniciar a execução, sucedido pelo subdiretório subdir .
nulo	(seqüência vazia) o nome proposto será o nome associado à variável de ambiente Talisman , caso esteja definida, ver capítulo Instalação e Ativação , seção Preparação do ambiente de execução . Não estando definida, será proposto o nome do diretório corrente ao iniciar a execução.
*Nome	o nome proposto será o nome associado à variável de ambiente Nome . Caso esta variável de ambiente não esteja definida, será proposto o nome do diretório corrente ao iniciar a execução.
*Nome\subdir	o nome proposto será o nome associado à variável de ambiente Nome , sucedido pelo subdiretório subdir . Caso esta variável de ambiente não esteja definida, será proposto o nome do diretório corrente ao iniciar a execução, sucedido pelo subdiretório subdir .

@ propõe o nome de diretório da base de software corrente. Caso não exista base de software aberta, será proposto o nome do diretório corrente ao iniciar a execução. Este valor só faz sentido com relação a arquivos *spool*, *backup* e/ou *log*.

Ao utilizar variáveis de ambiente, são observadas as seguintes regras adicionais em função do formato do texto associado à variável:

drv:\dir utiliza o diretório *dir* no “drive” *drv*.
\dir utiliza o diretório *dir* no “drive” do diretório corrente.
dir utiliza o diretório corrente sucedido do sub-diretório *dir*.
nulo (variável não definida) utiliza o diretório corrente.

Os nomes de extensão de arquivo utilizados são:

.BSW bases de software.
 .BKP cópias “backup” de bases de software.
 .FRM arquivos seqüenciais contendo programas de formulário.
 .LOG arquivos contendo uma cópia das teclas digitadas.
 .SPL arquivos seqüenciais contendo *spool* de impressão.
 .PRG arquivos seqüenciais contendo código fonte gerado. Esta é a extensão *default*. Deve ser alterada para o tipo de arquivo a ser gerado pelos linearizadores.
 .BAK arquivos “backup”.
 .\$\$\$ arquivos temporários, onde `\$` é qualquer caractere. Ao serem fechados, os arquivos temporários serão automaticamente eliminados.

9.2.2.1 Seleção de arquivo

A seleção de qualquer arquivo segue sempre o mesmo ritual. Inicialmente é executado um diálogo de edição do nome do arquivo. Ao solicitar um nome de arquivo, será proposto um nome de acordo com a definição de diretório existente para a respectiva categoria de arquivo. Edite este nome. O resultado da edição pode conter caracteres “*” e/ou “?” no NomeArquivo e/ou no NomeExtensão, ver abaixo. Cabe frisar que os nomes de arquivos em Talisman devem ser sempre nomes completos, formados por:

Drive : \ Diretório \ NomeArquivo . NomeExtensão

O resultado da edição poderá ser:

nome completo indicando “drive”, diretório, nome do arquivo e extensão. Sendo um nome válido, será aberto este arquivo.
nome incompleto faltam “drive” e/ou diretório. Extrai os dados em falta do diretório corrente ao iniciar a execução de TALISMAN.
nome ambíguo contendo os caracteres ‘’ ou ‘?’* ativa um menu de seleção contendo todos os subdiretórios e todos os arquivos que satisfaçam o nome ambíguo. Pode-se, agora, ou selecionar um destes arquivos, ou um dos subdiretórios. Sendo selecionado o subdiretório “\.”, o último subdiretório contido no “path-name”, caso exista, será eliminado. Sendo selecionado outro subdiretório qualquer, o diretório definido no nome é ajustado para incluir este subdiretório.

Caso o nome editado contenha caracteres “*” ou “?” (nome ambíguo), será ativado um menu de seleção de arquivos. Este menu contém os nomes de todos os arquivos e subdiretórios que satisfazem o nome e que estão contidos no diretório definido. Os nomes de diretórios, caso existam, terminam com o caractere “\.”. Pode-se navegar o sistema de arquivos em busca de um nome de arquivo específico. Para tal, forneça o nome completo do arquivo procurado, sucedendo-o pelo caractere “*.”.

Sendo selecionado um diretório, o valor do *nome de arquivo sendo editado* será ajustado para referenciar este diretório. Repete-se, agora, a seleção do arquivo através do menu de seleção descrito acima. Caso seja selecionado um arquivo este será o arquivo a ser aberto. No entanto, antes de abrir, o arquivo é validado (por exemplo: é

base de software, é arquivo existente para ler, etc.). Esta validação depende de cada caso. Conseqüentemente é descrita nas seções que tratam de manipulação de arquivos.

Este processo (edição de nome, seleção em menu de nomes) continuará até que, ou seja fornecido um nome de arquivo não ambíguo e válido, ou a seleção de arquivo seja cancelada com um comando [ESC].

9.2.3 Configurar saídas impressas

A geração de saídas impressas utiliza os seguintes elementos de configuração:

<i>número da página corrente</i>	redefinível diretamente no item correspondente do menu Serviços .
<i>Relatório</i>	define formato do papel e os títulos do relatório.
<i>Impressora</i>	seleciona uma impressora conhecida ou não. Caso seja escolhida uma impressora não conhecida (<i>Outras</i>) permite a edição dos comandos de controle específicos para a impressora a ser utilizada.
<i>spool</i>	determina se a impressão deve ser feita imediatamente, ou se deve ser gerado um arquivo <i>spool</i> , a ser impresso mais tarde.

9.2.3.1 Configurar impressora

Ativa um menu de seleção da impressora a utilizar. A impressora selecionada passa a ser a impressora *default*.

Sendo selecionada Outra é ativado um menu para a edição dos parâmetros de controle da impressora.

Cuidado somente selecione *Outra*, caso a sua impressora não opere corretamente com qualquer um dos controles padronizados fornecidos.

Para definir os parâmetros de controle da impressora consulte o manual de programação da impressora a ser utilizada. Os parâmetros de controle de impressoras requerem caracteres ASCII e não ASCII. Caracteres ASCII são digitados normalmente. Caracteres não ASCII são digitados de acordo com as regras a seguir:

\ddd onde ddd são os três dígitos decimais do caractere a introduzir.

A seqüência *tradução de caracteres* codifica como devem ser traduzidos os caracteres acentuados, encontrados no texto sendo impresso, para seqüências de controle de impressora que imprimam estes caracteres. Estas seqüências podem ser utilizadas, também, para gerar comandos para formatadores de texto, ou para converter caracteres no padrão PC, para caracteres no padrão ISO. Para cada caractere acentuado possível de ocorrer no texto sendo impresso, deve ser definida uma seqüência de controle. Cada seqüência de controle tem o formato:

la • n c1 c2 ... cn

la é o código do caractere acentuado (consulte o manual da placa de vídeo utilizada).

• é o caractere em branco, tem que existir!

n é o número de caracteres que compõem a seqüência a ser transmitida para a impressora.

c1...cn são os n caracteres que formam a seqüência a ser transmitida para a impressora.

Por exemplo, para gerar a seqüência "<176>" para todos os caracteres "ä" (a trema), a seqüência "<177>" para todos os caracteres "ö" (o trema) e a seqüência ", BKSP c" (vírgula backspace c) para todos os caracteres "ç" (cedilha) forneça:

\132•5<176>\148•5<177>\135•3,\008c

9.2.3.2 Configurar relatório

Relatórios definem um conjunto de parâmetros utilizados para controlar a impressão de diagramas, relatório gráfico, e de formulários, relatório texto.

O parâmetro *traduz acentos* determina se os caracteres acentuados contidos no texto sendo impresso devem ou não ser traduzidos para uma seqüência de caracteres de controle na impressora. Este parâmetro somente tem efeito em relatórios de texto. A tradução se dará em conformidade com o parâmetro *tradução de caracteres* da impressora selecionada, ver seção anterior.

O parâmetro *adição à inicialização* contém uma seqüência de comandos de impressora que será transmitida logo após à inicialização da impressora e antes de qualquer outra coisa. Este parâmetro permite estabelecer propriedades de impressão, tais como: *Impressão comprimida*, *impressão com qualidade carta* etc.

Textos são impressos alinhados na margem esquerda útil e linha útil inicial. Ao atingir a linha útil final, ocorrerá quebra de página. São impressos o título de rodapé e os títulos iniciais, caso estejam definidos. Ao imprimir texto, a *margem direita útil* bem como a *largura do papel* não são levadas em conta. Assegure que todas as linhas a serem impressas são menores do que a largura física do papel. Isto pode ser feito imprimindo em modo comprimido (valor \015 no parâmetro *adição à inicialização*).

Diagramas são centralizados em uma área definida pelos parâmetros: *linha útil inicial*, *margem esquerda útil*, *linha útil final*, e *margem direita útil*. Caso o diagrama seja maior do que o tamanho útil em alguma das dimensões, ele será impresso alinhado na *margem esquerda útil* e/ou na *linha útil inicial*, conforme o caso.

Ao imprimir diagramas deve-se definir a largura do papel, em caracteres, no parâmetro *largura do papel*. Caso um diagrama seja mais largo do que a largura do papel definida, o diagrama será impresso em várias *tripas*. Estas *tripas* poderão ser coladas uma à outra, formando o diagrama completo. O parâmetro *superposição gráfica* indica o número de caracteres a serem superpostos caso um diagrama impresso requeira mais do que uma *tripa*. Isto facilita o corte e a colagem das diferentes *tripas* que formam um único diagrama. Ao imprimir diagramas, o *tamanho da página* (número de linhas na página), não será considerado.

Os títulos de cabeçalho e de rodapé admitem os comandos de formatação específicos. Estes comandos de formatação são incluídos no texto do título:

%#	introduz o número da página em um campo de tamanho suficiente para o número.
%#nn	introduz o número da página alinhado à direita em um campo de tamanho nn.
%cnn	tabula até a coluna nn.
%d	introduz a data no formato dd/mm/aa.
%h	introduz a hora no formato hh:mm.
%b	introduz o nome do arquivo que contém a base de software.
%p	introduz o nome do projeto (nome lógico da base de software). Este nome é editado na Identificação da Base .
%n	<i>ao imprimir um formulário</i> : introduz o nome do elemento cujo formulário está sendo impresso.
%ann	<i>ao imprimir um formulário</i> : introduz o nn-ésimo aliás do elemento sendo impresso.
%t	<i>ao imprimir um formulário</i> : introduz o nome da classe (tipo) do elemento sendo impresso.
%f	<i>ao imprimir diagramas</i> : imprime o nome do diagrama sendo impresso.

9.2.3.3 Configurar *spool*

A impressão pode ser dirigida ou diretamente para a impressora, ou para um arquivo *spool*. Ao transmitir para um arquivo *spool* é gerada uma imagem de impressão. Esta pode ser transmitida para outro computador, gravada em algum disquete, ou finalmente, impressa em uma oportunidade mais conveniente. Para imprimir um arquivo *spool* gerado, utilize o comando DOS:

```
copy arquivo prn /b
```

onde *arquivo* é o nome do arquivo *spool* gerado. Cabe salientar que o arquivo *spool* somente pode ser copiado ou impresso depois que tenha sido fechado.

Impressora	fecha o arquivo <i>spool</i> e redireciona a saída impressa diretamente para a impressora.
Abrir <i>spool</i>	solicita o nome de um arquivo ASCII que conterà a saída impressa gerada, ver serviço Configurar diretórios para definir o diretório <i>default</i> . Enquanto o arquivo <i>spool</i> estiver aberto, toda a saída impressa gerada será direcionada para o este arquivo. Note que o arquivo somente poderá ser transmitido ou impresso depois de fechado.
Adicionar <i>spool</i>	solicita o nome de um arquivo ASCII ao final do qual será adicionada a saída impressa gerada.
Fechar <i>spool</i>	fecha o arquivo <i>spool</i> e redireciona a saída diretamente para a impressora.

9.2.4 Configurar mouse

Permite desligar o controlador do *mouse*. Permite, ainda, ajustar os parâmetros de sensibilidade horizontal e vertical do *mouse*. Estes parâmetros dependem do *mouse* e do modo de trabalho de cada usuário.

9.3 Modos de operação

9.3.1 Modo *auto-save*

Operando com *auto-save* ligado, Talisman salva as alterações à medida que forem sendo realizadas. Cabe salientar que edições em formulários são realizadas em um arquivo temporário, conseqüentemente serão registradas na base de software somente quando o formulário for fechado (comando [F3] ou navegação).

A gravação imediata das alterações assegura que a base de software esteja sempre atualizada, mesmo que ocorra algum acidente, como, por exemplo, falta de energia. A base de software somente poderá ser danificada caso o acidente ocorra *enquanto* estiver sendo gravada. O término da gravação é sinalizado para o usuário através da mensagem **SV** na parte de segundos do relógio.

Em geral, o modo *auto-save* não afeta o desempenho percebido pelo usuário de Talisman. Somente desligue o *auto-save*, caso isto efetivamente interfira com o desempenho observado.

Observação mesmo operando em modo *auto-save*, podem ocorrer acidentes com o sistema de arquivos. Para não perder trabalho, *gere cópias “backup” com alguma frequência.*

9.3.2 Modo LOG

O LOG destina-se à geração padronizada de bases de software. Através da execução de um LOG, adequadamente criado, pode-se compor uma base de software com características bem definidas. Como o LOG é uma macro, tal como descrita no capítulo **Macros e Lições**, não é possível utilizar macros ou lições enquanto se está registrando ou executando um arquivo LOG.

9.3.2.1 Criação de LOG

Para criar um LOG siga os seguintes passos:

- i. *antes de iniciar a execução* de Talisman, crie uma cópia “backup” da base de conhecimento e da base de software para a qual será gerado o LOG.
- ii. inicie a execução de Talisman.
- iii. no menu **Serviço**, posicione-se em **Modo LOG** e selecione a opção **Criar**. Caso se queira adicionar a um LOG relativo a um *backup* da base de software e respectivo LOG já existentes, pode-se selecionar a opção **Adicionar**.
- iv. retorne ao menu principal e abra a base de software. Após abrir a base de software, será solicitado o nome do arquivo LOG a ser gerado.

Uma vez aberto o arquivo LOG para gravação, as teclas e ações de “mouse” serão transcritos automaticamente para este arquivo LOG. O registro do LOG pode ser interrompido antes de fechar a base de software com a opção **Terminar** do comando [!M], ver capítulo **Macros e Lições**.

Quando a base de software for fechada, o arquivo LOG será fechado automaticamente.

Observação o LOG é, na realidade, uma macro. Para que ele possa executar corretamente, é necessário que todas as condições iniciais existentes ao criar o LOG existam também ao iniciar a sua execução. Por esta razão é imprescindível que sejam feitas cópias “backup” tanto da base de conhecimento como da base de software. Ao iniciar a execução do LOG, as duas bases devem ter sido restauradas para estarem em conformidade com o “backup”.

9.3.2.2 Executar o LOG

Para executar um LOG siga os passos:

- *antes de iniciar a execução* de Talisman, restaure a base de conhecimento e a base de software a partir das cópias “backup” criadas antes de iniciar o registro do arquivo LOG.

- inicie a execução de Talisman.
- no menu **Serviço**, posicione-se em Modo LOG.
- selecione a opção **Exec contínuo** se desejar iniciar a execução do LOG em modo contínuo, ou **Exec passo** se desejar iniciar em modo passo a passo. Os modos de execução estão descritos no capítulo **Macros e Lições**.
- retorne ao menu principal e abra a base de software. Após abrir a base de software, será solicitado o nome do arquivo LOG a ser executado.

Uma vez aberto o arquivo LOG para execução, as teclas gravadas serão interpretadas de acordo com o modo de execução solicitado. A operação da execução do LOG é a mesma que a utilizada para lições e macros, descrita no capítulo **Macros e Lições**.

9.4 Operações diversas

9.4.1 Editar arquivos seqüenciais

Talisman permite a edição de diversos arquivos seqüenciais junto com a edição da base de software. Esta facilidade pode ser útil, por exemplo, para corrigir erros reportados por um compilador. Neste caso um arquivo seqüencial conterá as mensagens de erro, e outro conterá o correspondente código fonte. Inspeccionando-se estes arquivos pode-se determinar os elementos da base de software a serem editados de modo que uma posterior *linearização* não mais provoque estes erros.

A edição de arquivos é realizada em uma cópia armazenada no arquivo de trabalho TALISMAN.\$\$1. As edições realizadas somente serão registradas se o arquivo sendo editado for salvo, ver abaixo. Caso este arquivo esteja em “ram-disk”, assegure que este possua tamanho suficiente.

Pode-se iniciar a edição de arquivos em qualquer lugar, através do comando [!Q]. Pode-se, também, iniciar a edição de arquivos através do menu **Serviços**. Selecionando o item **Editar arquivos** é ativado um menu para a seleção do arquivo a editar. Este menu admite os seguintes botões:

- | | |
|-------------|--|
| [INS] | solicita o nome de um arquivo a editar. Caso não exista cria este arquivo. |
| [F4] | retoma a edição do arquivo selecionado. |
| [DEL] | exclui o arquivo da lista de arquivos em edição. Caso o arquivo tenha sido alterado e esta alteração não tenha sido gravada, será confirmada a exclusão. |
| [F5] | troca o nome do arquivo. Depois de trocado o nome, todas as operações de salvamento se darão com relação a este novo nome. |
| [F6] Salvar | salva todas as alterações realizadas e continua a edição. |
| [ESC] | retorna ao ponto em que foi ativada a edição de arquivos. |

A edição é realizada pelo **Editor de Formulários**. Ao editar um arquivo, o comando [F3] *fechar formulário*, somente suspenderá a edição, permitindo que ela seja retomada mais tarde.

9.4.2 Copiar arquivos

Podem ser geradas cópias de quaisquer arquivos. Ao ativar este serviço são solicitados os nomes dos arquivos origem e destino.

10. MACROS E LIÇÕES

10.1 Conceitos

Macros e lições são seqüências de teclas e janelas explanatórias que podem ser interpretadas por solicitação do usuário.

Macros reduzem o trabalho do usuário, permitindo que seqüências de teclas repetitivas sejam associadas a um único comando. Executando-se este comando, toda a seqüência de teclas será fornecida automaticamente.

Macros estão associadas a *editores*, ver capítulo **Apresentação do Ambiente Talisman**. Ao serem registradas, as origens das macros são vinculadas ao editor ativo no início do registro. Para poderem ser executadas, o editor correspondente deve estar ativo. Durante o registro e execução da macro pode-se mudar livremente de editor. A associação das macros a editores reduz a possibilidade de se executar uma macro fora do seu contexto de registro, evitando a produção de resultados imprevisíveis.

Lições oferecem um mecanismo de instrução programada, ver capítulo **Treinamento e Auxílio**. Lições explicam a dinâmica de operações complexas. Lições podem ser criadas e editadas pelo usuário, possibilitando, assim, a criação de mecanismos de treinamento específicos para cada ambiente de desenvolvimento. Na prática lições e macros são a mesma coisa, diferindo-se somente quanto à maneira de serem ativadas.

O sub-sistema de execução e registro de macros é capaz de registrar e executar, ainda, *logs de comandos*. Estes são seqüências de teclas gravadas em arquivos. A execução de um log a partir de uma base de software “backup” permite restaurá-la em caso de acidente. O uso e a operação de logs é explicada no capítulo **Serviços**.

10.2 Ativação

10.2.1 Ativação de macro

- [!X tecla] inicia a execução da macro tecla definida no editor corrente. Macros estão associadas a editores, podendo ser executadas somente se o editor correspondente estiver ativo. Não estando definida a macro, será automaticamente ativado o comando [!M] abaixo.
- [!M] ativa o menu de operação do sistema de macros. Selecionando *Selecionar macro*, é ativado um menu de seleção contendo todas as macros disponíveis no editor corrente. Selecionando uma delas com o botão [ENTER], ela será executada.

10.2.2 Ativação de lição

Lições são ativadas no **Menu Principal**, ver capítulo **Treinamento e Auxílio**. Selecionando o item *selecionar treinamento* é ativado um menu de seleção contendo todas as lições disponíveis na base de auxílio. Selecionando uma delas com o botão [ENTER], ela será executada.

10.3 Execução

A execução de macros inicia no modo **Contínuo**. A execução de lições inicia no modo **Passo**. Os modos de execução, bem como a operação da execução de macros e lições estão descritos mais adiante neste capítulo.

Para assegurar transparência, ao iniciar a execução de uma macro ou lição, as chaves de modo de edição *Inserção*, *Acentuação* e *Endentação* são alteradas, respectivamente, para: **INSERIR**, **ACENTUAR** e **ENDENTAR**. Durante o registro da macro estes modos podem ser alterados livremente.

10.4 Criação

Para criar uma macro ou lição, efetue o seguinte o procedimento:

- i. planeje a tarefa a ser realizada pela macro ou lição, inclusive o que e quando será explicado.
- ii. registre as teclas e as janelas explanatórias da macro ou lição digitando diretamente a seqüência de ações e janelas explanatórias planejadas. Durante o registro as teclas são interpretadas normalmente. Ao execu-

tar mais tarde a macro ou lição, o efeito será exatamente igual ao resultante da execução das teclas sendo registradas.

iii. execute a macro ou lição verificando se está correta.

iv. edite o corpo da macro ou lição para realizar pequenas correções.

Ao criar uma **lição** tome o cuidado de fazê-la ficar *transparente*. Ou seja, assegure que o efeito provocado pela execução da lição seja imperceptível pelo sistema. No caso de ilustrações de edição de base de software, procure sempre editar uma base temporária (ver **Serviços Configurar diretório**). Assegure, também, que as lições operem a partir de uma base de conhecimento padrão. Um bom padrão é a base de conhecimento original acrescida de programas de formulário. Finalmente, adote a convenção de iniciar a lição com uma janela explanatória, explicando o objetivo e os requisitos da lição. Logo após a esta janela, introduza uma tecla de função **Espera**. A criação ou edição e lições altera o conteúdo da base de auxílio.

Ao criar uma **macro** assegure-se que ela seja *insensível* ao contexto. Ou seja, assegure que ela possa ser executada corretamente, independentemente do conteúdo da base de software, bem como da base de conhecimento. Caso a macro seja sensível ao contexto, ao encontrar uma condição de conflito, serão emitidas mensagens de erro. A criação ou edição de macros altera a base de conhecimento. Como foi mencionado acima, os valores das chaves de controle de edição são ajustados para um valor padrão ao iniciar a execução de uma macro ou lição.

CUIDADO se, durante o registro de uma macro ou lição, for exibida alguma mensagem de erro *cancele a mensagem somente com o comando [ESC]*. Qualquer outra tecla, embora cancele a mensagem de erro durante o registro da macro ou lição, durante a execução da macro ou lição provocará o cancelamento da execução.

10.4.1 Início do registro

Para criar uma macro ou lição, siga as instruções acima até se posicionar no menu de seleção da macro ou lição. O botão:

[INS] inicia a criação da macro ou lição. Será solicitada a *tecla identificadora* e logo após será solicitado o nome da macro ou lição. A tecla identificadora é utilizada no comando [!X tecla] para executar a macro. O nome será apresentado nos respectivos menus de seleção. Após terem sido definidos a tecla identificadora e o nome, é iniciado o registro da macro ou lição. Para tal transcrevem-se, automaticamente, as teclas digitadas e/ou as ações de “mouse” para o corpo da macro ou lição. Cabe frisar que as teclas serão interpretadas enquanto estão sendo registradas.

Observação: macros e lições não podem ser aninhadas.

10.4.2 Término do registro

Para terminar o registro de uma macro ou lição, ative o menu de macros, comando [!M], e selecione Terminar.

10.4.3 Operações especiais

Durante a execução de macros ou lições poderão ser executadas *teclas de função especiais*. Estas teclas não correspondem a teclas disponíveis no teclado, servindo somente para controlar a execução da macro ou lição. Elas são inseridas pelo menu de macros, comando [!M]. As operações e correspondentes teclas especiais são:

Inserir espera	inclui uma tecla <Espera> no corpo da macro ou lição, ao encontrar esta tecla durante a execução, esta será suspensa, até que o usuário pressione alguma tecla. No canto direito inferior aparecerá <Espera>.
Muda para contínuo	inclui uma tecla <VaiCnt> no corpo da macro ou lição. Ao encontrar esta tecla durante a execução, o modo de execução passará a ser <i>Contínuo</i> .
Muda para passo	inclui uma tecla <VaiPas> no corpo da macro ou lição. Ao encontrar esta tecla durante a execução, o modo de execução passará a ser <i>Passo</i> .
Empilha modo	inclui uma tecla <Empil> no corpo da macro ou lição. Ao encontrar esta tecla durante a execução, o modo de execução corrente será empilhado.
Desempilha modo	inclui uma tecla <DesEmp> no corpo da macro ou lição. Ao encontrar esta tecla durante a execução, o modo de execução corrente passará a ser o modo contido topo da pilha, a seguir a pilha será reduzida. As teclas <i>Empilha modo</i> e <i>Desempilha modo</i> permitem

gerar macros contendo seqüências de comando operando em variados modos de execução (p.ex. passo misturado com contínuo).

Ler dado inclui uma tecla <LerVal> no corpo da macro ou lição. Ao encontrar esta tecla, a execução da macro é suspensa. Durante a execução desta função o usuário poderá fornecer qualquer seqüência de teclas, comandos e ações. A execução da macro ou lição é retomada quando o usuário teclar [ENTER] ou [ESC]. Estas teclas não são interpretadas, servido, simplesmente, de terminadores da função *Ler dado*. Durante a gravação da macro ou lição, as teclas e ações de “mouse” fornecidas após a inserção de <LerVal> até a primeira ocorrência de [ENTER] ou [ESC] não serão gravadas. Tanto durante a gravação como durante a leitura, o campo *Estado Corrente* do vídeo indicará *Ler valor*.

Iniciar janela introduz uma janela explanatória no corpo da macro ou lição. Uma janela explanatória é formada por:

- um conjunto de parâmetros de definição da janela, tipo da janela, origem, tamanho, tecla <CabJan>.
- dois títulos, cada qual iniciando com a tecla <TitJan>.
- pelo texto a ser exibido na janela terminando com a tecla <FimJan>.

Ao atingir uma tecla <CabJan> durante a execução da macro ou lição, será exibida a janela explanatória formada pelas teclas a seguir, até a tecla <FimJan>. Para registrar a janela explanatória, serão solicitados a origem (canto esquerdo superior), o tamanho e os dois títulos da janela. Após a janela será aberta conforme a especificação dada. Nesta janela edita-se o texto da janela. A edição do texto da janela é realizada pelo **Editor de Formulários**. No interior da janela valem quaisquer caracteres, mesmo acentuados. Estes caracteres não serão interpretados pelo processador de macros. Termina-se a edição da janela explanatória com o comando [F3] *Terminar edição de formulário*, ou com o comando [ESC] *Cancelar edição de formulário*.

Os parâmetros de definição de tipo da janela são:

- | | |
|---|---|
| 0 | janela posicionada no canto esquerdo superior. |
| 1 | janela centralizada horizontalmente e posicionada no topo do vídeo. |
| 2 | janela posicionada no canto direito superior. |
| 3 | janela centralizada verticalmente e posicionada na margem direita. |
| 4 | janela posicionada no canto direito inferior. |
| 5 | janela centralizada horizontalmente e posicionada na base do vídeo. |
| 6 | janela posicionada no canto esquerdo inferior. |
| 7 | janela centralizada verticalmente e posicionada na margem esquerda. |
| 8 | janela centralizada horizontal e verticalmente. |
| 9 | janela posicionada sobre o cursor. Se a origem for (0, 0), o canto esquerdo superior da janela coincidirá com a posição do cursor no momento da abertura da janela. |

Os valores dos parâmetros de origem da janela podem ser:

- | | |
|-----------|---|
| Positivos | a origem será posicionada para a direita ou para baixo da origem relativa. Note que a origem relativa pode ser absoluta (posicionamento nas margens esquerda e superior do vídeo), ou calculada (nos demais casos). |
| Negativos | a origem será posicionada para a esquerda ou para cima da origem relativa. |
| Nulos | a origem será posicionada sobre a origem relativa. |

Os parâmetros de tamanho da janela podem ser:

- | | |
|-----------|--|
| Positivos | será utilizado o valor fornecido. |
| Negativos | será utilizado o tamanho do vídeo, subtraindo-se o valor do parâmetro. |
| Nulos | será utilizado o tamanho do vídeo. |

10.5 Edição do corpo de macro ou lição

Para editar uma macro ou lição, siga as instruções acima até se posicionar no menu de seleção da macro ou lição. Botões:

- [F4] edita o corpo da macro, inclusive a especificação e o texto das janelas explanatórias.
- [F5] edita o nome da macro.

No corpo de uma macro ou lição em edição não poderão aparecer caracteres acentuados (teclas com valor decimal maior ou igual a 128). Digite sempre o caractere acento e, depois o caractere a acentuar, operando em modo não acentuado (comando [!A]), ver capítulo **Editor de formulários**. A acentuação será realizada ao executar a macro pelo interpretador de macros.

As teclas caractere são apresentadas pelo caractere ASCII correspondente. As teclas de função são apresentadas com o seu nome entre parêntesis angulares, por exemplo <INS> ou <!A>. Esta mesma regra vale para todas as teclas especiais descritas acima, inclusive as de controle de janelas explanatórias.

Fim de linha no texto do corpo é meramente um comando de formatação, não cabendo a ele qualquer efeito na execução da macro ou lição. Os comandos de formatação são incorporados ao texto da macro. Para introduzir um caractere fim de linha, digite a seqüência <ENTER>.

Para introduzir caracteres através de seu valor decimal, digite a seqüência \nnn•, onde o caractere “•” é branco. Este caractere é consumido ao converter a seqüência para o formato interno. O valor de nnn não deve ser maior do que 127. Esta forma serve, por exemplo, para introduzir caracteres de tabulação (\008•) no texto no qual a macro venha a ser executada.

A edição do nome e do corpo é realizada pelo **Editor de Formulários**. Ao editar o corpo de uma macro ou lição podem ser realizadas quaisquer operações, inclusive as de ler e gravar arquivos. A edição do nome e do corpo é realizada pelo programa de formulários **Serviço**.

10.6 Exclusão de macro ou lição

Para excluir uma macro ou lição, siga as instruções acima até se posicionar no menu de seleção da macro ou lição. O botão:

- [DEL] exclui a macro ou lição.

10.7 Modos de execução de macros

A execução de uma macro ou lição pode ocorrer em diversos *modos*. Estes determinam o modo de interação do executor de lições com o usuário. A execução de uma macro sempre inicia no modo Contínuo.

Para mudar de modo de execução (ver abaixo), ou para cancelar a execução da lição, tecele **[F2]** durante a espera ou interpretação de uma tecla. Isto ativar um menu que permitir escolher um dos modos de execução, descritos a seguir.

- Passo as teclas são lidas e interpretadas uma a uma, passando-se para a seguinte por solicitação do usuário. Antes de interpretar uma tecla, esta será exibida no canto direito inferior da tela. Para prosseguir, tecele qualquer tecla, por exemplo <Espaço> ou {ESQ} (botão da esquerda do “mouse”).
- Tecla é semelhante a **Passo**, sendo que, para prosseguir, o usuário deve digitar a tecla exibida. Este modo de execução reforça o aprendizado.
- Retardo é semelhante a **Passo**, sendo que a tecla exibida é interpretada automaticamente após decorrido um pequeno tempo. O modo retardo é iniciado ao selecionar retardo Janela ou Retardo tecla no menu de seleção do modo de execução de lições. Os valores destes itens definem o tempo de espera, em milissegundos, após ao qual a execução prosseguirá. A interrupção do processamento em modo retardo pode ser realizada apertando-se qualquer tecla, exceto a tecla [ESC].
- Contínuo as teclas são interpretadas imediatamente sem retardo e sem que sejam exibidas no vídeo. De maneira geral o vídeo não é redesenhado. O vídeo será redesenhado ao terminar a execução da lição, ou, se necessário, antes de exibir uma janela explanatória. A interrupção do processamento em modo contínuo pode ser realizada apertando-se qualquer tecla, exceto a tecla [ESC].

- Termina** cancela a execução da lição. A base de software permanecerá aberta. No entanto, se ela for um arquivo temporário (caso normal em lições), qualquer edição realizada será perdida.
- Erro de execução** sinaliza mensagens e alarmes sonoros. Tecle qualquer coisa e depois termine a execução da lição. Erros de execução ocorrem se o contexto inicial de execução não estiver de acordo com o esperado pela lição. Isto ocorre, tipicamente, se a base de conhecimento estiver alterada. Para evitar esta situação, opere lições sempre a partir da base de conhecimento original.

10.8 Macros instaladas

Estão instaladas diversas macros de teclado. Estas macros assumem um contexto de desenvolvimento de programas dirigido para “C”. A seguir descreveremos sucintamente estas macros. Nas macros apresentadas, a parte inicial identifica a tecla da macro, a ser utilizado em comando **[!X tecla]**. A seguir é exibido o título. Ao final é dada uma breve explanação do objetivo da macro.

10.8.1 Macros do Editor de Formulários

c	case “C”	gera duas linhas contendo case e break.
d	gera #define	
f	for “C”	gera as linhas de um comando for.
I	if “C”	gera as linhas de um comando if.
N	gera #include	
s	switch “C”	gera as linhas de um comando switch.
U	gera unsigned short	
w	while “C”	gera as linhas de um comando while.
!Z	desloca para a direita	desloca a linha 3 caracteres para direita, inserindo brancos.
!X	desloca para esquerda	desloca a linha 3 para esquerda, eliminando os três primeiros caracteres.
!B	alinha	alinha a margem direita
!s	se Talisman	gera as linhas de um comando se em Talisman.
!e	enquanto Talisman	gera as linhas de um comando enquanto em Talisman.
!p	ParaTodos Talisman	gera as linhas de um comando ParaTodos em Talisman
!c	ComObjeto Talisman	gera as linhas de um comando ComObjeto em Talisman.

Página propositalmente deixada em branco

11. MENUS E DIÁLOGOS

Menus são utilizados para selecionar um ou mais itens em um conjunto de itens. *Diálogos* são utilizados para o fornecimento de dados ao sistema. Em TALISMAN campos de diálogo podem ser utilizados também como selecionadores. Isto torna uniforme o tratamento de diálogos e de menus.

Menus podem ser, ainda, *constantes* ou *listas variáveis*. Um menu constante é utilizado para selecionar ações a serem realizadas. Menus de listas de variáveis são utilizados para selecionar um ou mais objetos sobre os quais será realizada alguma ação. A figura 11.1 ilustra um menu lista variável.

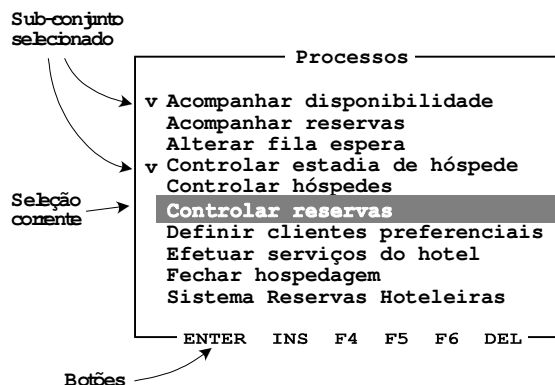


Figura 11.1. Exemplo de janela de menu

11.1 Conceitos básicos

Cada menu possui um ou mais *itens*. Um item pode ser *selecionável*, neste caso existirá pelo menos uma ação disparada por este item. Para um mesmo item podem ser disparadas diferentes ações, neste caso são utilizados *botões* para determinar a ação específica a ser realizada com o item selecionado. Caso possam ser executadas várias ações, o botão **[F6]** ativa um menu no qual poderá ser selecionada a ação.

Itens podem ser *editáveis* pelo usuário. Ou seja, um item editável corresponde a um diálogo. Os itens editáveis podem ser *predefinidos*, admitindo a seleção de um entre vários valores de um conjunto de valores predefinidos. Os itens editáveis podem ser *valor*, admitindo uma sequência de caracteres.

Menus permitem:

- Seleção simples* seleciona um item.
- Seleção em conjunto* seleciona um ou mais itens de um conjunto variável de itens.
- Operação sobre elementos de conjunto* efetua uma operação selecionada sobre um ou mais elementos de um conjunto variável de itens. A operação é selecionada por meio de *botões*.
- Edição de item com valor predefinido* em um item editável, seleciona um valor de um conjunto de valores predefinidos.
- Edição de item valor* em um item editável, edita a sequência de caracteres do item.

11.2 Comandos padronizados

- [ENTER] *item simples*: executa a ação associada com o item. *Item editável*: caso tenha ocorrido alguma edição, confirma o novo valor. Não tendo ocorrido edição, executa a ação associada com o item.
- [ESC] *item simples*: cancela a execução do menu. *Item editável*: caso tenha ocorrido alguma edição, restaura o valor anterior. Não tendo ocorrido edição, cancela o menu.
- [F3] *ao editar itens valor editável*: confirma o valor editado e efetua a ação associada com o item.

11.3 Selecionar subconjunto

Dependendo do contexto, menus de seleção de conjunto permitem a seleção de um subconjunto de zero ou mais elementos, ver figura acima. A ação selecionada operará sobre todos os elementos deste subconjunto. Marca-se os elementos de um subconjunto com:

- [SP] marca/desmarca item de subconjunto. A execução seguirá a ordem em que foram marcados os itens.
- [ENTER] marca o item corrente e termina a seleção do subconjunto.
- [ESC] cancela a seleção.

11.4 Selecionar ação, botões

Dependendo do contexto, menus de seleção admitem a realização de uma ação dentro de um conjunto de ações, onde esta ação atua sobre todos os elementos selecionados. O conjunto de ações disponíveis em um dado menu estarão exibidos na margem inferior deste menu. Cada botão define uma operação a ser efetuada com relação ao elemento ou conjunto de elementos selecionados. Os botões disponíveis são:

- [F6] ativa um menu contendo todas as ações que podem ser realizadas com os elementos selecionados. Além das ações ativadas diretamente pelos botões listados abaixo, podem existir outras ativáveis somente por intermédio do botão [F6].
- [INS] insere um elemento no conjunto.
- [F4] edita o formulário *Editar Nome da classe* dos elementos selecionados (ver capítulo **Editor de Formulários**).
- [F5] edita o formulário *Criar Nome da classe* dos elementos selecionados.
- [DEL] exclui os elementos selecionados.

11.5 Posicionar item

- [setas] posiciona no item selecionável na direção da seta. Em itens contendo valor editável, os comandos [ESQ] e [DIR] posicionam o cursor de edição. Para posicionar em itens à esquerda ou à direita, utilize os comandos [^ESQ] e [^DIR] descritos abaixo.
- [^ESQ] posiciona o cursor no item selecionável anterior.
- [^DIR] posiciona o cursor no item selecionável a seguir.
- [PGUP] posiciona o cursor uma janela acima.
- [PGDN] posiciona o cursor uma janela abaixo.
- [HOME] posiciona o cursor no primeiro item do menu.
- [END] posiciona o cursor no último item do menu.
- [letra] *menu contém item selecionável por [letra]:* posiciona o cursor neste item e executa a ação correspondente. A letra que seleciona um item é exibida como maiúscula. As demais letras são minúsculas. *Menu contém item com campo editável e selecionável por [letra]:* somente posiciona no campo, a seguir espera ações de edição.

Observação [letra] não vale para menus de seleção em conjuntos.

11.6 Editar itens contendo valores predefinidos

- [ESPAÇO] muda para a próxima opção dentro do conjunto. Caso esteja na última opção, muda para a primeira opção do conjunto.
- [letra] caso os valores do conjunto predefinido sejam selecionáveis, seleciona o valor predefinido associado a esta letra.

- [ENTER] *caso tenham ocorrido uma ou mais trocas:* confirma o valor corrente. *Caso não tenha ocorrido troca após última confirmação:* efetua a ação associada com o item. *Observação:* para trocar e selecionar são necessários dois [ENTER], o primeiro confirma, o segundo seleciona.
- [ESC] ou [F3] ou [^F3] *caso tenham ocorrido uma ou mais trocas:* restaura o valor anterior. *Caso não tenha ocorrido troca após última confirmação:* cancela o menu corrente. *Observação:* para cancelar o menu são usualmente necessários dois [ESC], o primeiro restaura o valor inicial, o segundo cancela o menu.

11.7 Editar item valor

- [ENTER] *caso tenham ocorrido uma ou mais edições:* registra o valor corrente. *Caso não tenha ocorrido edição após última confirmação:* efetua a ação associada com o item. *Observação:* para editar e selecionar são necessários dois [ENTER], o primeiro registra, o segundo seleciona. Para confirmar e selecionar em uma única ação, utilize o comando [F3] descrito abaixo.
- [ESC] *caso tenham ocorrido uma ou mais edições:* restaura o valor anterior. *Caso não tenha ocorrido edição após última confirmação:* cancela o menu corrente. *Observação:* para cancelar o menu são usualmente necessários dois [ESC], o primeiro restaura o último valor registrado (valor inicial), o segundo cancela o menu.
- [^ESQ] *no início de campo valor:* posiciona no campo anterior do menu. *No meio ou final de campo valor:* posiciona na palavra anterior do valor.
- [^DIR] *no final de campo valor:* posiciona no campo a seguir do menu. *No início ou meio de campo valor:* posiciona na palavra a seguir do valor. Os delimitadores de palavra e as regras de identificação de palavras estão descritos no capítulo **Editor de Formulários**.
- [DEL] exclui o caractere sob o cursor.
- [BKSP] exclui o caractere à esquerda do cursor.
- [F8 DEL] exclui todos os caracteres do cursor, inclusive, até o final do valor.
- [F8 P] exclui a palavra a seguir.
- [F8 L] exclui todos os caracteres do campo.
- [F2] ativa um menu para a edição dos parâmetros do editor de valores.
- [^F2] ativa um menu contendo caracteres não digitáveis normalmente, o caractere escolhido ser inserido no local do cursor.
- [F3] termina a edição do valor e seleciona a ação correspondente ao item do menu.

11.8 Modos de edição de valor

- [INS] inverte o modo de inserção, de **insere** para **sobre-escreve** e vice-versa.
- [!A] inverte o modo de acentuação, de **acentua** para **não acentua** e vice-versa. As regras de acentuação estão descritas no capítulo **Editor de Formulários**.

Página propositalmente deixada em branco

12. AUXÍLIO E TREINAMENTO

12.1 Condição básica para o uso

O subsistemas de auxílio e de treinamento de Talisman poderão ser utilizados somente se a base de auxílio, arquivo TALISMAN.BAX, estiver disponível. Não estando disponível, será emitida uma mensagem de erro, sempre que se tentar utilizar um destes subsistemas. De resto, o ambiente Talisman operará normalmente, mesmo sem a presença da base de auxílio.

[F1] *fora do executor de menus:* ativa a régua de comandos do editor corrente. Para ativar o sistema de auxílio, selecione a opção **Auxílio** nesta régua, ou, então, ative mais uma vez [F1]. *Dentro do executor de menus:* ativa a tela de auxílio correspondente a este menu.

O sistema de auxílio é operado como um menu, ver capítulo **Menus e Diálogos**. Em todas as telas de auxílio existirão itens selecionáveis que permitirão *explorar* o sistema de auxílio ou terminar o seu uso. Os itens selecionáveis são realçados.

A cada momento a linha superior do vídeo exibe o **Estado Corrente** em que se encontra o ambiente, ver capítulo **Convenções**. O estado corrente identifica o comportamento corrente do ambiente. Quando em dúvida, examine o conteúdo deste campo.

O sub-sistema de auxílio opera, ou em função do menu corrente, ou a partir da régua de comandos do editor corrente. Cada editor possuía sua régua de comandos. Além de permitir a escolha do comando via uma sucessão de menus, esta régua serve, ainda, de índice ao sistema de auxílio. Caso o ambiente se encontre operando sobre um menu ou diálogo, o auxílio será exibido em conformidade com este menu.

12.2 Lições

Para facilitar o aprendizado do seu uso, Talisman possui um sub-sistema de *instrução programada*. Este sub-sistema executa *lições*. Uma lição é uma *seqüência de teclas*, entremeada de *janelas explanatórias*. Na realidade, uma lição é uma *macro de teclado*, ver capítulo **Macros e Lições**. A seqüência de teclas é *interpretada* pelo processador de lições, exibindo para o usuário o efeito destas teclas à medida que vão sendo interpretadas. As lições contêm janelas explanatórias explicando em mais detalhe o efeito dos comandos sendo executados.

Ao executar uma lição, usualmente a próxima tecla a ser interpretada será exibida no canto direito inferior da tela. *Teclas caractere* são identificadas pelo caractere correspondente, *teclas de função* são identificadas pelo nome contido entre parêntesis angulares, por exemplo <ENTER> ou <DESCE>. Em adição às teclas mencionadas, existem diversas *funções internas* identificadas por teclas de função. Estas teclas também são exibidas contidas em parêntesis angulares. Por exemplo, a função *espera por ação do usuário* é exibida sob a forma <Espera>.

Lições podem ser criadas e editadas pelo usuário. Isto permite, por exemplo, a criação de instruções de uso de normas técnicas específicas da instalação. A criação e edição de lições é idêntica à criação e edição de macros de teclado, ver capítulo **Macros e Lições**.

Observação 1 o sistema de treinamento depende do estado da base de conhecimento. Caso esta esteja inconsistente com a lição (por exemplo devido à alteração de parâmetros), a execução da lição poderá ser cancelada por erro de execução. Para evitar estes problemas, opere sempre com uma cópia original da base de conhecimento. Por convenção, a primeira janela explanatória da lição indicará as condições especiais que devem ser satisfeitas para que a lição possa ser executada. Cancele a lição caso estes programas de formulário não estejam instalados. Para isto tecle [ESC] para sair da janela, e tecle [F2] logo a seguir (tecla exibida é <Espera>).

Observação 2 para evitar a interferência do processador de lições com bases de software porventura existentes, as bases utilizadas em lições são arquivos temporários. Arquivos temporários possuem a extensão .\$\$*. Ao fechar um arquivo temporário, ele será automaticamente excluído do sistema de arquivos.

12.2.1 Seleção da lição a executar

Estando no **Menu Principal** *sem base de software aberta*, situação padrão ao iniciar, selecione o item **selecionar treinamento**. Ser ativado um menu contendo a lista de lições disponíveis.

Para executar uma lição basta selecioná-la, comando [ENTER] sobre o nome da lição a ser executada.

A execução de uma lição sempre inicia em modo **passo**, ver abaixo. Durante a execução da lição o modo de processamento pode ser mudado ou automaticamente, ou por ação do usuário.

Sempre que uma janela explanatória for aberta, leia-a atentamente. Para prosseguir, tecla [ESC].

12.2.2 Modos de execução de lições

A execução de uma lição pode ocorrer em diversos *modos*. Estes determinam o modo de interação do executor de lições com o usuário.

Para mudar de modo de execução (ver abaixo), ou para cancelara execução da lição, tecla [F2] durante a espera ou interpretação de uma tecla. Isto ativará um menu que permitirá escolher um dos modos de execução, descritos a seguir.

Passo	as teclas são lidas e interpretadas uma a uma, passando-se para a seguinte por solicitação do usuário. Antes de interpretar uma tecla, esta será exibida no canto direito inferior da tela. Para prosseguir, tecla qualquer tecla, por exemplo <Espaço>.
Tecla	é semelhante a Passo , sendo que, para prosseguir, o usuário deve digitar a tecla exibida. Este modo de execução reforça o aprendizado.
Retardo	é semelhante a Passo , sendo que a tecla exibida é interpretada automaticamente após decorrido um pequeno tempo. O modo retardo é iniciado ao selecionar retardo Janela ou Retardo tecla no menu de seleção de modo de execução de lições. Os valores destes itens definem o tempo de espera, em milisegundos, após ao qual a execução prosseguir. A interrupção do processamento em modo retardo pode ser realizada apertando-se qualquer tecla, exceto a tecla [ESC].
Contínuo	as teclas são interpretadas imediatamente sem retardo e sem que sejam exibidas no vídeo. De maneira geral o vídeo não é redesenhado. O vídeo será redesenhado ao terminar a execução da lição, ou, se necessário, antes de exibir uma janela explanatória. A interrupção do processamento em modo contínuo pode ser realizada apertando-se qualquer tecla, exceto a tecla [ESC].
Termina	cancela a execução da lição. A base de software permanecerá aberta. No entanto, se ela for um arquivo temporário (caso normal em lições), qualquer edição realizada será perdida.
Erro de execução	sinaliza mensagens. Tecla qualquer coisa e depois termine a execução da lição. Erros de execução ocorrem se o contexto inicial de execução não estiver de acordo com o esperado pela lição. Isto ocorre, tipicamente, se a base de conhecimento estiver alterada. Para evitar esta situação, opere lições sempre a partir da base de conhecimento original.

13. IMPORTAR E EXPORTAR

Os mecanismos de importação e exportação destinam-se a:

- estabelecer mecanismos de comunicação entre diferentes bases de software. Por exemplo, pode-se gerar bases iniciais contendo porções do dicionário de dados corporativo. Estas porções iniciais correspondem a visões parciais do dicionário corporativo como um todo. Em sentido inverso, pode-se integrar bases de trabalho com bases centralizadas.
- estabelecer mecanismos de comunicação entre diferentes ambientes e/ou processadores. Por exemplo, através do mecanismo de importação pode-se importar as definições de organização de bancos de dados. Já através da exportação pode-se gerar arquivos apropriados a serem importados por estes mesmos bancos de dados.
- estabelecer mecanismos de transformação que requeiram processamento externo a **Talisman**. Por exemplo, pode-se exportar uma sequência de comandos de exclusão. Estes comandos são verificados externamente. Ao serem importados, os elementos que permaneceram no arquivo de exclusão serão excluídos.
- estabelecer mecanismos de engenharia reversa. Com o uso de analisadores sintáticos devidamente programados, pode-se converter código fonte em arquivos de importação, internalizando desta forma o projeto de um programa para a linguagem projeto estruturado.

A *exportação* é efetuada por programas de formulário. Conseqüentemente, a sua capacidade de geração atende a virtualmente todos os formatos dos diferentes processadores para os quais se deseja exportar.

A *importação* é efetuada por um processador especializado e depende de um formato próprio. Este formato é definido mais adiante neste capítulo. A importação pode ser utilizada para adicionar objetos e/ou atributos à base de software, bem como para excluir objetos e/ou atributos. Como já foi mencionado, é relativamente simples criar analisadores sintáticos capazes de produzirem texto no formato de importação.

13.1 Exportação

A exportação é realizada por intermédio de programas de formulário, ver capítulo **Programação de Formulários**. A seleção dos objetos origem, bem como do formulário origem, é realizada através do **Editor de Dicionários**, comandos [**>**] Exportar usando formulário *default* e [**<**] Exportar selecionado usando um formulário selecionado. Em linhas gerais, a exportação segue as mesmas etapas que a linearização de estruturas modulares, ver capítulo **Estruturas Modulares**, seção Linearização.

13.2 Importação

A importação é realizada lendo-se *comandos de importação* e *valores a importar* a partir de um arquivo sequencial. A importação pode ser limitada segundo algumas regras:

- importar somente objetos novos.
- importar somente objetos existentes.
- importar objetos quaisquer.
- importar somente atributos novos.
- importar somente atributos existentes.
- importar atributos quaisquer.

Os comandos de importação podem ser:

- importação de objetos.
- importação de atributos.
- exclusão de objetos.
- exclusão de atributos.

Caso ocorra um erro de sintaxe (erro de formação do arquivo de importação), é exibida uma mensagem no vídeo e a importação é descontinuada. Conflitos decorrentes de restrições de importação produzem mensagens, porém não cancelam a execução. As mensagens podem ser dirigidas para um arquivo LOG. Este tem um formato semelhante ao de um arquivo de importação. Isto permite que se possa corrigir e, posteriormente, importar este arquivo,

Ao importar um valor *nome*, está-se importando um *objeto*. Sendo permitido criar objetos e o nome sendo o de um objeto novo, será criado o objeto. Já ao importar valores *não nome*, o objeto correspondente precisa *existir* para que o atributo possa ser importado. Sugere-se que os arquivos de importação sejam formados em duas partes:

- | | |
|----------------|--|
| Primeira parte | contém somente comandos de importação de nomes. Assegura-se, assim, que todos os objetos necessários tenham sido criados, antes que os correspondentes atributos sejam importados. |
| Segunda parte | contém somente os comandos de importação de atributos não nome. |

13.3 Ativação da importação

A importação é ativada a partir do **Menu Principal**.

A importação de objetos corresponde à importação de *nomes*. No menu de importação, o item *Modo objeto* admite as seguintes respostas:

- | | |
|-----------|---|
| Cria/usa | se um objeto sendo importado tem nome igual a outro objeto da classe de objetos sendo importada, o objeto importado é considerado igual ao objeto já definido. Se o objeto importado não possui nome igual a outro objeto já definido, ele será criado. |
| Só novo | se um objeto sendo importado tem nome igual a outro objeto da classe sendo importada, será gerada uma mensagem de erro de importação. Se o objeto importado não possui nome igual a outro objeto já definido, ele será criado. |
| Só existe | se um objeto sendo importado tem nome igual a outro objeto da classe sendo importada é considerado igual ao objeto já definido. Se o objeto importado não possui nome igual a outro objeto já definido, será gerada uma mensagem de erro de importação. |

Ao importar um valor *não nome*, está-se importando um atributo de um objeto. Ao importar um atributo, o correspondente objeto precisa estar definido. Caso não esteja, será gerada uma mensagem de erro de importação. Para evitar estes erros, importe primeiro os objetos (atributo **Nome**) e depois os atributos correspondentes, ver seção **Formato do Arquivo de Importação** mais adiante neste capítulo.

O item *Modo atributo* admite as seguintes respostas:

- | | |
|-----------|---|
| Cria/usa | se um atributo sendo importado tem identificador (ou índice) de atributo igual a um índice de um atributo já definido para o correspondente objeto, o valor importado será copiado para a base de software, sobre-gravando o valor existente. Se o valor do atributo importado for vazio, o atributo será eliminado do objeto correspondente. Se o atributo importado não existir no objeto, ele será criado com valor igual ao valor lido. |
| Só novo | se um atributo sendo importado tem nome (ou índice) igual a de atributo já definido, será gerada uma mensagem de erro de importação. Se o atributo importado não existir no objeto, ele será criado com valor igual ao valor lido. |
| Só existe | se um atributo sendo importado tem nome (ou índice) igual a de atributo já definido, o valor importado será copiado para a base de software. Se o atributo importado não existir no objeto, será gerada uma mensagem de erro de importação. |

O item *Modo Log* define com serão processadas as mensagens de erro. Os valores possíveis deste item são:

- | | |
|---------------|---|
| Não registra | cada mensagem de erro de importação será exibida no vídeo. |
| Registra tudo | todas as mensagens serão copiadas para um arquivo seqüencial de registro dos erros de importação. Neste arquivo, as mensagens de erro são apresentadas sob a forma de comentário de arquivo de importação. Este arquivo tem formato compatível com o arquivo de importação. Isto permite examinar a causa das mensagens, saná-las e, posteriormente, submeter o arquivo de erro como se fosse arquivo de importação. Erros de sintaxe (for- |

mação do arquivo de importação) são exibidas tanto no arquivo de registro de erros, como no vídeo. Caso ocorra um erro de sintaxe, a importação é descontinuada.

13.4 Formato do arquivo de importação

O arquivo de importação obedece à seguinte gramática:

```

<Arquivo de importação> ::= 0-[ <Comando de importação> ]
                               /* Até fim de arquivo */ .

<Comando de importação> ::= (   <Comentário>
                               | <Importação>
                               | <Exclusão de objeto>
                               | <Exclusão de atributo> ) .

<Comentário>   ::= /* inicia com * e termina em fim de linha */

<Importação>   ::= DA <Classe objeto> <Nome objeto>
                  <Atributo a importar> .

<Atributo a importar> ::= (   <Nome>
                              | <Aliás>
                              | <Texto>
                              | <Relação> ) .

<Nome>         ::= Nome .
<Aliás>        ::= Aliás <Identificação atributo> <Valor aliás> .
<Texto>        ::= Texto <Identificação atributo> <Lista texto> .
<Relação>      ::= Relação <Identificação atributo> <Lista texto> .

<Identificação atributo> ::= /* Nome simbólico ou número
                              identificador do atributo,
                              ver Programação de formulários */ .

<Valor Aliás> ::= /* linha a seguir da identificação do Aliás,
                  contendo 0 ou mais caracteres até o
                  fim da linha */ .

<Lista texto> ::= /* lista contendo as zero ou mais linhas a seguir
                  da identificação de atributo texto ou relação,
                  terminando com uma linha contendo exatamente
                  #@# alinhado na margem esquerda */ .

<Exclusão de objeto>   ::= XO <Classe objeto> <Nome objeto> .

<Exclusão de atributo> ::= XA <Classe objeto> <Nome objeto>
                          <Atributo a excluir>
                          <Identificação de atributo>

<Atributo a excluir>   ::= (   Aliás
                              | Texto
                              | Relação ) .
    
```

Observações:

- os valores simbólicos estão definidos nos apêndices Tabela de Linguagens de Representação e Tabela de Palavras Reservadas.
- as linhas de valor de atributos (*aliás*, *texto* e *relação*) começam sempre em linha nova imediatamente após à linha que contém o identificador de atributo.
- linhas de valor de atributo são copiadas tal como fornecidas, sem se preocupar com possíveis endentações à esquerda.
- comentários iniciam no primeiro caractere “*” e terminam no caractere fim de linha da correspondente linha de código de importação.

- comentários valem somente no meio de texto de código de importação. Ou seja, comentários não podem ser inseridos em linhas de valor de atributo, tampouco podem estar presentes após o identificador de atributo e antes da correspondente linha de valor.
- exceto as linhas de valor, todas as linhas são livres de formato.
- a importação de atributos **Nome** corresponde à importação de um objeto. Os demais atributos importados, efetivamente geram atributos no objeto identificado.
- em atributos **relação**, a lista texto contém os nomes dos objetos relacionados, um por linha. Ao processar a importação de uma relação, serão criados objetos para cada nome ainda não definido no dicionário, desde que o modo de importação de nome seja **Só novo** ou **Cria/Usa**. Caso o modo seja **Só existe**, será gerada uma mensagem de erro de importação.
- caso seja importada uma relação e sendo necessário que esta relação fique ordenada, por exemplo ao importar estruturas de programas, importe somente esta relação e *nenhuma* de suas inversas. Se for importante importar as inversas também, assegure que a relação ordenada esteja antes das correspondentes inversas.

No exemplo a seguir ilustramos um arquivo de importação simples.

```
* Lista dos objetos a importar
*
DA "Pacotes programa" "Rotina raiz quadrada" Nome
DA "Blocos programa" "Corpo de Rotina raiz quadrada" Nome
DA "Blocos programa" "Calcular aproximação inicial" Nome
DA "Blocos programa" "Calcular valores cada vez mais próximos" Nome
DA "Blocos programa" "Calcular nova aproximação" Nome
*
* Atributos dos objetos
*
DA "Pacotes programa" "Rotina raiz quadrada" Texto 3
X é assumido ser sempre maior do que zero
#@#
DA "Pacotes programa" "Rotina raiz quadrada" Texto 7
Calcular o valor da raiz quadrada de um número
#@#
DA "Pacotes programa" "Rotina raiz quadrada" Texto 21
X - valor do qual se deve extrair a raiz
#@#
DA "Pacotes programa" "Rotina raiz quadrada" Relação Blocos
Corpo de Rotina raiz quadrada
#@#
DA "Blocos programa" "Calcular valores cada vez mais próximos" Relação
Decomp
Calcular nova aproximação
#@#
DA "Blocos programa" "Corpo de Rotina raiz quadrada" Texto 18
float RaizQuadrada( float X )
#@#
DA "Blocos programa" "Corpo de Rotina raiz quadrada" Relação 57
Calcular aproximação inicial
Calcular valores cada vez mais próximos
#@#
*fim de arquivo de importação
```


14. RECUPERAÇÃO E OTIMIZAÇÃO DE BASES

A base de software é um componente vulnerável de um ambiente de desenvolvimento. Qualquer corrupção de seu conteúdo pode por meses de trabalho a perder. Para evitar possíveis problemas, foram tomadas diversas precauções de projeto ao desenvolver Talisman. Entre elas:

- a base de software é atualizada automaticamente por intermédio de um mecanismo *auto-save*. Este é ativado após completar a geração dos dados de uma transação, e é executado nos ciclos ociosos da máquina. Em geral, poucos décimos de segundo após ativar uma transação (“commit”), a base estará atualizada. No entanto, podem ocorrer problemas como veremos mais adiante.
- a base de software é virtualmente protegida contra mau funcionamento do próprio Talisman. Esta proteção foi implementada na forma de assertivas executáveis. Estas consistem em código de controle de integridade incluído em pontos estratégicos do programa. Sempre que estes códigos encontrarem estruturas de dados incorretas, eles interceptarão a execução do programa. Impedem, assim, que uma transação estruturalmente incorreta possa corromper a base de software. Evidentemente, nem todas as causas de defeitos podem ser interceptadas.

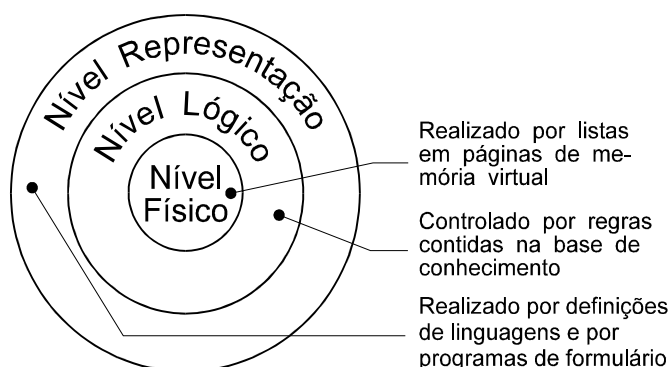


Figura 14.1. Níveis organizacionais da base de software

A seguir descrevemos os níveis que compõem a base de software, ver figura 14.1. O primeiro nível (nível físico) é formado por um sistema de listas operando em memória virtual simulada. Em geral, erros de uso, ou do contexto operacional (energia, sistema operacional, etc.), afetam este nível. O segundo nível (nível lógico) é formado por um conjunto de dicionários, objetos, atributos de objetos e relacionamentos entre objetos. A organização deste nível obedece a regras precisas. Estas regras se encontram na base de conhecimento. A organização dos dados neste nível é rígida e independe da configuração do meta-ambiente. Em geral, erros de funcionamento de Talisman afetam esta organização. O terceiro nível (nível de representação) corresponde às representações contidas na base de software. Esta interpretação é configurável. A validação dos dados neste terceiro nível é realizada pelos programas de formulário de validação. Ou seja, no terceiro nível a organização válida dos dados é flexível e determinada por programa redigido pelo usuário. Os erros encontrados neste nível decorrem de projetos incompletamente e/ou incorretamente criados pelo usuário. É, portanto, normal que, pelo menos durante algum tempo, existam erros neste nível.

Ao desenvolvermos o gerente de bases de software, procuramos utilizar sistematicamente estruturas robustas de dados. Estas estruturas possuem redundância cuidadosamente introduzida, permitindo, assim, a detecção de erros nas diversas estruturas de dados armazenadas. Esta redundância é necessária para que se possa validar a organização dos dados, caso as estruturas estejam comprometidas. Note que bits de paridade e bits de controle contidos em mensagens são redundâncias introduzidas para controlar o funcionamento de equipamento de computação.

14.1 Causas para erros em bases de software

Apesar de todos estes esforços, a estrutura da base de software poder ser corrompida. Podem comprometer a base de software:

- Erro de ambiente ou de equipamento estes são consequência de quaisquer eventos externos que interrompam e cancelem subitamente a execução de Talisman. São exemplos: erro de gravação no disco, erro de paridade da CPU, falta de energia enquanto a transação está sendo completada, etc.

- Erro de operação** estes são consequência de ações de operação que interrompem a execução de *Talisman*, diferentes da saída normal através do menu principal. São exemplos: interromper a exceção com um “reset” da máquina; desligar a máquina sem sair de *Talisman*; existindo aceleradores de disco (“cache”), desligar a máquina antes que este termine de gravar tudo o que está em sua área de comunicação, etc.
- Erro de uso de ambientes tipo Windows** estes são consequência de uso incorreto, ou desavisado, de *Talisman* em ambientes de multiprogramados. *Talisman* é mono-usuário e não está preparado para controlar o compartilhamento de bases de dados. Qualquer tentativa de compartilhamento de bases, provocará erros de execução, podendo comprometer a integridade da base de software. Mais especificamente: ao utilizar Windows, deve-se assegurar que cada *Talisman* ativo não compartilhe as bases de software, de conhecimento, de auxílio e de trabalho.
- Erro de uso de redes** estes são consequência de uso incorreto, ou desavisado, de *Talisman* em ambientes distribuídos. Como já foi mencionado, *Talisman* é mono-usuário e não está preparado para controlar o compartilhamento de bases de dados. Qualquer tentativa de compartilhamento de bases provocar erros de execução, podendo comprometer a integridade da base de software. Mais especificamente, qualquer tentativa de compartilhamento de bases, local e/ou remotamente, provocar erros de execução, podendo comprometer a integridade da base de software.
- Erro de software** é claro que, apesar de todos os nossos esforços, podem ainda existir erros no software capazes de corromper a base de software.
- Evolução de versão** ao validar e recuperar bases, diversas evoluções de versão serão corrigidas. De modo geral as variações de versão afetam aspectos de atributos transparentes (irrelevantes) para o correto funcionamento do sistema. No entanto, optamos por uniformizar todos os atributos, mesmo que isto não tenha impacto sobre o correto funcionamento do sistema.

14.2 O programa TALISVAL

Para evitar a perda irrecuperável de trabalho, foi desenvolvido o programa *TALISVAL*. Este programa tem por finalidade:

- recuperar bases de software por mais corrompidas que estejam.
- reorganizar a sequência de registros, otimizando a manipulação destes.

14.2.1 Operação básica de TALISVAL

Em linhas gerais, a operação de *TALISVAL* é idêntica à descrita para *Talisman*.

Assegure-se que nenhuma outra instância ativa de *Talisman* ou *TALISVAL* esteja operando e utilizando arquivos que virão a ser utilizados pela instância de *TALISVAL* a ser ativada. Estes arquivos são: base de conhecimento, base de auxílio, arquivo de trabalho e bases de software. Note que isto somente poderá ocorrer quando executando em rede, em Windows, ou ao utilizar algum software que permita operar diversos programas quase simultaneamente. Ative o programa com o comando:

```
TALISMAN [/v<vídeo>] [/l<linhas texto>] [<base de software>]
```

A ordem de apresentação dos parâmetros é qualquer, e todos os parâmetros são opcionais. O significado dos parâmetros está descrito em 2.5.3 *Ativação de Talisval*.

Após carregado na memória, será exibido o “logo” do programa. Apertando qualquer tecla, o programa validará a base de conhecimento. Se esta estiver correta, será ativado o **Menu Principal** do programa *TALISVAL*.

A base de conhecimento utilizada por *TALISVAL* é idêntica à utilizada pelo ambiente *Talisman*. No entanto, para evitar o compartilhamento accidental, recomenda-se que *TALISVAL* seja mantido em um diretório distinto de *Talisman*. Neste diretório deve ser mantida uma cópia operacionalmente completa, formada pelo programa, pela base de software e pelos “drivers” de vídeo.

O **Menu de Serviço** permite configurar alguns dos parâmetros da base de conhecimento. A operação deste menu e dos configuradores é idêntica ao descrito no capítulo 9 *MENU SERVIÇOS*.

14.3 Recuperar uma base de software

Abra a base de software a recuperar. Para isto proceda tal como descrito no capítulo **Menu Principal**. Como a recuperação de uma base de software se dá sobre o próprio arquivo que contém a base a recuperar, é fortemente recomendado gerar um “backup” da base antes de se realizar a recuperação.

Ative a ação **Recuperar**. Será perguntado se as mensagens devem ser enviadas para um arquivo ou para o vídeo. Recomenda-se que as mensagens sejam enviadas para um arquivo.

Caso as mensagens devam ser enviadas para um arquivo, será solicitado o nome do arquivo LOG. A seguir será iniciada a recuperação. Caso as mensagens devam ser enviadas para o vídeo, a recuperação inicia imediatamente. Sempre que a tela encha de mensagens, o sistema permanecerá parado até que seja teclado qualquer coisa. O progresso da recuperação será exibido no rodapé do vídeo. Terminada a recuperação, o programa volta ao **Menu Principal**.

As mensagens exibidas no vídeo ou transmitidas para o LOG de erros são idênticas. No entanto, dada a limitação de colunas do vídeo, a parte das mensagens que excede a largura do vídeo não será visualizada. É claro que esta restrição não afeta a geração do arquivo LOG de erros. Enfatiza-se, mais uma vez, a recomendação de enviar o LOG de recuperação para um arquivo.

A recuperação pode ser interrompida se desejado, independentemente de se as mensagens estão sendo dirigidas para LOG ou vídeo. No entanto, *caso seja interrompida, a base de software poderá estar corrompida!* Enfatizamos, mais uma vez, a conveniência de se gerar uma cópia “backup” da base antes de iniciar a recuperação. Para interromper a recuperação, aperte, e mantenha apertada, a tecla espaço (<SP>), até que apareça a janela de confirmação da interrupção. Após selecione se deseja ou não cancelar e tecla <ENTER>.

Ao recuperar a base de software, serão eliminados atributos em erro. Sempre que possível, a mensagem de recuperação conterá o valor dos atributos eliminados. Tendo sido gerado um LOG, os valores poderão ser copiados para a base utilizando os comandos de edição de arquivo seqüencial ([!Q]) e cópia de blocos ([F9 ENTER] e [F9 INS]) disponíveis no editor de formulários de Talisman.

14.4 Otimizar uma base de software

Ao editar a base de software, inserindo e/ou excluindo atributos, elementos de textos etc., os atributos de um mesmo objeto podem ficar muito dispersos no arquivo. Em adição, a estrutura interna da base impede que ela seja diminuída, mesmo que se tenha excluído um número grande de elementos. Isto tudo pode contribuir para uma redução do desempenho no acesso aos elementos contidos na base de software.

Para solucionar estes problemas, foi desenvolvido um módulo otimizador de bases de software. O otimizador copia a base desorganizada para uma nova base, reestruturando internamente todos os registros físicos. Note que o otimizador opera como um gerador de “backup”. Cabe salientar que o resultado do otimizador quando aplicado a uma base de software corrompida é indefinido. Assim, sempre que for otimizar uma base, efetue, antes, a recuperação desta base.

Caso ainda não esteja, abra a base de software a recuperar. Para isto proceda tal como descrito no capítulo **Menu Principal**. A seguir selecione o item **Otimizar** deste menu. Será solicitado o nome da base destino a ser gerada pelo otimizador. Não deve existir arquivo com nome igual a esta base.

14.5 Mensagens do recuperador de bases

Muitas das mensagens de erro informam pouco ao usuário que não conheça a estrutura interna das bases de software. Como nem sempre será possível utilizar valores simbólicos, as mensagens de erro apresentarão campos correspondentes a valores internos da base de software. Para o usuário normal, estes valores dizem muito pouco. Para tornar as mensagens um pouco mais inteligíveis, vamos descrever, superficialmente, a organização da base de software. Para um usuário mais experiente o LOG de mensagens gerado ao validar uma base de software pode servir de apoio ao diagnóstico das falhas de execução encontradas.

A base de software armazena objetos. Cada objeto possui um nome simbólico e uma chave única. Esta chave é conhecida por *Ido* (identificação de objeto). É tarefa dos dicionários converter os nomes simbólicos para os correspondentes Ido's. Alguns objetos pertencem a classes para as quais não está definido um dicionário. Um exemplo de classe de objeto não dicionarizável é a *cardinalidade* na linguagem **Entidades e Relacionamentos**.

Os atributos de um objeto, por exemplo fragmentos de texto, aliases, relações, etc., são conhecidas internamente por um número de identificação, o *índice do atributo*. Os capítulos descritivos da linguagens suportadas, bem

como o arquivo TALISMAN.LST fornecido junto com o sistema, permitem estabelecer a correspondência entre o índice do atributo e o nome simbólico de cada um dos atributos. Em adição, os atributos possuem um número identificador de classe de atributo, o *tipo do atributo*. Desta forma, ao acessar um atributo qualquer, o sistema sabe determinar como este atributo deve ser interpretado. Pode-se, assim, disparar processamentos específicos para relações, se o tipo do atributo for uma relação, ou específicas para texto, se o tipo do atributo for texto, etc.

Os diagramas (*folhas*) referenciam objetos por intermédio de instâncias destes objetos. Cada instância possui a sua chave única, gerada ao editar a folha. A chave de instância é interna à folha. Uma mesma chave de instância em folhas diferentes corresponde a instâncias diferentes. As instâncias, por sua vez, também possuem atributos. Estes são os atributos necessários para que se possa gerar os diagramas.

Finalmente, para evitar a geração de um número muito grande de diferentes mensagens, erros de natureza semelhante são sinalizados pela mesma mensagem, sendo diferenciados por intermédio de um *código*. Esta é uma solução de compromisso entre tamanho de tabela e engenharia humana. Optamos por minimizar o tamanho da tabela, uma vez que as mensagens serão de pouco uso para usuários que desconheçam a estrutura interna da base de software.

Nas seções a seguir listaremos todas as mensagens que podem ser geradas durante a recuperação. Para cada uma delas daremos, ainda, uma pequena explanação.

14.5.1 Sintaxe das mensagens de erro

- %n** é um número. Este número pode identificar diversas coisas. Por exemplo Páginas, objetos, relações, fragmentos de texto, etc. Os números identificadores são únicos. Desta forma pode-se determinar se um conjunto de erros diz respeito ao mesmo elemento. Quando uma mensagem menciona diversos números, cada uma conter um dígito à guisa de identificação no texto.
- %s** é um símbolo. Símbolos podem identificar diversas coisas, por exemplo o nome de um dicionário, o nome de um objeto, o nome de uma folha, etc.
- %cod** é um número, em geral negativo, que especializa a natureza do erro encontrado.

14.5.2 Mensagens indicativas das fases do recuperador

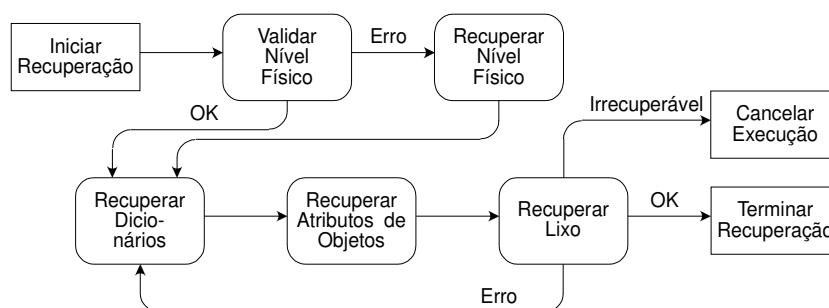


Figura 14.2. Seqüência de validação e recuperação

Estas mensagens meramente indicam o progresso da recuperação. Não correspondem a erros. A figura 14.2 ilustra a seqüência de fases executadas ao validar e recuperar uma base de software.

**** Validando listas

Durante esta fase será validada a estrutura física da base de software. Todos os atributos são armazenados em listas. Estas possuem regras de organização e redundância muito bem definidas. A redundância viabiliza verificar se a estrutura está correta. Caso a estrutura não esteja correta, ser ativada a fase **Recuperando BTREE**.

**** Recuperando BTREE

Esta fase somente será ativada se a estrutura de listas for encontrada em erro. Durante esta fase será gerada uma cópia dos atributos visando reconstruir a Btree. Assegura-se, assim, a validade do acesso aos dados armazenados na base de software. A validade lógica da base de software não é restabelecida durante esta fase.

**** Recuperando dicionários

Durante esta fase, é verificado se os elementos registrados nos diferentes dicionários de classes de objetos referenciam atributos (nomes de objetos) conhecidos. Elementos sem correspondentes atributos, bem como dicionários de classes desconhecidas serão eliminados.

**** Recuperando atributos

Durante esta fase é verificada a validade lógica do conteúdo da base de software. Esta validação é realizada utilizando-se as regras de organização lógica da base de software contidas na base de conhecimento. Atributos encontrados em erro são, em geral, eliminados.

**** Eliminando lixo

Durante esta fase, são eliminadas as páginas e as listas que não estejam vinculadas a algum atributo conhecido na base de software. De maneira geral, o ambiente Talisman elimina lixo durante o processamento. Porém é possível que lixo seja criado em consequência da eliminação de atributos em erro.

As fases **Recuperando dicionários**, **Recuperando atributos** e **Eliminando lixo** são efetuadas repetidas vezes, até que não seja encontrado mais erro algum. Ao eliminar um atributo em erro, pode-se propagar erros para outros atributos. Caso estes atributos sejam antecessores do atributo em erro, considerando a ordem com que são validados, é necessário repetir-se a validação para que os erros porventura propagados possam ser detectados na próxima passagem. O processo pára uma vez que os atributos em erro são eliminados.

Em situações extremas, destruição tão catastrófica que provoca erro no recuperador, a recuperação pode levar a uma estrutura de base de software inconsistente. Esta inconsistência se manifesta ao reconhecer a lista de listas ou as listas da Btree como sendo lixo. Neste caso, a execução de TALISVAL ser cancelada.

14.5.3 Erros da estrutura de acesso básica

Página %n -> contém lixo, desalocada (código %cod)

- Código -1 foi encontrada uma página que não contém nada aproveitável. Este erro é detectado durante a validação da estrutura de acesso (fase 1). Pode ser causado pelo mau funcionamento de hardware e/ou software.
- Código -2 foi encontrada uma página que não está vinculada a atributos contidos na base. Este erro é detectado durante a fase de eliminação de lixo. Usualmente, é um erro consequente da eliminação de atributos nas etapas anteriores.

Página %n -> lista livre cíclica

As páginas de memória virtual da base de software que porventura estejam sem uso, formam uma lista linear encadeada. Foi encontrado um ciclo nesta lista.

Página %n -> lista de listas mal formada (código %cod)

A base de software é formada por um conjunto de listas. Uma destas listas identifica todas as listas contidas na base de software. Por sua vez, cada lista possui uma auto-identificação capaz de verificar se está contida na lista de listas.

- Código -1 A lista de listas referencia uma lista desconhecida.
- Código -2 A página referenciada pela lista de listas não é origem de lista.
- Código -3 A página origem da lista de listas não é conhecida.
- Código -4 A página origem de uma lista não se encontra na lista de listas.
- Código -5 A página referenciada pela lista de listas não é conhecida.
- Código -6 A página referenciada pela lista de listas não é origem de lista.

Página %n -> erro de encadeamento (código %cod)

As listas são formadas por uma ou mais páginas. As páginas precisam satisfazer uma série de regras de encadeamento.

- Código -1 Página à esquerda desconhecida.
- Código -2 Página à esquerda não referencia a Página corrente.
- Código -3 Página à direita desconhecida.
- Código -4 Página à direita não referencia a Página corrente.

Página %n -> tipos vizinhos não iguais

As lista possuem tipo. Todas as páginas de uma mesma lista devem possuir o mesmo tipo.

Página %n -> erro de formação na BTREE (Código %cod)

A estrutura Btree é formada por um conjunto de listas encadeadas. Este conjunto deve satisfazer diversas regras de encadeamento.

- Código -1 A página estrutural pai não é conhecida.
- Código -2 O tipo da página estrutural pai não vale.
- Código -3 A página estrutural pai não contém chave referenciando a página corrente.
- Código -4 A chave na página estrutural pai não contém referência à página corrente.
- Código -5 Página que não pertence à estrutura Btree, possui tipo de página Btree.
- Código -6 Página raiz da estrutura Btree possui encadeamentos ilegais.
- Código >=0 O encadeamento com a cód-ésima página filho não vale.

Página %n -> chaves desordenadas (código %cod)

Cada elemento de uma lista possui uma chave. Diversas listas, por exemplo a Btree exigem que estas chaves estejam ordenadas.

- Código -1 A chaves dos elementos internos a uma página estão desordenados.
- Código -2 A chaves entre as páginas corrente e esquerda estão desordenadas.
- Código -3 A chaves entre as páginas corrente e direita estão desordenadas.

14.5.4 Erros em dicionários

Ido %n Dic %s -> não achou o objeto (código %cod)

O dicionário contém uma referência a um objeto que não se encontra na base de software. A referência é eliminada do dicionário. A classe de objetos é identificada pelo nome simbólico %s. Este erro é freqüentemente acompanhado de diversos erros de relacionamentos com objetos não conhecidos, ver abaixo. Para se certificar que os erros são relativos ao mesmo objeto, utilize o número identificador %n do objeto.

- Código -1 O objeto referenciado pelo dicionário não existe.
- Código -2 O descritor do objeto referenciado pelo dicionário não existe.
- Código -3 O objeto pai de folha referenciado pelo dicionário de folhas não existe.
- Código -4 O descritor de objeto do objeto pai de folha referenciado pelo dicionário de folhas não existe.
- Código >=0 O dicionário da classe de objetos %cod contém referência a objeto desconhecido.

Ido %n1 %s Tipo %n2 -> tipo errado (código %cod)

O dicionário referencia o objeto %n1 cujo indicador de classe é inconsistente com a classe do dicionário %n2. A referência é eliminada do presente dicionário. Posteriormente, o objeto em erro será redicionarizado no dicionário de classe correto. A mensagem de redicionarização identifica o objeto pelo seu número.

- Código -1 A classe do objeto referenciado pelo dicionário não corresponde à classe de objetos do dicionário.
- Código -2 A classe do objeto referenciado pelo dicionário não corresponde à classe de objetos folhas.

Ido %n1 %s Tipo %n2 -> dicionário fora de ordem

O objeto %n1 %s contido no dicionário da classe de objetos %n2 está fora de ordem alfabética com relação ao objeto anterior neste dicionário. As referências desordenadas são eliminadas do presente dicionário. Posteriormente, os objetos erro serão redicionarizados na ordem correta. A mensagem de redicionarização identifica os objetos pelo seu número.

Lista %n %s -> erro lista dicionário (código %cod)

Dicionários são armazenados em listas. Estas devem satisfazer uma série de condições para que possam ser consideradas íntegras. O valor %n é o identificador da lista do dicionário. O valor %s é o nome simbólico da classe de objetos do dicionário em questão. Pode não ser o nome de um dicionário quando se trata de um dicionário desconhecido (código -8) ou quando se trata de um objeto erroneamente dicionarizado (código >= 0). Neste último caso, o nome que aparece na mensagem corresponde ao objeto faltoso.

- Código -1 A lista do dicionário principia com uma página incorreta.
- Código -3 O tipo da lista de não é de lista de dicionário.
- Código -4 A origem do dicionário não corresponde à origem de uma lista.
- Código -5 A lista do dicionário não pertence à lista de listas.
- Código -6 Código -7 Código -8 O encadeamento da lista de dicionário não está correto (-6 antecessor, -7 sucessor, -8 tipo de Páginas).
- Código -9 Foi encontrado um dicionário para um objeto pertencente a uma classe de objetos desconhecida ou não dicionarizada.
- Código -10 O dicionário foi esvaziado ao colecionar lixo.
- Código >=0 Foi encontrado um dicionário de uma classe de objetos desconhecida ou não dicionarizada. O código é o número identificador da classe de objetos. O nome simbólico exibido é "???".
- Observação** Os erros código -9 e >= 0 podem ocorrer em bases que foram criadas na versão 3.3 ou anterior de *Talisman*. Nestas versões os objetos não dicionarizáveis eram mantidos em um dicionário especial. Na versão 4 em diante, objetos não dicionarizáveis, efetivamente não possuem dicionários. A *cardinalidade* em diagramas de **Entidades e Relacionamentos** é um exemplo de uma classe de objetos não dicionarizável. Objetos não dicionarizáveis aparecem exclusivamente como instâncias em diagramas.

Ido %n1 %s Tipo %n2 -> inserido em dicionário

Foi encontrado um objeto dicionarizável e que não se encontra dicionarizado no dicionário correspondente à sua classe. O objeto é inserido no dicionário. Cabe observar que o dicionário em questão pode ser o de diagramas (Tipo = 5). %n1 identifica o objeto em erro, %s é o nome deste objeto, %n2 é o identificador de classe do objeto.

Ido %n1 %s Tipo %n2 -> nome duplicado eliminado

O objeto dicionarizável %n1 %s possui nome igual ao de outro objeto que já se encontra dicionarizado. Os nomes de objeto devem ser únicos em cada dicionário de classe. %n2 é o identificador de classe do objeto.

14.5.5 Erros em objetos

Ido %n %s -> tamanho de descritor ajustado

O descritor do objeto %n %s possui tamanho inconsistente.

Ido %n1 %s Tipo %n2 -> tipo errado (código %cod)

O identificador de classe do objeto corrente está em erro. O objeto será totalmente eliminado.

- Código -1 A classe do objeto é ilegal.
- Código -2 A classe do objeto é desconhecida pela base de conhecimento.

Ido %n %s -> ajustou contador de objetos

O gerador de identificações únicas de objetos está inconsistente com o objeto corrente.

Ido %n1 %n2 %n3 %n4 -> descritor de objeto eliminado

O objeto está sendo totalmente eliminado. Os identificadores %n_i são as chaves do descritor de objeto sendo eliminado.

Ido %n %s %n2 %n3 %n4 -> atributo eliminado

Os atributos de um objeto estão sendo eliminados. Os identificadores %n_i são as chaves do atributo sendo eliminado. O nome simbólico do objeto pode não ser conhecido.

Ido %n %s -> nome de objeto eliminado

O atributo nome do objeto %n %s está sendo eliminado. Isto terá como consequência que o objeto será integralmente eliminado. O nome de um objeto sempre será eliminado antes que outros atributos do objeto sejam eliminados. A identificação dos demais atributos se fará pelo número identificador do objeto (Ido).

Ido %n1 Ali s %n2 -> eliminado

O atributo aliás índice %n2 do objeto %n1 está sendo eliminado. O valor do aliás estará na linha a seguir a esta mensagem. Ver mensagem “---%s”.

Ido %n1 Texto %n2 -> eliminado

O atributo ali s índice %n2 do objeto %n1 está sendo eliminado. As linhas que formam este texto serão exibidas logo a seguir desta mensagem. Ver mensagem “---%s”.

---%s

Exibe o valor de um aliás, ou de uma linha de texto sendo eliminada. Caso o LOG de recuperação tenha sido dirigido para um arquivo, aliases e textos podem ser copiados para dentro de um objeto, utilizando o editor de arquivos sequenciais (comando [!Q]) e comandos de marcar e copiar blocos ([F9 ENTER] [F9 INS]), disponíveis em Talisman.

Ido %n1 %s Texto %n2 -> lista texto errada (código %cod)

O objeto %n1 %s referencia um texto longo de índice %n2 e que apresenta algum problema. Textos podem ser longos ou curtos. Um texto curto cabe na própria página do objeto. Já textos longos são armazenados em listas. Estas listas devem satisfazer diversas condições.

- Código -1 A lista do texto principia com uma página incorreta.
- Código -2 O tipo da lista de não é de lista de texto. O tipo de listas de texto não era controlado nas versões anteriores à versão 4. Esta mensagem indica, portanto, que o identificador de tipo de lista do texto longo foi atualizado.
- Código -4 A origem do texto não corresponde à origem de uma lista.
- Código -5 A lista de texto não pertence à lista de listas.
- Código -6 Código -7 Código -8 O encadeamento da lista de texto não está correto (-6 antecessor, -7 sucessor, -8 tipo de Páginas)
- Código -9 Foi encontrado um erro na lista de texto que requer que o atributo seja eliminado.

Ido %n1 Relação %n2 -> eliminada

O atributo relação índice %n2 do objeto %n1 está sendo eliminada. A tabela TALISMAN.LST fornecida junto com o sistema, permite determinar o nome simbólico da relação sendo eliminada. Os nomes dos objetos relacionados são apresentados logo após esta mensagem.

>>>>>>%n>>>>>>%s

O objeto %s é um dos objetos da relação %n sendo eliminada. Caso o LOG de recuperação tenha sido dirigido para um arquivo, os nomes que formam a relação podem ser copiados para dentro de um objeto, utilizando o editor de arquivos sequenciais (comando [!Q]) e comandos de marcar e copiar blocos. ([F9 ENTER] [F9 INS]) disponíveis em Talisman.

>>>>>>%n1>>>>>>Ido %n2 -> objeto relacionado desconhecido

O objeto %n2 é um dos objetos da relação %n1 sendo eliminada. No entanto, o nome deste objeto não está definido.

Ido %n1 %s Rel %n2:%n3 -> relação desconhecida

O objeto %n1 %s contém uma relação %n3 que não é conhecida pela base de conhecimento com respeito ao tipo %n2. Ao migrar para versão 4, algumas relações tiveram que receber nova identificação em virtude de ambigüidades existentes na versão 3. Este problema afeta as linguagens de representação **Estado e Transição** e **Análise da Organização**.

Ido %n1 %s Rel %n2 -> relação ajustada (código %cod)

As diversas relações possuem organizações próprias. Os parâmetros de armazenagem da relação %n2 do objeto %n1 %s não estão consistentes.

- Código -1 O tamanho da relação não é um múltiplo inteiro do descritor de relacionamento.
- Código -2 O tamanho da relação é menor do que a área alocada para a relação.
- Código -3 O tamanho da relação é maior do que a área alocada para a relação.
- Código -4 O indicador de relacionamento corrente não é consistente com o tamanho da relação.

Ido %n1 %s Rel %n2 Destá%n3 -> objeto não existe

O objeto %n1 %s possui um atributo relação %n2. Este relaciona o objeto %n3 que não existe.

Ido %n1 %s Rel %n2 Destá%n3 -> objeto tipo errado

O objeto %n1 %s possui um atributo relação %n2. Este relaciona o objeto %n3 cujo tipo não corresponde ao tipo destino da relação.

Ido %n1 %s Rel %n2 Destá%n3 -> inversa não existe

O objeto %n1 %s possui um atributo relação %n2. Este relaciona o objeto %n3 que, por sua vez, não possui o atributo relação inversa.

Ido %n1 %s Rel %n2 Destá%n3 -> inversa não contém %n1

O objeto %n1 %s possui um atributo relação %n2. Este relaciona o objeto %n3 que, por sua vez, não relaciona o objeto %n1.

Ido %n1 %s Rel %n2 -> elimina relação vazia

O objeto %n1 %s possui um atributo relação %n2 que é ou se tornou vazio.

14.5.6 Erros em diagramas

Ido %n1 Repr %n2 -> estrutura de folhas errada (código %cod)

Cada representação gráfica %n2 possui um objeto *raiz da estrutura de folhas* e uma *folha corrente*.

- Código -1 O objeto %n1 não é uma origem válida para a estrutura de folhas da representação %n2. Este erro pode ocorrer ao migrar para a versão 4, caso se tenha desenvolvido diagramas na linguagem Estados e Transições.
- Código -2 O objeto %n1 não é uma folha corrente válida para a estrutura de folhas da representação %n2. O valor é ajustado.

Folha %n %s -> errada, eliminada (código %cod)

- Código -1 A folha %n não possui objeto pai.
- Código -2 A folha %n %s não corresponde a uma representação gráfica conhecida. Este erro pode ocorrer ao migrar da versão 2 para a versão 4, uma vez que a representação Projeto Modular deixou de ser gráfica. Pode ocorrer, ainda, ao migrar da versão 3 para a versão 4 caso tenham sido desenvolvidos diagramas de Estados e Transições.
- Código -3 O tipo de rótulo da folha %n %s não é conhecido na base de conhecimento.
- Código -4 O objeto pai da folha não existe.
- Código -5 O tipo do objeto pai da folha não é consistente com a folha.
- Código -6 O tipo do objeto pai da folha não admite decomposição.
- Código -7 A folha referenciada no objeto pai não corresponde à folha corrente.
- Código -8 A folha origem da representação não corresponde à folha origem de uma estrutura de folhas.
- Código -9 O tipo da folha origem não é compatível com o tipo do objeto origem de estrutura correspondente à folha.
- Código -10 O tipo do objeto referenciado como sendo raiz da estrutura de folha não vale.

- Código -11 O objeto raiz da estrutura de folhas não é conhecido.
- Código -13 A folha possui mais do que uma instância pai.
- Código -14 A instância pai da folha não existe.
- Código -15 A instância pai da folha é de um tipo que não admite decomposição.
- Código -16 O objeto raiz da estrutura de objetos não é conhecido.
- Código -17 O objeto raiz da estrutura de objetos corresponde a uma folha que não é origem desta estrutura, uma vez que possui instância pai.
- Código -18 A lista de instâncias da folha foi eliminada ao coletar lixo.

Folha %n %s -> vazia, eliminada (código %cod)

- Código -1 A folha %n %s contém nenhuma instância de objeto.
- Código -2 A recuperação da folha %n %s tornou-a vazia.

Folha %n1 %s -> excluiu atributo %n2 %n3 %n4

O atributo de chave %n2, %n3, %n4 da folha %n1 %s foi eliminado. Em geral este erro é consequência de outros erros.

Folha %n1 %s : %n2 -> ligação errada (código %cod)

Ligações relacionam duas instâncias de objeto. Todas as ligações possuem descritores e possuem *ligação gráfica*. A ligação gráfica é uma sequência de pontos de passagem. Ligações podem, ainda, relacionar um rótulo. Este é uma instância de objeto.

- Código -1 A instância de ligação %n2 não possui ligação gráfica (lista de pontos).
- Código -2 A instância de objeto inicial da instância de ligação %n2 não existe.
- Código -3 A instância de objeto final da instância de ligação %n2 não existe.
- Código -4 A instância de ligação referenciada por uma instância de objeto não possui ligação gráfica.
- Código -5 A instância de ligação referenciada por uma instância de objeto não existe.
- Código -6 O tipo da instância de ligação referenciada por uma instância de objeto é inconsistente, ou, então, a instância de ligação não inclui referência à instância de objeto.
- Código -7 A lista de instâncias de ligação da instância de objeto %s está, ou tornou-se, vazia.

Folha %n %s : %n2 -> rótulo errado (código %cod)

- Código -1 O tipo de especificação da instância de rótulo é incompatível.
- Código -2 O tipo da instância de rótulo é incompatível com a representação.
- Código -3 O tipo da instância de rótulo está errado
- Código -4 O objeto de especificação da instância de rótulo não existe.
- Código -5 O objeto de especificação da instância de rótulo possui tipo incompatível.
- Código -6 A instância de ligação correspondente à instância de rótulo não existe.
- Código -7 O tipo da instância de ligação correspondente à instância de rótulo é incompatível.

Folha %n %s : %n2 -> imagem errada (código %cod)

Imagens são relacionamentos entre folhas. Na versão 4 imagens somente existem em diagramas de fluxo de dados. Imagens são agrupadas em uma relação de imagens. Esta relação estabelece o vínculo entre a imagem numa folha e a ligação na folha pai. Imagens possuem ainda uma relação de instâncias de ligações imagem e uma relação de instâncias de objetos imagem.

- Código -1 A instância de ligação não possui, ou não pode possuir, imagem em folha descendente na estrutura.
- Código -2 A imagem da instância de ligação não referencia a instância de ligação.
- Código -3 A imagem da instância de ligação não foi encontrada na folha descendente

Código -4 A lista de instâncias imagem de uma instância de ligação imagem não existe.

Código -5 A lista de instâncias imagem não referencia a correspondente instância de ligação imagem.

Folha %n1 %s : %n2 %n3 -> relação imagem errada (código %cod)

Código -1 Existe relação de imagens em folha que não possui instância pai.

Código -2 A %n3-ésima imagem da relação de imagens não possui descritor de instância de imagem.

Código -3 A %n3-ésima imagem da relação de imagens não possui instância de ligação da qual é imagem.

Código -4 A %n3-ésima imagem da relação de imagens não é referenciada na instância de ligação da qual é imagem.

Código -5 A relação de imagens da folha é, ou tornou-se, vazio.

Código -6 Não foi encontrada relação de imagens correspondente a uma instância de imagem.

Código -7 A relação de imagens não contém a referência à imagem correspondente à instância de imagens.

Código -8 A instância de objeto ou de ligação correspondente à instância de imagem não existe.

Código -9 A instância de objeto ou de ligação não corresponde à instância de imagem.

Código -10 A relação de instâncias de objetos ou de ligações imagem é, ou tornou-se, vazia.

Código -11 A instância de objeto ou de ligação não corresponde a descritor de imagem.

Folha %n : %n2 -> ligação sem instância (código %cod)

Código -1 A ligação gráfica não possui instância de ligação.

Código -2 A ligação gráfica corresponde a uma instância de ligação com tipo inconsistente.

Folha %n %s : %n2 -> instância eliminada (código %cod)

Código -1 O descritor de instância de objeto não existe.

Código -2 O descritor de instância de objeto tem tipo inconsistente com a instância de objeto relacionada.

Código -3 O descritor de instância de objeto tem tipo desconhecido pela base de conhecimento.

Código -4 A instância de objeto referenciada no objeto especificação não existe.

Código -5 A lista de instâncias referenciada no objeto especificação não existe.

Código -6 O objeto especificação correspondente à lista de instâncias de objeto tem tipo inconsistente.

Código -7 O objeto especificação correspondente à lista de instâncias de objeto não existe.

Código -8 A lista de instâncias de objeto imagem correspondente à instância de objeto não existe.

Código -9 A lista de instâncias de objeto imagem correspondente à instância de objeto não referencia esta instância de objeto.

Código -10 O descritor de imagem correspondente à instância de objeto não existe.

Código -11 O descritor de imagem correspondente à instância de objeto é, ou se tornou, vazio.

Código -12 O tipo da instância de objeto não é conhecido pela base de conhecimento.

Código -13 A instância de objeto não existe na lista de instâncias da folha.

Folha %n1 %s lista %n2 -> lista instância errada (código %cod)

Esta é uma coletânea de erros de formação de listas de instâncias de objetos contidos na folha %n1 %s. O valor %n2 é o identificador da lista de instâncias na folha

Código -1 A lista de instâncias principia com uma Página incorreta.

Código -3 O tipo da lista de não é de lista de instâncias.

Código -4 A origem da lista de instâncias não corresponde à origem de uma lista.

Código -5 A lista de instâncias não pertence à lista de listas.

Código -6 Código -7 Código -8 O encadeamento da lista de instâncias não está correto (-6 antecessor, -7 sucessor, -8 tipo das Páginas).

Ido %n %s -> folha filho errada (código %cod)

- Código -1 O objeto não admite folha filho.
- Código -2 O objeto não admite decomposição.
- Código -3 A folha filho do objeto não existe.
- Código -4 O tipo da folha filho do objeto não é folha.

Ido %n %s -> relação instância errada (código %cod)

- Código -1 O objeto possui instância mas não é instanciável.
- Código -2 A folha da instância do objeto não existe.
- Código -3 O tipo da folha da instância do objeto não é folha.
- Código -4 A folha da instância do objeto não contém a instância do objeto.
- Código -5 A relação de instâncias do objeto foi alterada.
- Código -6 A relação de instâncias do objeto está, ou se tornou, vazia.

14.5.7 Erros de eliminação de lixo

Lista %n -> é lixo, eliminada (código %cod)

- Código -1 A origem da lista a marcar não é Página válida.
- Código -2 A origem da lista a marcar não é origem de lista.
- Código -3 A lista a marcar já foi marcada. Trata-se de uma referência múltipla para a mesma lista.
- Código -4 A lista a marcar não estava na lista de listas. Foi inserida na lista de listas.
- Código -5 A lista a marcar possui Páginas de diferentes tipos.
- Código -6 Uma ou mais Páginas da lista sendo marcada já foi marcada anteriormente. Trata-se de referência múltipla a uma mesma Página.
- Código -7 A lista de listas foi identificada como lixo. Este erro tem com consequência o cancelamento da execução, uma vez que a base de software passou a não valer mais.
- Código -8 Uma das listas Btree foi identificada como lixo. Este erro tem com consequência o cancelamento da execução, uma vez que a base de software passou a não valer mais.
- Código -9 A lista identificada não está vinculada a qualquer atributo na base de software.

Fim do documento