

EDITOR GRÁFICO DE GRAFCET

Couto, Carlos Alberto *
Fonseca, Jaime **

* Departamento de Electrónica Industrial,
Universidade do Minho,
4719 Braga CODEX, PORTUGAL,
Tel. +351 53 615010, Fax: +351 53 615046,
e-mail: dei@uminho.pt

** Departamento de Electrónica Industrial,
Universidade do Minho,
4719 Braga CODEX, PORTUGAL,
Tel. +351 53 615010, Fax: +351 53 615046,
e-mail: dei@uminho.pt

RESUMO

Esta comunicação descreve a implementação de um editor gráfico para a linguagem de programação de autómatos programáveis GRAFCET funcionando em ambiente Windows. Tratando-se de uma linguagem essencialmente gráfica a ferramenta desenvolvida facilita extraordinariamente a programação na medida em que possibilita a visualização dos programas.

1. INTRODUÇÃO

O GRAFCET - GRAFo de Comando Etapa-Transição - inspirado nos diagramas de estado foi no início uma ferramenta utilizada para a especificação de problemas de automação. Mais recentemente tem sido feita a apologia da sua utilização como linguagem de alto nível na programação de PLCs (Controladores Lógicos Programáveis).

Embora já oferecido por um considerável grupo de fabricantes, especialmente os de influência francesa, ainda está longe de ser uma linguagem universal, já que a maioria dos PLCs admitem apenas como linguagem de programação as funções booleanas e/ou a sua representação gráfica (diagramas de escada ou "ladder Diagrams").

O Departamento de Electrónica Industrial tem vindo a investir consideráveis esforços no estudo e desenvolvimento de soluções utilizando o GRAFCET, tendo nomeadamente desenvolvido um controlador baseado no microcontrolador INTEL 8051, programável naquela linguagem, e mais recentemente, a pedido de um fabricante nacional - Patrónica - cuja a arquitectura do PLC é igualmente baseada no INTEL 8051, desenvolveu um interpretador GRAFCET. Em ambos casos o Grafo é traduzido "à mão" para instruções alfanuméricas editadas em qualquer editor ASCII como o editores do SK, do BorlandC ou o Wordstar. Esta solução deixa a cargo do programador toda a visualização gráfica do programa e sua respectiva tradução em código alfanumérico, tornando difícil a detecção de erros no grafo ou na sua tradução, sempre que este apresenta uma determinada complexidade.

Com vista a diminuir os problemas anteriormente citados implementou-se, um editor gráfico de GRAFCET, que permite a programação através do desenho do grafo de uma forma simples. A tradução para código alfanumérico realiza-se nos momentos que precedem a sua transferência para o PLC.

2. ESPECIFICAÇÃO E CONCEPÇÃO

Pretendia-se um editor amigável tirando partido dos modernos interfaces gráficos. A escolha pelo ambiente Windows foi natural pela sua universalidade e conseqüente portabilidade, além das facilidades de interacção com a vasta gama de programas já existentes para aquele ambiente, permitindo nomeadamente o seu funcionamento em modo concorrente.

Assim o trabalho iniciou-se com o estudo do ambiente Windows e das respectivas ferramentas de desenvolvimento.

Tendo em vista um interface fácil e agradável com o utilizador optou-se pela utilização de Menus Desdobrados (Menus PopDown), como forma de acesso a Caixas de Diálogo (Dialog Box) através das quais é feita a entrada de parâmetros. Para as instruções de edição do grafo, tendo em vista operações mais rápidas e intuitivas, optou-se por um acesso complementar através de símbolos gráficos contidos numa Barra de Ferramentas (Tool Bar). Para alguns dos outros comandos/opções decidiu-se pelo acesso directo por teclas específicas em complemento ao rato.

Para eventual apoio do utilizador na geração do grafo implementou-se um esquema de ajuda, geral e no contexto, disponível no modo de funcionamento normal do Windows.

Na organização dos menus decidiu-se seguir a estrutura comum à generalidade das aplicações em Windows, com as opções de gestão de ficheiros na primeira coluna, edição na segunda, configuração e "diálogo" com o PLC na terceira e ajuda na quarta. É essa estrutura que a seguir se descreve:

- **opções de gestão de ficheiros:** permitem várias operações com ficheiros tais como abrir ficheiro, novo, guardar ficheiro, guardar como;
- **opções de edição:** permitem a construção do grafo e activam as Caixas de Diálogo que requerem a introdução dos parâmetros necessários à caracterização das acções e condições respeitantes a cada etapa-transição.
- **opções de configuração e diálogo com o PLC:** permitem configurar a transmissão, tais como o porto de transmissão e velocidade de transmissão, dar "ordens" ao PLC e transferir o código alfanumérico correspondente ao grafo para o PLC;
- **opções de ajuda:** permitem aceder às facilidades de ajuda. Uma delas utilizando o menu disponível na janela do ajuda, a outra, ajuda no contexto, é obtida pelas teclas Shift-F1 com a opção/comando seleccionada.

Em anexo 1 apresenta-se o conjunto de instruções Grafcet, em termos gráficos e alfanuméricos, que serão contempladas.

Para validação dos parâmetros das instruções Grafcet optou-se por nas Caixas de Diálogo além de se sugerir a gama das entradas correctas, implementar o filtro que apenas permite o fecho de Diálogo após introdução dos parâmetros correctos. Deste modo garante-se a validação sintáctica instrução a instrução. Esta estratégia, de implementação mais simples, foi considerada perfeitamente aceitável dado o pequeno conjunto de instruções, em alternativa à verificação sintáctica imediatamente antes da compilação do grafo para o código de execução.

3. FERRAMENTAS UTILIZADAS

Juntamente com o estudo do ambiente Windows foi estudado o Borland C++ 3.0 e o utilitário WorkShop.

O Microsoft Windows é uma "extensão" gráfica ao sistema operativo MS-DOS, desenvolvida no sentido de aumentar as características deste, permitindo tirar partido das potencialidades a nível de "hardware" que actualmente se dispõe. Relativamente ao MS-DOS apresenta as seguintes vantagens :

- fornece um conjunto "standard" de objectos de interface tais como menus, janelas e teclas de controle comuns a todos os programas Windows, introduzindo assim o conceito de recurso;
- filas de espera para escalonamento de mensagens vindas de diferentes aplicações;
- suporte gráfico independente do dispositivo;
- capacidade de multitarefa;
- permite troca de dados entre aplicações;
- superior gestão de memória, introduzindo o conceito de "Dynamic Link Libraries " (DLL). As DLL são bibliotecas de subrotinas "linked" no momento em que se carrega ou durante a execução do programa, em vez de ser na compilação;

A utilização do Borland C++ 3.0 prendeu-se com a disponibilidade de primitivas extremamente poderosas, que facilitam o desenvolvimento de aplicações para o Windows com relativa facilidade. É de salientar a primitiva "WinHelp", que possibilita a chamada de um ficheiro de "Help" previamente compilado pelo compilador de Help HC. A estrutura deste é utilizada em todas as aplicações Windows sendo a única diferença o seu conteúdo. A edição do Help pode ser feita em qualquer editor de texto que corra em Windows (Ex : WinWord).

O WorkShop é um utilitário que permite a construção e modificação de recursos usados em aplicações que corram sobre Windows. Os recursos são entidades que ficam ligadas ao programa executável, sendo lidos em "run-time" quando necessário, podendo posteriormente libertar-se o espaço de memória por eles ocupado. Este utilitário inclui editores para :

- "accelerator tables", que são tabelas que associam combinações de teclas a opções de menus, tornando mais rápida a selecção da opção;
- "bitmaps", permitindo desenhar imagens, símbolos sugestivos das opções/comando;

- "cursor", que é um tipo especial de bitmap limitado a uma dimensão específica de 32x32 pontos. O cursor é utilizado para seleccionar elementos gráficos no écran, ou para localizar posições para entradas de inserção;
- "dialog box", ou seja, janela com diversos objectos (botões, listas, texto, etc) cujo objectivo é fornecer um modo de interface com o utilizador;
- "string table", que são tabelas de strings constantes, que permite restringir o uso de memória, podendo-se fazer a leitura de uma string para memória em run-time e posteriormente libertá-la;
- "icons", que são pequenos bitmaps que se associam às aplicações para uma fácil identificação;
- "menus", permitindo a descrição de menus PopDown.

O WorkShop cria um ficheiro de recursos com extensão .RES, sendo a sua ligação ao ficheiro executável como mostra a figura a seguir.

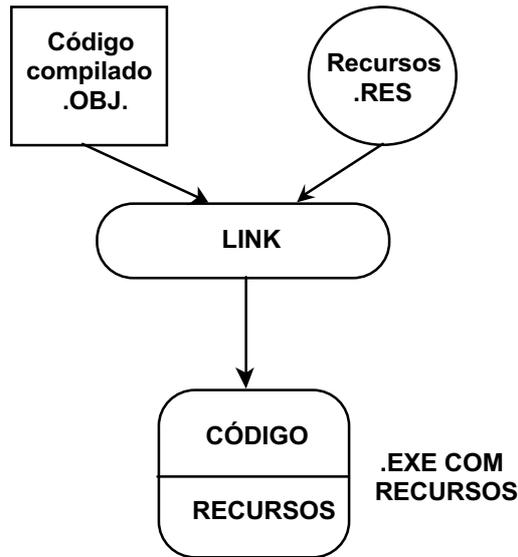


Fig.1- Construção de uma aplicação Windows utilizando o utilitário WorkShop

4. ALGORITMO BASE

Todos os programas que correm sobre o Windows são do tipo "event-driven" ou "message-driven", que permitem ao utilizador estabelecer a sequenciação de passos requeridos para realizar uma tarefa, ao contrário dos programas tradicionais do tipo "sequence-driven" ou "procedure-driven" onde a sequenciação é fixa e definida pelo próprio programa.

Esta mudança do tipo de programação "sequence-driven" para "event-driven", requer uma nova maneira de raciocinar e exige inicialmente um esforço suplementar no que diz respeito à especificação e estruturação do programa. Por outro lado depois de algum tempo, estes pontos deixam de ser problemáticos, trazendo como vantagem a possibilidade de reutilizar o código produzido desta forma.

Na especificação do algoritmo foi utilizado um Diagrama de Fluxo de Dados (DFD) como se pode ver na figura 2 a seguir.

5. IMPLEMENTAÇÃO

O editor gráfico é um programa do tipo "event-driven" ou "message-driven", em que toda a execução é baseada na gestão de **mensagens**. Estas não são mais que entradas de uma aplicação, e representam acontecimentos a que as aplicações necessitam responder. Uma mensagem contém sempre um identificador e diversos parâmetros. O conteúdo destes parâmetros varia consoante o tipo de mensagem.

O desenvolvimento de qualquer aplicação para correr sobre Windows necessita de duas funções básicas :

- função WinMain;

- procedimento Window.

A função WinMain é o ponto de entrada para qualquer aplicação Windows. Visto que nenhuma aplicação pode correr sem ela. A função WinMain no editor gráfico é responsável pelas seguintes tarefas:

- efectua as inicializações necessárias para registar duas classes "janela", sendo uma delas responsável pela janela principal onde se encontram todos os comandos do editor, e a outra onde é editado o grafo e o código alfanumérico. Realiza ainda a inicialização dos bitmaps que facilitam a construção do grafo;
- contém um ciclo que permite o processamento de mensagens a partir da fila de mensagens da aplicação;
- termina a aplicação quando surge a mensagem WM_QUIT.

As classes janela têm um conjunto de atributos que permitem definir as janelas respectivas. Antes de qualquer aplicação poder criar e usar uma janela, terá que ser criada e registada para essa janela uma classe. O Windows ao registar uma classe copia os seus atributos para uma área de memória, e utiliza estes atributos armazenados internamente quando a aplicação se refere à classe pelo nome.

Quanto ao ciclo que processa as mensagens, o funcionamento é o seguinte: o Windows não emite entradas directamente para uma aplicação; em vez disso coloca todas as entradas do rato e teclado dentro da fila de mensagens da aplicação; esta vai buscar a mensagem à sua fila e encaminha-a para o procedimento windows apropriado para processamento.

A função do procedimento Windows é processar todas as mensagens emitidas para todas as janelas da respectiva classe. O Windows emite mensagens para o procedimento do tipo "windows" quando recebe uma entrada a partir do utilizador na sua janela ou quando é necessário ao procedimento activar alguma acção na sua janela, como por exemplo pintar uma determinada zona. Um procedimento Windows pode receber o seguinte tipo de mensagens:

- mensagens de entrada a partir do teclado ou rato;
- pedidos de informação, como por exemplo o título da janela;
- mensagens que dão oportunidade ao procedimento windows de modificar o sistema inicial respondendo a determinadas acções, como por exemplo ajustar os menus antes de serem visualizados;
- pedido para enviar uma acção para a sua janela ou área cliente, como por exemplo abrir uma janela numa determinada área;
- informações acerca do seu estado nas relações com outras janelas, como por exemplo tornar activa uma janela;

Muitas das mensagens que um procedimento windows recebe vêm do Windows, mas pode igualmente receber mensagens a partir de outras janelas, inclusive janelas próprias. Estas últimas mensagens não passam pela fila de mensagens da aplicação, indo directamente ao procedimento windows (utiliza-se uma primitiva interna para enviar mensagens).

Para além destes dois procedimentos básicos necessários a qualquer programa desenvolvido para correr sob Windows, no editor gráfico foram definidos um conjunto de recursos que permitem um interface muito agradável com o utilizador. Esses recursos foram desenvolvidos no utilitário WorkShop. A eles é atribuído um nome ou número único (dentro dum mesmo tipo) que funciona como identificador. Estes recursos descrevem características visuais de elementos do programa. O Windows vê estes recursos como objectos que respondem a determinados tipos de mensagens. Estas mensagens são encaminhadas para estes objectos através de procedimentos windows.

Em suma, toda a implementação do editor gráfico está baseado na gestão de determinados tipos de mensagens, tendo sido codificado em C (Borland C++ 3.0). Todos os recursos utilizados, tais como Menus Desdobrados, Caixas de Diálogo (Dialog Box), Símbolos Gráficos (Bitmaps, Ícons e Cursor) e Tabelas de Strings (String Tables) foram gerados com o utilitário WorkShop, e armazenado originando em ficheiro posteriormente ligado ao código executável.

A figura 3 mostra como o Windows faz o processamento de mensagens nas diferentes aplicações que estão a correr.

6. RESULTADOS E CONCLUSÕES

O editor implementado correspondeu às expectativas:

- a sua interface resultou amigável e fácil;
- permite a visualização do grafo e da sua caracterização;
- Faz uma validação estrutural do grafo e uma validação ao nível da sintaxe do código alfanumérico.

Características testadas por pessoas estranhas ao projecto. No anexo 2 inclui-se a organização dos menus desdobrados e ainda uma ilustração da utilização do editor.

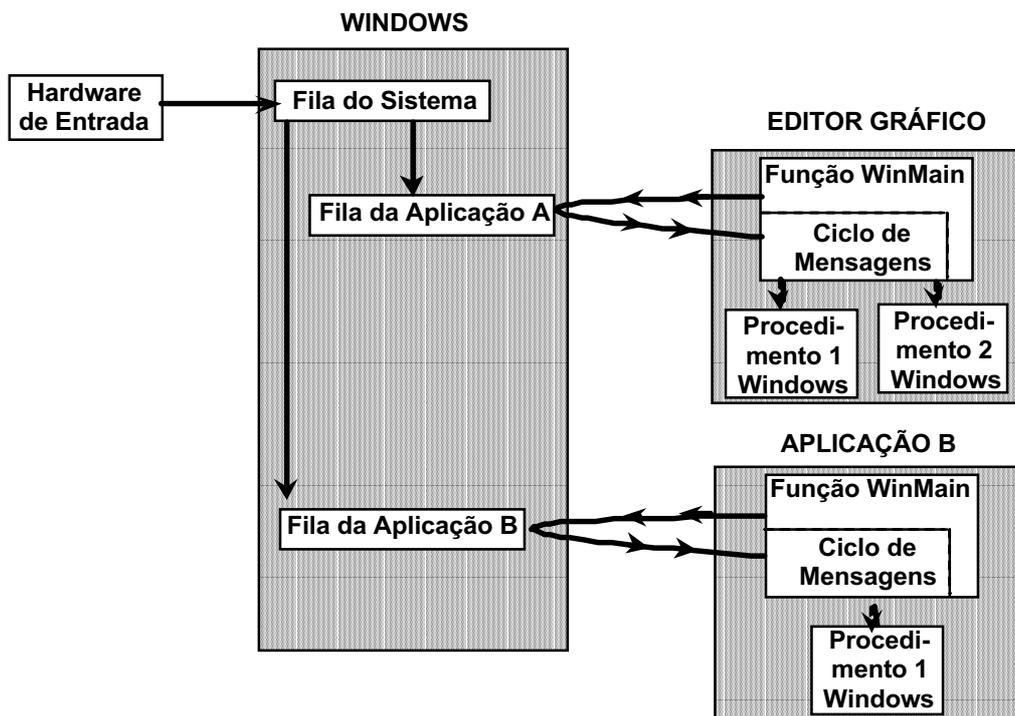


Fig.3- Gestão de Mensagens no Windows

Como continuação deste trabalho estão previstos vários melhoramentos, alguns dos quais se enumeram em seguida.

- Possibilidade de impressão do grafo e respectiva codificação.
- Monitorização da execução do grafo sendo necessário para tanto desenvolver uma aplicação que graças às vantagens do Windows poderá ser executada concorrentemente com outras aplicações. Este programa de monitorização receberá, em tempo real, a evolução das variáveis do PLC permitindo:
 - actualização/visualização das etapas activas em cada momento;
 - visualizar a validação das condições de transição bem como o valor de temporizadores e contadores;

De salientar que esta monitorização facilita a depuração do grafo bem como a avaliação do desempenho de diversas soluções alternativas.

É de referir ainda que as soluções actualmente disponíveis no mercado respeitantes à utilização de Grafset, têm somente como objectivo a sua utilização como linguagem de especificação que posteriormente é "traduzida" para notação booleana ou diagramas de escada. Perante este panorama está em estudo o desenvolvimento de "drives" que possibilitem a utilização de programação em Grafset nos diferentes autómatos do mercado.

ANEXO 1

Neste anexo começamos por apresentar uma listagem das instruções alfanuméricas utilizadas na codificação, e em seguida apresenta-se um grafo que utiliza alguma da codificação apresentada.

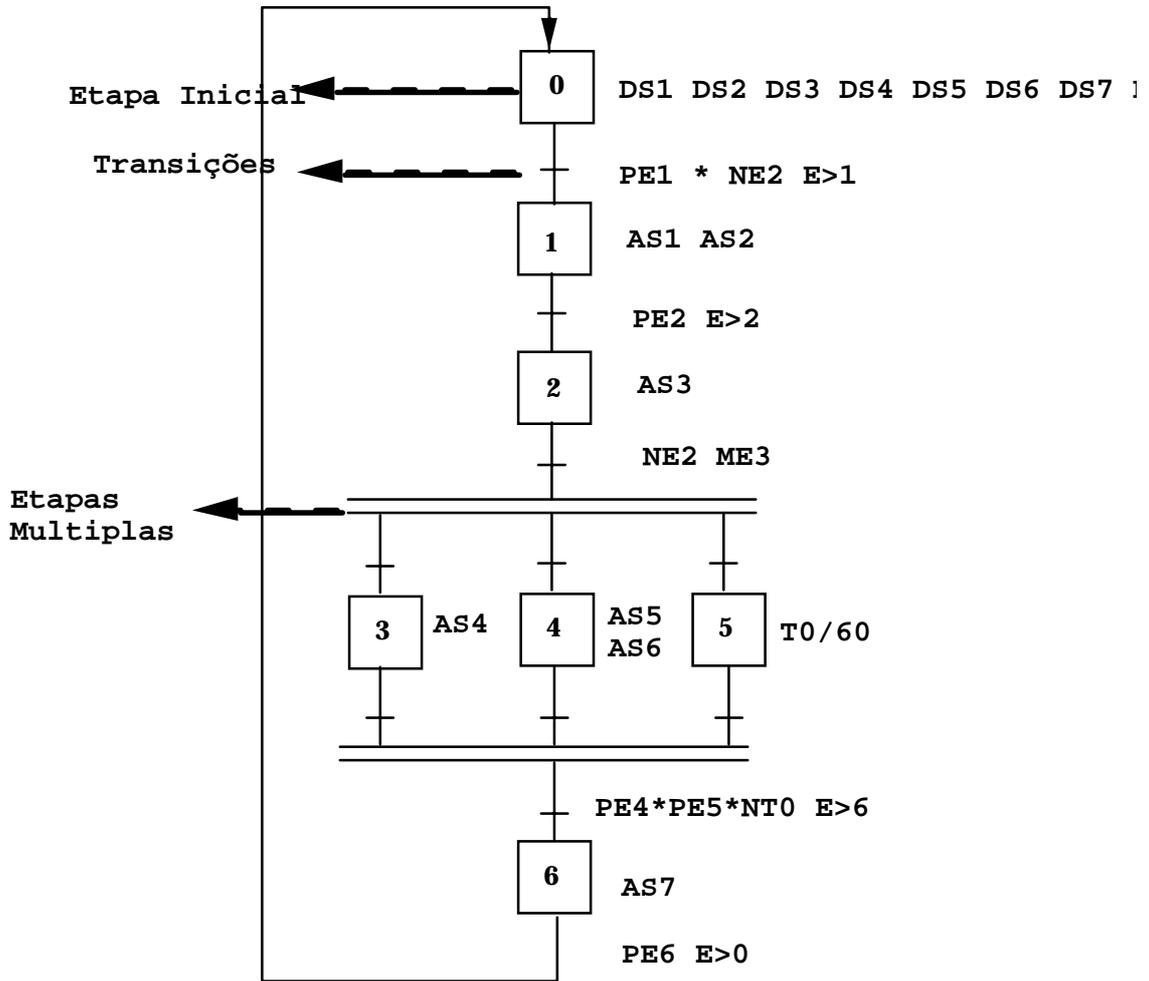
Listagem das instruções alfanuméricas respeitantes às acções:

- C(valor1)/(numero), inicializa o contador especificado com "valor1" com o valor atribuído a "numero";
- T(valor1)/(numero), inicializa o temporizador especificado por "valor1" com o valor atribuído a "numero";
- DS(valor), desactiva a saída especificada por "valor";
- AS(valor), activa a saída especificada por "valor";
- IC(valor), incrementa o contador especificado por "valor" de uma unidade;
- DC(valor), decrementa o contador especificado por "valor" de uma unidade;
- ASTTL(valor), activa a saída TTL especificada por "valor";
- DSTTL(valor), desactiva a saída TTL especificada por "valor";

Listagem das instruções alfanuméricas respeitantes às condições:

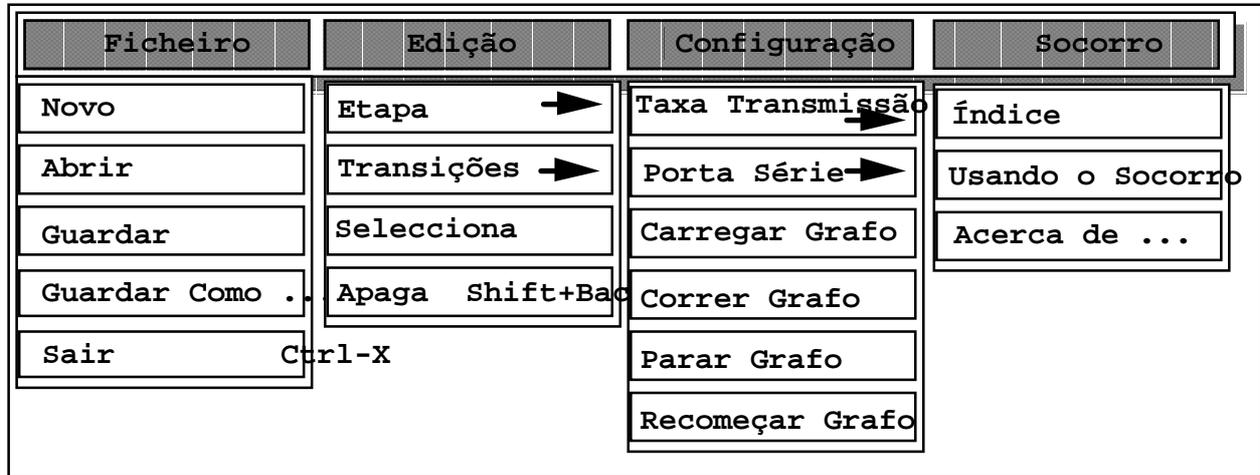
- PE(valor), verifica se a entrada especificada por "valor" está activa;
- NE(valor), verifica se a entrada especificada por "valor" está inactiva;
- NT(valor), verifica se o temporizador especificado por "valor" tem o valor zero;
- NC(valor), verifica se o contador especificado por "valor" tem o valor zero;
- NS(valor), verifica se a saída especificada por "valor" está inactiva;
- PDS(valor), verifica se a saída especificada por "valor" está activa;
- CMPC(valor1)(><=)(numero), verifica se o valor do contador especificado por "valor1" é menor, maior ou igual que o valor atribuído a "numero";
- PETTL(valor), verifica se a entrada TTL especificada por "valor" está activa;
- NETTL(valor), verifica se a entrada TTL especificada por "valor" está inactiva;
- PSTTL(valor), verifica se a saída TTL especificada por "valor" está activa;
- NSTTL(valor), verifica se a saída TTL especificada por "valor" está inactiva;
- *, trata-se do E lógico;
- +, trata-se do OU lógico;
- E>(valor), indica a transição para a etapa especificada em "valor";
- ME(valor), indica o número de etapas activas simultaneamente.

Apresentação do grafo e respectiva codificação:



ANEXO 2

A) A estrutura dos Menus Desdobrados desenvolvida é apresentada abaixo, assinalando-se também os comandos/opções alvo de acesso directo pelo simples toque de teclas.



B) A seguir apresenta-se a sequência de operações a executar para implementar um grafo, tendo em vista ilustrar as potencialidades do editor.

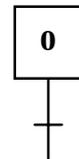
OPERAÇÃO

- 1) Na opção ficheiro selecciona **Novo** (se estiver a editar pela primeira vez o editor abre automaticamente para edição).
- 2) Em seguida pode utilizar a sequência **Menu Edição** → **Etapa** → **Acções** ou mais facilmente seleccionar na barra de ferramentas **Acções**.

RESULTADOS NO ÉCRAN

Apresenta zona de edição limpa.

É desenhado na zona de edição o símbolo de etapa com o correspondente identificador e a respectiva transição, como mostra a figura a seguir.



Quer na barra de ferramentas quer nos Menus desdobrados ficam disponíveis todas as opções que permitem a codificação das acções.

- 3) Pode utilizar a sequência anterior **Menu Edição** → **Etapa** → todas as opções disponíveis respeitantes à **codificação das acções** ou mais facilmente

seleccionar na barra as opções que permitem essa codificação.

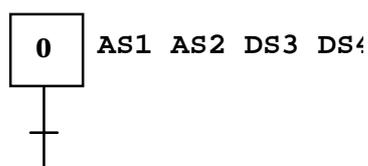
- 4) Após ter codificado todas as acções correspondentes a essa etapa, terá que codificar as condições respeitantes à transição. Para efectuar essa operação pode utilizar a sequência **Menu Edição**—> **Transições**—> **Condições** ou mais facilmente seleccionar na barra de ferramentas **Condições**.

- 5) Pode utilizar a sequência anterior **Menu Edição**--> **Transições**—> todas as opções disponíveis **respeitantes à codificação** das condições ou mais facilmente seleccionar na barra as opções que permitem essa codificação.

- 6) Após seleccionar todas as condições terá que seleccionar uma das seguintes duas opções (a selecção é feita como foi enunciado no ponto anterior): transita para a etapa ... (E>(valor)) ou etapas múltiplas (ME(valor)).

Surgem caixas de diálogo que permitem a introdução de parâmetros. É de salientar que se os parâmetros introduzidos não estiverem correctos, é activada uma nova caixa de diálogo que indica o erro e qual deverá ser a codificação correcta.

Desactiva todas as opções respeitantes à codificação das acções e activa as opções respeitantes à codificação das condições. É editado o código correspondente às acções anteriores como pode ser visto na figura a seguir.



São activadas caixas de diálogo idênticas às anteriores.

É editada o código correspondente às condições anteriores como pode ser visto na figura a seguir.

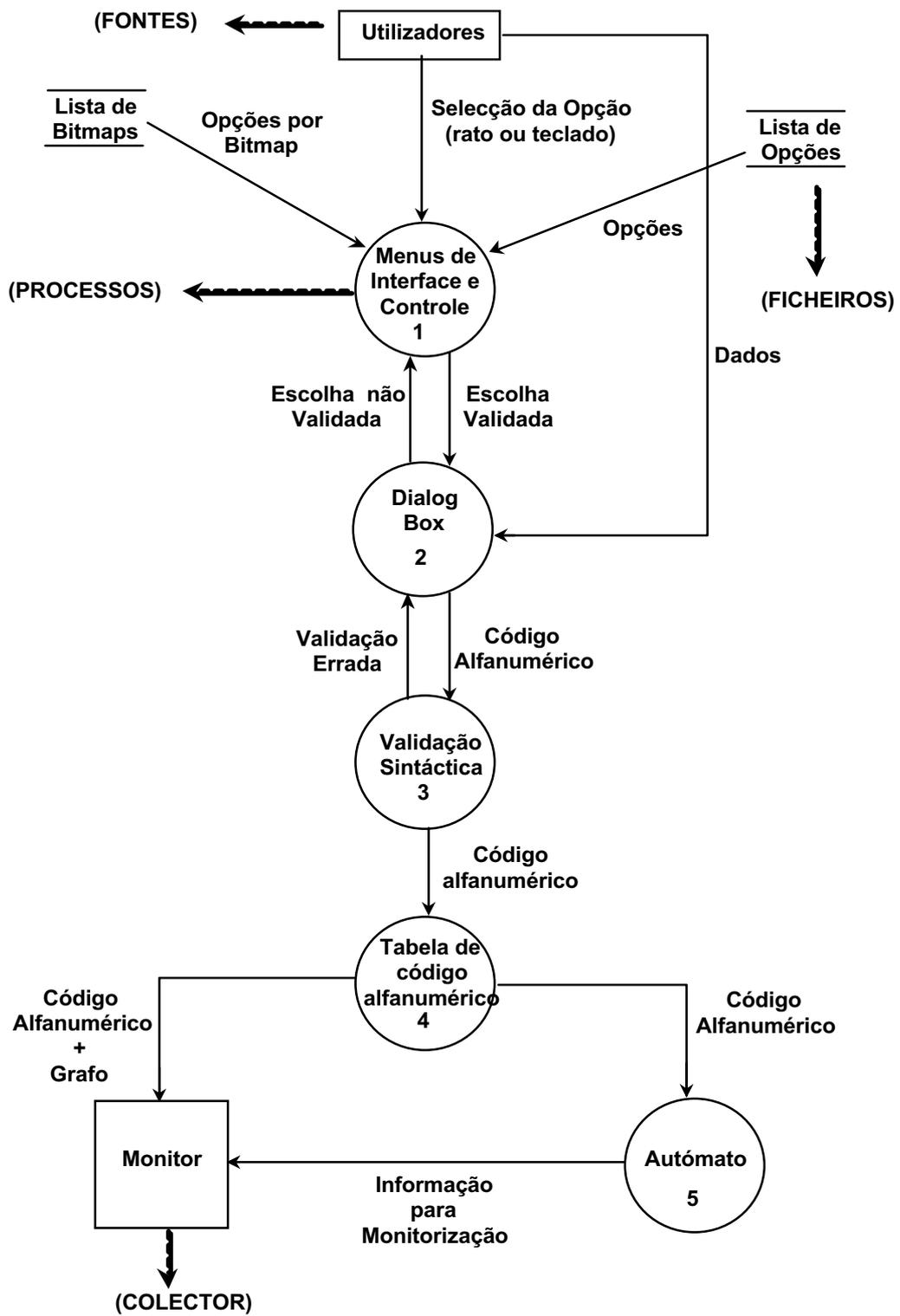


Fig. 2- Algoritmo Base