SOFTWARE BASICO PARA CRIAÇÃO DE DIAGRAMADORES COM UTILITARIOS INTEGRADOS

João Sérgio dos Santos Assis
José Antonio dos Santos Borges

Núcleo de Computação Eletrônica da
Universidade Federal do Rio de Janeiro
Caixa Postal 2324
20.001 - Rio de Janeiro, RJ

1. INTRODUÇÃO

Um dos campos em que existe mais demanda de gráficos no mundo é a produção interativa de diagramas. Um diagrama, falando de forma simples, é um desenho relativamente pequeno, que cabe numa folha de listagem, e que contém uma série de figurinhas (caixinhas, retângulos, bolinhas, bonequinhos, etc.), textos (associados ou não a esses elementos) e elementos de união (tais como traços, setas, etc.).

Os diagramas são utilizados nos campos mais variados, como engenharia (diagramas elétricos e eletrônicos), administração (demonstrativos de vendas, projeções), computação (diagramas de fluxos de dados, diagrama de operação de sistemas, fluxogramas), e tem como principal função descrever idéias de forma clara e organizada.

A utilização do computador na produção de diagramas tornou possível que se realizasse uma série de operações a partir dos diagramas criados, como por exemplo: verificação de consistência, simulação, geração automática (por exemplo, de placas de circuito impresso) e, obviamente, a reprodução dos diagramas numa impressora. Todas essas operações tornam necessário que sejam construídas uma série de rotinas ou programas utilitários que devem interagir com o editor de diagramas.

2. IMPLEMENTAÇÃO DE DIAGRAMADORES EM MICROCOMPUTADORES

Nos últimos tempos, com a crescente popularização dos microcomputadores com alguma capacidade gráfica, surgiu uma grande quantidade de programas para criação interativa de diagramas voltados para este tipo de equipamentos.

Este trabalho foi parcialmente financiado com recursos do projeto ESTRA da SID INFORMATICA.

Para uma edição confortável, é interessante que a tela do computador seja grande, de tal maneira que o operador visualize todo o digrama (ou pelo menos, uma grande parte dele). Na maior parte dos micros, entretanto, a tela é bastante reduzida, e a qualidade da imagem gerada não é suficiente para trabalhos gráficos com um número razoável de detalhes.

Por outro lado, os micros oferecem uma facilidade muito grande de interface com equipamentos de interação (mouse, mesa digitalizadora), e são suficientemente baratos para que se tenha "um micro para cada projetista". Assim, foram desenvolvidas soluções para minorar os problemas da tela. Essas soluções quase sempre fazem uso frequente de "zoom" e "pan", selecionado, aumentando e diminuindo o trecho do desenho que é visível na tela.

Porém, cada vez que uma dessas operações é realizada, grande parte da tela deve ser redesenhada, o que pode consumir alguns segundos. A imagem é guardada numa estrutura chamada "lista de imagem", que à hora de regerar a tela (pan ou zoom) é interpretada, e transofmrada numa série de comandos gráficos básicos. A maioria dos microcomputadores não possui um controlador gráfico inteligente, e a visualização da memória de vídeo é feita diretamente pelo processador, que possui uma (relativa) baixa velocidade. Uma operação muito comum, passear sobre o desenho, torna-se uma tarefa extremamente cansativa e dispersiva para o usuário.

Para completar este quadro, dentro de um diagrama ainda existem textos e números. Operações de troca de escala, em caracteres, são problemas não triviais em processamento gráfico. As letras, escaladas, perdem sua forma, e quando se ve o gráfico todo na tela, num fator de escala reduzido, ficam completamente ilegíveis.

3. CONCEITOS FUNDAMENTAIS DE TELA VIRTUAL

Hoje é comum que um microcomputador pessoal possua mais do que 500 Kbytes de memória. Imaginando que desejamos representar uma imagem binária, ou seja, contendo apenas as cores preta e branca, uma memória desse tamanho seria suficiente para conter 4 Megapixels.

Uma aplicação simples é a geração de listagens gráficas, onde se utiliza uma área de memória para preparar a imagem que será mandada para uma página de impressora matricial. Nas impressoras comuns (tipo EPSON) uma página contém da ordem de 1600 pontos na horizontal por 800 na vertical, ou seja, 1.3 Megapixel. Um pacote gráfico, para geração de gráficos em impressoras, foi construído, utilizando essa idéia, por um dos autores [1].

Como o vídeo do computador também é uma matriz de bits, é possível copiar o seu conteúdo para a memória principal, ou vice-versa. Uma operação desse tipo leva da ordem de 100 milisegundos (no máximo). Esse fato é utilizado amplamente por diversos programas que trabalham com popup menus ou com algum tipo de animação.

Unindo essas duas observações notamos que é possível utilizar uma área de memória organizada de forma semelhante a memória do vídeo e sobre a qual são executadas as primitivas gráficas. Dessa forma a imagem do desenho criada na memória deve ser movida para a tela de tempos em tempos (ou em horas determinadas). Existem softwares básicos que dão algum suporte a esse tipo de operação, por exemplo, o Turbo Graphix [2].

Essa possibilidade nos leva a idéia de que é possível nos livrarmos das limitações de tamanho da tela e trabalharmos com uma "tela virtual" de tamanho adequado à aplicação gráfica (no caso, o diagramador), como visto na figura 1. O programa seria construído como se estivesse gerando a saída em um terminal externo conectado ao computador. A medida que a operação do programa obrigasse a visualização de um trecho do diagrama, esse seria instantaneamente (em 100 milisegundos) apresentado numa viuporte da tela.

E importante notar que nesse esquema a operação de "pan" é absolutamente natural. Consegue-se uma suavidade na sequência de apresentação das telas da panorâmica comparável a uma panorâmica utilizando uma câmera de televisão. Isso traz um conforto muito grande à operação de um editor de diagramas. Caso seja utilizado um mouse para controle do "pan" a resposta visual é tão eficiente como a movimentação manual de uma folha de papel.

Talvez a maior vantagem do uso de telas virtuais seja a possibilidade do programador trabalhar imaginando que seu programa pode produzir a saída em vários terminais simultaneamente. Repare que isso é muito mais poderoso do que produzir a saída em várias viuportes da tela. Primeiramente, a tela virtual é completamente independente do tamanho da viuporte. Em segundo lugar, uma tela virtual pode ter várias viuportes associadas a ela (eventualmente nenhuma).

4. SOFTWARE BASICO

As primitivas de desenho são construídas a partir de rotinas básicas extremamente simples: pintar um ponto na memória virtual, trazer um pedaço da tela virtual para a viuporte e trazer um pedaço da viuporte para a tela virtual. Usando estas rotinas se constrói um conjunto de rotinas que depende muito da aplicação.

Nossa primeira implementação, que foi suficiente para implementar todo o sistema SOFTLAB [3], continha apenas rotinas de reta, letra, icone e pop-up menu. Nessa implementação as rotinas possuíam três possibilidades de traçado: risco em preto, risco em branco e risco com ou-exclusivo.

Usar o conceito de tela virtual não traz nenhuma dificuldade ao uso das técnicas de computação gráfica, como por exemplo, movimentação de cursores, arrastamento de elementos ("dragging") e conexão elástica ("rubberbanding"). As mesmas técnicas podem ser utilizadas, com praticamente a mesma performance.

Muitas vezes é necessário visualizar todo o digrama (zoom). A utilização de tela virtual também facilita esse tipo de operação. O zoom pode ser realizado de 3 maneiras:

a) Zoom por redesenho da estrutura de dados

Isso é análogo que que seria feito caso não se estivesse utilizando telas virtuais, ou seja, varrer a lista de imagem e reproduzir o desenho, ou numa segunda tela virtual ou diretamente na viuporte.

b) Zoom por compactação

Neste caso um algoritmo muito simples toma vários pontos da tela virtual (por exemplo 8x8 pontos) e se algum dos pontos estiver ligado, então liga-se um ponto na viuporte.

c) Zoom por desenho simultâneo

As rotinas básicas de desenho podem ter a habilidade de trabalhar com várias telas ao mesmo tempo, cada uma com uma escala diferente.

A impressão gráfica é outra vantagem. Os dados da matriz virtual podem ser copiados quase que diretamente para a impressora como num dump de vídeo.

5. USO DE MÚLTIPLAS JANELAS

A necessidade da utilização de várias janelas em um editor de diagramas não é motivada apenas pelo processo de edição em si. Quase sempre o diagramador estará associado a outras operações (por exemplo, programas de consistência ou simulação), cujo resultado deve ser simultaneamente apresentado com o diagrama.

Com base em estudo realizado em diversos softwares gráficos, notamos que existem duas formas mais comuns de manusear múltiplas janelas, que chamaremos automática e controlada.

Na forma automática a ativação de um determinado comando promove a criação de uma viuporte num ponto qualquer do vídeo. Nesse caso, o usuário tem pouco (ou nenhum) controle sobre a viuporte, sendo o seu posicionamento definido pelo programa. Esse tipo de viuporte, algumas vezes, é temporária e se extingue com o fim do comando que a criou.

Na forma controlada, o usuário tem o poder de mudar as dimensões e o posicionamento da viuporte, segundo sua própria conveniência. Normalmente o usuário pode também selecionar uma das viuportes e com ela interagir. Esse tipo de operação é geralemnte encontrada em sistemas operacionais baseados em janelas (como o WINDOWS) [4].

A forma automática é trivialmente implementada com telas virtuais, e o processamento é exatamente o mesmo da operação direta no vídeo. Uma facilidade adicional que existe aqui é que, como todas as viuportes têm uma cópia na memória principal, é possível movê-las, superpô-las ou mesmo apagá-las temporariamente, com um esforço quase zero de programação.

Já a segunda forma exige que o usuário, em tempo de execução, possa alterar parâmetros na estrutura de viuporte. Numa máquina mono-programada como é a maioria dos micro-computadores, fica complicada uma implementação através de processos externos de controle ativados pelo sistema operacional.

Uma solução que foi utilizada nos nossos sistemas, para esse caso, foi o gerenciador de viuporte, uma subrotina que é ativada constantemente pelo programa e que é responsável pelo controle do posicionamento das viuporte no vídeo, alteração interativa de dimensões, operação de pan, além de fornecer informações que permitam o controle pelo programa do status das viuportes.

6. ALGORITMO PARA CRIAÇÃO DE MÚLTIPLAS VIUPORTES UTILIZANDO DE TELAS VIRTUAIS

Para suportar múltiplas viuprotes, o software básico utiliza a estrutura de dados mostrada na figura 2. Esta estrutura contém basicamente uma lista de telas virtuais, e as ifnormações sobre sua posição e trecho mostrado na tela real. Deve-se usar alocação dinâmica em todo este processo, pois o tamanho de cada tela virtual é variável.

As operações que existem são:

. criar uma tela virtual
. definir uma ordem de apresentação das viuportes
. selecionar uma tela de trabalho
. remover uma tela virtual
. selecionar trecho mostrado de uma tela virtual
. recriar imagem do vídeo

Todas essas operações são triviais, exceto recriar a imagem do vídeo. Uma implementação ultra-simplificada poderia desenhar todas as viuportes na ordem definida, obtendo após todas serem desenhadas, a imagem completa. Essa não é uma boa solução, pois, além de gastar muito tempo, não é visualmente agradável.

Uma solução melhor utiliza um algoritmo de rastreio, pintando cada linha de uma vez, do topo para a base do vídeo. Nessa operação, cada linha do vídeo é gerada apenas uma vez, a partir da análise da lista de viuportes. Uma descrição simplificada do algoritmo se segue:

```
1. Cria lista de viuportes ordenada
   pela linha inicial de apresentação na tela
2. Inicializa uma lista de viuportes interceptantes
3. Para cada linha do vídeo
   3.1 Para cada viuporte de lista
       3.1.1 Se linha atual é igual a linha inicial da viuporte
             Coloca viuporte na lista de interceptantes
             Retira da lista ordenada
   3.2 Inicializa a linha do vídeo com a cor de fundo
   3.3 Para toda viuporte da lista de interceptantes
       3.3.1 Move o trecho correspondente da tela virtual para a linha
             do vídeo
       3.3.2 Se é a última linha dessa viuporte
             remove viuporte da lista de interceptantes
```

Nota: É recomendável utilizar um vetor auxiliar ao invés de alterar diretamente a memória do vídeo. O uso desse vetor traz duas vantagens: a linha de varredura não pisca e diminui a probabilidade de bloqueio da CPU do micro, devido ao acesso à memória de vídeo disputado com o controlador.

7. AMBIENTE MULTI-UTILITARIO

Como foi dito anteriormente, um programa diagramador está sempre associado a uma série de funções auxiliares, por exemplo, compilação, consistência, simulação e impressão. Num caso simples, especialmente em microcomputadores sem multiprogramação, essas funções são ativadas de forma independente, mesmo que visualmente se apresentem os resultados em viuportes diferentes e simultaneamente apresentadas.

Porém, as aplicações mais sofisticadas exigem que esses utilitários se intercomuniquem, processando realemnte de forma cooperativa e simultânea. Mesmo naqueles sistemas em que estes requisitos não são muito fortes, a exi

gência de um mínimo de interação entre as funções traz uma certa complexidade para a implementação. Os sistemas monoprogramados como o PC-DOS não fornecem facilidades adequadas para tais implementações.

Naturalmente sempre é possível criar esquemas particulares para a implementação de tais ambientes integrados, porém dois grandes problemas atrapalham tremendamente o desenvolvimento: a criação independente (em separado) dos utilitários, e o seu teste. Em alguns casos, por exemplo, quando dois utilitários se alimentam mutuamente, o que ocorre em algumas simulações de diagramas, o problema é difícil de programar.

Assim, o software básico pode perfeitamente incorporar um esquema genérico que atenda estes requisitos. O esquema que vamos sugerir possibilita uma implementação modular e uma estrutura simples de testes. E também adequado para a ativação do gerenciamento de viuportes, mencionado anteriormente.

8. ESQUEMA GENERICO PARA CRIAÇÃO DE DIAGRAMADORES COM UTILITARIOS

A idéia é muito simples. Os utilitários são criados na forma de uma subrotina independente, que será ativada diretamente de um escalonador. Cada utilitário executa sua função e retorna ao escalonador.

Cada utilitário possui uma caixa postal através da qual envia e recebe mensagens de outros utilitários. Isso possibilita um desenvolvimento muito simples dos utilitários e um esquema do escalonador, que é criado pelo usuário seguindo sempre uma fórmula fixa:

```
inicializa
repete até fim_do_programa
    se tem mensagem na caixa postal ≠ 1
            chama o utilitário ≠ 1
    senão
    se tem mensagem na caixa postal ≠ 2
            chama o utilitário ≠ 2
    senão
            ... etc ...
    senão
    se tem mensagem na caixa postal ≠ 0
            chama a rotina IDLE
    senão
        manda mensagem (ACORDE, caixa postal ≠ 0)
```

A rotina "IDLE" pertence ao software básico e é ativada se não há processo ativo. Ela simplesmente espera que haja uma intervenção manual (por

exemplo, um click de mouse) sobre uma viuporte, e envia uma mensagem para aquele utilitário, o que promove sua ativação.

9. CONCLUSOES

As idéias aqui apresentadas começaram a ser exploradas na criação do software básico para um sistema de CASE desenvolvido por um dos autores (PC-DFD, da Base Tecnologia) [5]. As características desse sistema exigiam uma velocidade altíssima de "pan" e "dragging". O sistema foi criado e respondeu bem às expectativas.

Na UFRJ, o software básico foi reescrito, incorporando as facilidades de múltiplas viuportes e o suporte ao tratamento de mensagens. O esquema padrão de tratamento de utilitários foi utilizado com muito êxito na criação do sistema de CAD para ensino de eletrônica, o SOFTLAB, executando em máquinas PC-XT, com tela monocromática (figura 3).

O software básico foi escrito em Turbo Pascal 4.0, com um total de 2500 linhas (incluindo 10% de comentários). Não foram utilizadas subrotinas em assembler, pois o desempenho foi satisfatório.

O software básico, depois de concluído apresentou dois problemas: o primeiro é a quantidade de memória utilizada pelas telas virtuais, o que rouba 50% do espaço que poderia ser utilizado pelos utilitários. A segunda é uma relativa dependência das características físicas do vídeo, obrigando a existência de diversas versões uma para cada tipo de interface de hardware (felizmente, não existem no Brasil muitos tipos de interfaces de vídeo para PC).

Esses problemas estão sendo agora estudados e está sendo pensada a criação de uma nova versão que incorpore maior portabilidade, que permita também sua utilização em hardware fora da linha PC, e no qual seja possível manter pedaços de telas virtuais, ou fora da memória principal, ou compactados.

BIBLIOGRAFIA

[1] - Borges, J.A. GRIMP - Sistema Gráfico para Impressora, Documentação Interna, NCE/UFRJ, 1985;

[2] - Borland Co. Turbo Pascal Graphix Toolbox - Owners Handbook, version 4.0 1987;

[3] - Assis, J.S. SOFTLAB - Sistema de Ensino de Eletrônica Linear, Projeto de Fim de Curso de Informática - UFRJ, em preparação;

[4] - Borges, J.A.; Schmitz, E. SOFTLAB - Uma Estação de Instrução Auxiliada por Computador para Ensino de Eletrônica. Congresso da SUCESU, 1986;

[5] - Microsoft Co. Microsoft Windows - User's Guide, version 2.0, 1987;

[6] - Base Tecnologia Ltda. - PC-DFD PLUS - Manual de Operação.

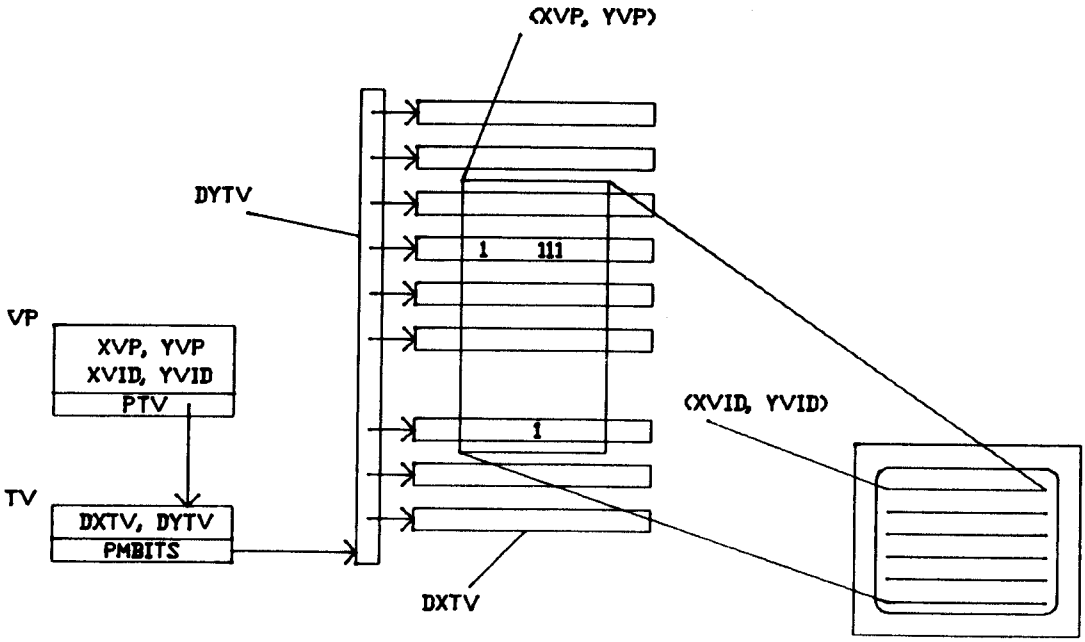

(XVP, YVP)
DYTV
1 111
VP
XVP, YVP
XVID, YVID
PTV
(XVID, YVID)
1
TV
DXTV, DYTV
PMBITS
DXTV

FIG. 1

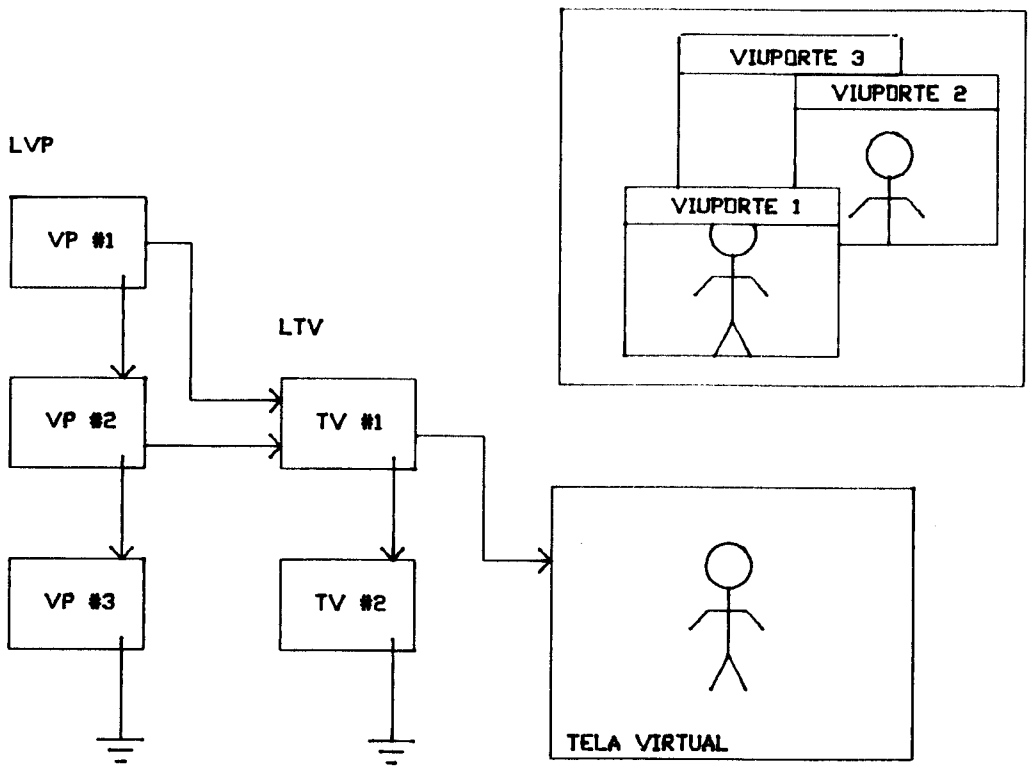
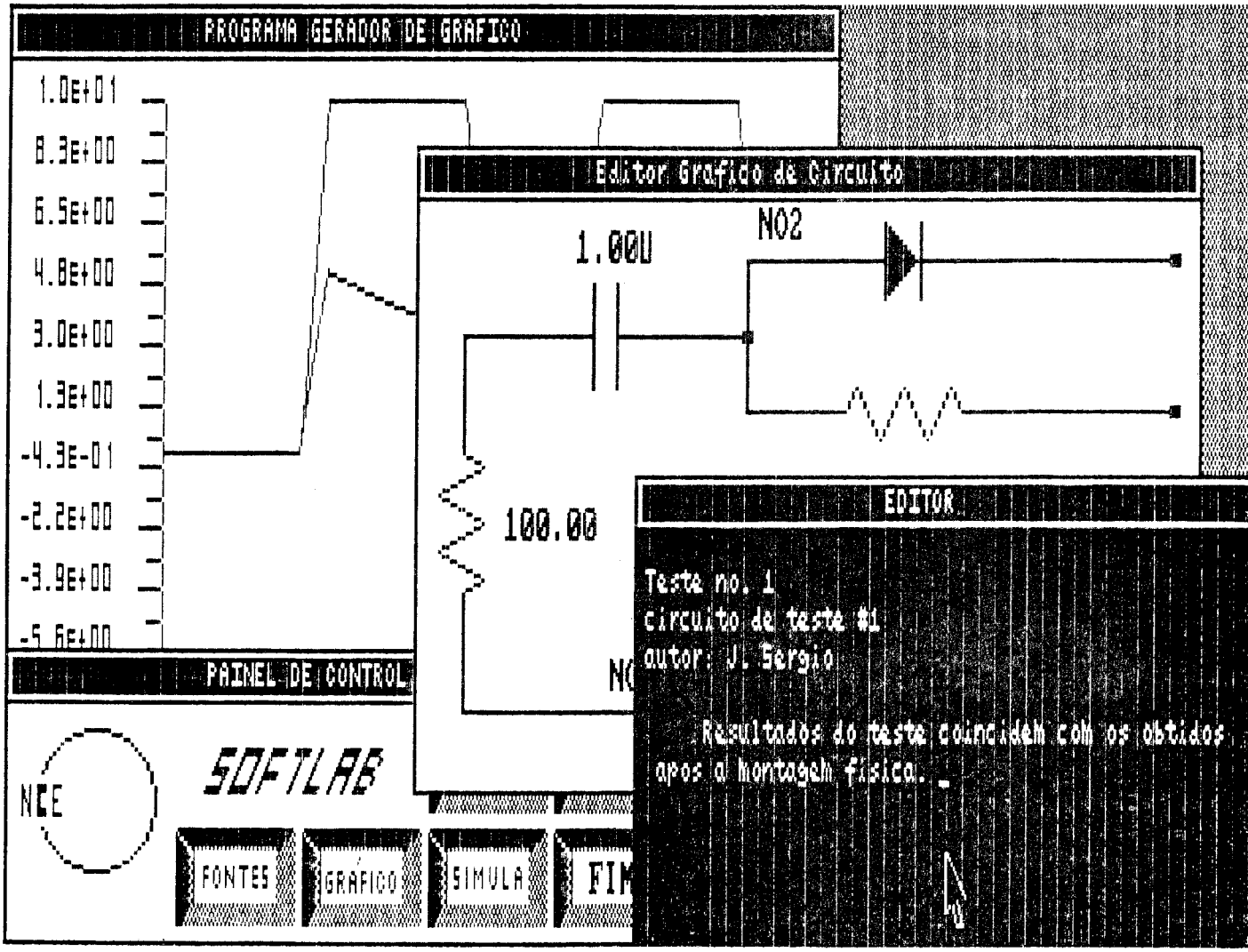

VIUPORTE 3
VIUPORTE 2
VIUPORTE 1
LVP
VP #1
LTV
VP #2
TV #1
VP #3
TV #2
TELA VIRTUAL

FIG. 2


PROGRAMA GERADOR DE GRAFICO
1.0E+01
8.3E+00
6.5E+00
4.8E+00
3.0E+00
1.3E+00
-4.3E-01
-2.2E+00
-3.9E+00
Editor Grafico de Circuito
1.00U
NO2
100.00
EDITOR
Teste no. 1
circuito de teste #1
autor: J. Sergio
Resultados do teste coincidem com os obtidos
apos a montagem fisica.
PAINEL DE CONTROL
SOFTLAB
NCE
FONTES
GRAFICO
SIMULA
FIM