

UNIVERSIDADE ESTADUAL DE CAMPINAS
INSTITUTO DE QUÍMICA

*"Este exemplar corresponde à redação
final da Tese defendida por Claudimir Lucio
do Lago e aprovado pela Comissão Julgadora."*

Concetta Kascheres

Título:

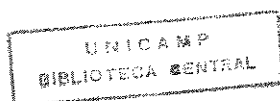
**"DESENVOLVIMENTO DE SOFTWARE E HARDWARE
PARA ESPECTROMETRIA DE MASSAS E
CROMATOGRAPHIA GASOSA/ESPECTROMETRIA
DE MASSAS"**

Claudimir Lucio do Lago
Tese de Doutorado

Orientadora: Profa. Dra. Concetta Kascheres

Campinas - 1991

BC/9104483



Dedico este trabalho a quem tenho dedicado minha vida:

Lilian,

Felipe

e ao neném que vem em junho

**"X.3 1 A BM' K Xaffz v bb Lh,, n jicxc
felipe**

"Se eu falasse a lingua dos anjos..."

AGRADECIMENTOS

Meus agradecimentos:

À minha orientadora Professora Concetta Kascheres pela dedicação, incentivo, amizade e por acreditar em minha capacidade, dando liberdade para realizar este trabalho.

À direção do IQ da UNICAMP e IQ da USP.

À FAPESP pela bolsa de mestrado (processo nº 86/2879-0).

Aos professores Serruya, Maria Helena e Pio e ao técnico Tenório da UFPa que, apesar da pouca disposição da direção, colaboraram para o desenvolvimento do sistema de dados para GC/MS.

Ao pessoal da biblioteca, xerox, vidraria e das oficinas mecânica, de manutenção e eletrônica.

Ao Aparecido (Cidão) pela paciente ajuda dada durante estes quase seis anos de trabalho na sala de Espectrometria de Massas.

Ao pessoal do grupo da Profa. Connie -Gina, Luciana, Regina e Rodinei (Rudão)- pelo auxílio dado quando tive que entrar no laboratório.

Ao mestre Aurélio pelas pacientes explicações na área de computação e matemática.

Aos revisores desta tese: Valdir, Marcinha, Fábio (Fabiosa), Marcelo e em especial ao amigo e sócio Valmir.

Ao seu João, à dona Dirce e aos meus irmãos.

À Lilian que, apesar da falta de colaboração, é minha fonte de inspiração.

Obrigado

RESUMO

Foram desenvolvidos dois sistemas de dados - incluindo *software* e *hardware* - para espectrometria de massas e cromatografia gasosa acoplada a espectrometria de massas. Um dos sistemas, instalado no Instituto de Química da UNICAMP, adquire dados de dois espectrômetros de massas, sendo um Finnigan 1015S/L quadrupolar e outro MAT311A com dupla focalização. O outro sistema, instalado na UFPa, controla um cromatógrafo a gás Finnigan 9600 enquanto adquire dados de um espectrômetro de massas quadrupolar 4021, do mesmo fabricante.

O espectrômetro 1015S/L opera na faixa de 1 a 100d com resolução constante de 1 dalton. Embora atinja valores acima de 600d a baixa resolução, foi possível a aquisição de espectros no MAT311A, com valores confiáveis de massa, somente até 100d.

Este segundo equipamento não possuía um sensor para campo magnético e, assim, desenvolvemos um baseado na atuação do efeito Hall sobre o transistor unijunção 2N2646. Com a utilização de um par destes transistores trabalhando próximos, foi possível diminuir o efeito da deriva térmica que causa, para região de trabalho citada, um desvio inferior a $0,5d/^{\circ}C$.

Para aquisição de dados do espectrômetro 4021, foi necessário desenvolver um conversor A/D com 12 bits de resolução. O tempo de conversão, não otimizado, é de aproximadamente $70\mu s$. A precisão foi determinada por cálculos como sendo aproximadamente 0,1%, para um intervalo de $\pm 30^{\circ}C$, ou 0,04%, para um intervalo de trabalho mais realista de $\pm 3^{\circ}C$.

Foram desenvolvidos programas em linguagem Turbo Pascal para calibração, aquisição e tratamento de espectros. Para o sistema da UNICAMP, o processo de calibração é semi-automático, enquanto que o sistema da UFPa faz a calibração automaticamente, bastando injetar a amostra do padrão.

Além dos programas básicos, foram desenvolvidos dois outros com finalidades específicas: um para análise de padrões isotópicos e outro para resolução de espectros de misturas por análise fatorial. Estes programas permitiram o desenvolvimento de uma técnica de interpretação de espectros que utiliza a variação da energia de ionização. A idéia geral é tratar o espectro de massas como mistura de uma grande variedade de íons. Com a variação do potencial aplicado na câmara de ionização, a proporção dos íons formados se altera, permitindo obter os padrões isotópicos destes íons componentes através do programa de análise fatorial.

ABSTRACT

Two data systems were developed -software and hardware- for mass spectrometry and gas chromatography/mass spectrometry. The first, located at IQ-UNICAMP, does the data acquisition of two mass spectrometers: a Finnigan 1015S/L quadrupole and a MAT311A double focusing system. Located at UFPa, the other system controls a Finnigan 9600 gas chromatograph while it acquires data from a Finnigan 4021 quadrupole mass spectrometer.

The mass range of the 1015S/L spectrometer is 1 to 100d with constant resolution of one dalton. Although the MAT311A spectrometer scans above 600d, it was only possible to acquire mass spectra up to 100d with adequate accuracy. This mass spectrometer does not have a magnetic field probe. Therefore, we developed a probe with the 2N2646 unijunction transistor. A pair of components working close together reduces the temperature coefficient to $0.5\text{d}/^{\circ}\text{C}$.

In order to acquire data from the 4021 mass spectrometer, it was necessary to develop a 12 bit A/D converter. The conversion time is $70\mu\text{s}$ and the accuracy is 0.1% for $\pm 30^{\circ}\text{C}$ or 0.04% for $\pm 3^{\circ}\text{C}$.

Developed in Turbo Pascal, the software calibrates, acquires and works up the spectra. The calibration procedure for the UNICAMP system is semiautomatic, while for the UFPa system it is fully automatic.

Two programs were created for special treatment: Isotope Pattern Analysis and Factor Analysis. These programs can be used for mass spectra interpretation. We can consider a mass spectrum as a mixture of overlapping isotope patterns which have intensities that are determined by the electron energy. The strategy of the method is based on the separation of the pure isotope patterns from a set of spectra obtained by scanning the electron energy.

ÍNDICE

Resumo	1
Abstract	3
Índice	4
Abreviações e Símbolos	7
Capítulo I: Objetivos	I.1
Capítulo II: Introdução	II.1
1 Espectrometria de Massas	II.1
1.2 Analisador de Massas	II.2
1.2.1 Setor Magnético	II.3
1.2.2 Filtro Quadrupolar	II.5
1.3 Detector	II.8
2 Cromatografia Gasosa	II.9
3 O Microcomputador IBM PC	II.10
4 Bibliografia	II.11
Capítulo III: Interfaceamento	III.1
1 Equipamentos Utilizados	III.1
1.1 FINNIGAN 1015 S/L	III.2
1.2 MAT 311A	III.2
1.3 FINNIGAN 4021	III.3
2 Sistema de Dados do IQ-UNICAMP	III.3
2.1 Hardware	III.4
2.2 Sensor para Campo Magnético	III.9
2.2.1 Princípio de Funcionamento	III.13
2.2.2 Interpretação do Efeito do Campo sobre V_p e V_v	III.20
2.2.3 Avaliação do Sensor	III.21

2.3 Software	III.24
3 Sistema de Dados do CCEN-UFPA	III.27
3.1 Hardware	III.27
3.2 Conversor Analógico/Digital	III.32
3.2.1 Princípio de Funcionamento	III.32
3.2.2 Avaliação do Conversor A/D	III.37
3.3 Software	III.43
4 Bibliografia	III.45

Capítulo IV: Programas para Os Sistemas de Dados	IV.1
1 Programas para o Finnigan 1015S/L	IV.1
1.1 O Programa CAL100	IV.2
1.2 O Programa AJUSTE	IV.4
1.3 O Programa COL100	IV.5
1.4 O Programa MD2	IV.6
2 Programas para o MAT311A	IV.7
2.1 O Programa MATCOL	IV.7
2.2 O Programa MATTRATN	IV.9
3 Programas para o Finnigan 4021	IV.10
3.1 O Programa CAL4021	IV.11
3.2 O Programa GCMS	IV.12
3.3 O Programa TRATA	IV.17
4 Bibliografia	IV.23

Capítulo V: Análise de Padrões Isotópicos	V.1
1 Fundamentos e Implementação do Programa AC4	V.3
1.1 Condições de Análise	V.3
1.2 Geração de Padrões Isotópicos	V.5
1.3 Comparação de Padrões Isotópicos	V.8
1.4 Análise de Misturas de A com A-1	V.11
1.5 Eliminação de Padrões Contaminantes	V.13
2 Parte Experimental	V.14
3 Avaliação do Programa	V.15
4 Bibliografia	V.19

Capítulo VI: Análise Fatorial	VI.1
1 Fundamentos Utilizados no Programa FA3	VI.2
2 Avaliação do Programa FA3	VI.5
3 Estudo sobre o Potencial de Ionização	VI.12
3.1 Aplicação do Método	VI.14
3.2 Discussão e Conclusões sobre o Método	VI.21
4 Bibliografia	VI.23

Capítulo VII: Avaliação e Conclusões	VII.1
1 Sistema de Dados do IQ-UNICAMP	VII.1
2 Sistema de Dados do CCEN-UFPa	VII.2
3 Programas AC4 e FA3	VII.3

Apêndice 1: Relação dos Componentes dos Diagramas Eletrônicos

Apêndice 2: Listagem dos Programas

Apêndice 3: Resultados de Análises Descritas no Capítulo VI

ABREVIACÕES E SÍMBOLOS

A/D	analógico(a)/digital
CT	coeficiente de temperatura
D/A	digital/analógico(a)
DC	corrente contínua
E/S	entrada/saída
GC/MS	cromatografia gasosa/espectrometria de massas
H	após um valor numérico indica base hexadecimal
LSB	bit menos significativo
MS/MS	espectrometria de massas/espectrometria de massas
m/z	massa/carga
RAM	memória de acesso aleatório
RC	resistor-capacitor
RF	rádio frequência
S&H	<i>sample and hold</i>
σ	desvio padrão
TIC	corrente iônica total

CAPÍTULO I

OBJETIVOS

A espectrometria de massas, nascida no início do século, viu o surgimento dos computadores eletrônicos, absorvendo rapidamente as inovações nesta área e tornando a técnica cada vez mais poderosa na análise de composição e estrutura química. Inicialmente, devido ao custo e recursos dos computadores, os procedimentos eram *off line*, porém a medida que a eletrônica se desenvolveu, os espectrômetros foram acoplados diretamente aos computadores, criando-se inclusive sistemas dedicados.

Devido a constante evolução, os sistemas de dados ficam obsoletos muito antes que o espectrômetro e a manutenção de antigos computadores tem custo elevado. Este é o caso de dois equipamentos da Finnigan Co. instalados no Brasil: um espectrômetro modelo 1015 S/L instalado no Instituto de Química da UNICAMP e outro 4021 instalado no Centro de Ciências Exatas e Naturais da Universidade Federal do Pará.

Para casos como estes torna-se atrativo o acoplamento a microcomputadores nacionais de custo razoável e fácil manutenção. Dentre os padrões existentes no país, o IBM PC/XT se firmou na década de 80 como o mais versátil e comercial microcomputador. Outra vantagem deste padrão é a compatibilidade de *software* e *hardware* de modelos mais antigos com os mais recentes, permitindo que o trabalho desenvolvido para um sistema não se perca com a evolução da linha.

- OBJETIVOS -

Assim, os objetivos deste trabalho que envolvem a criação e aplicação de novos sistemas de dados são:

-Interfaceamento de um espectrômetro de massas quadrupolar Finnigan 1015 S/L e outro com dupla focalização MAT 311A a um microcomputador compatível com o IBM PC.

-Interfaceamento de um cromatógrafo gasoso 9600 acoplado a um espectrômetro de massas Finnigan 4021 a um microcomputador compatível com o IBM XT.

-Desenvolvimento de programas para calibração, aquisição e interpretação de espectros para os sistemas criados.

-Estudo utilizando a variação do potencial de ionização como forma de separar misturas de *clusters* de fragmentos.

CAPÍTULO II

INTRODUÇÃO

Neste capítulo serão apresentados alguns fundamentos das técnicas e equipamentos utilizados neste trabalho. Serão abordados aspectos relacionados com o trabalho desenvolvido, procurando dar subsídios para facilitar o entendimento dos capítulos seguintes.

1 Espectrometria de Massas

A técnica de espectrometria de massas surgiu no início do século com trabalhos como de J.J. Thomson de 1913, F.W. Aston e A.J. Dempster [1] e, desde então, uma grande variedade de métodos de ionização, separação e detecção vêm sendo introduzidos.

Apesar da grande variedade, o princípio de funcionamento ainda é o mesmo. As moléculas da amostra são ionizadas e por estarem no estado gasoso podem ter seus movimentos controlados por campos elétricos e magnéticos externos. Esta propriedade é utilizada para separar os íons de acordo com sua relação massa/carga (m/z), permitindo que somente íons dentro de uma estreita faixa desta relação atinjam o detector em um determinado momento. Para que não ocorram colisões entre os íons formados e outras moléculas gasosas, a pressão no interior do espectrômetro de massas deve estar abaixo de 10^{-4} Torr. A seguir cada uma das etapas do processo será analisada.

- INTRODUÇÃO -

1.1 Fonte de íons

Existe uma grande variedade de métodos de ionização que podem produzir íons positivos ou negativos, além da fragmentação da molécula [2]. Os métodos mais utilizados são a ionização por impacto de elétrons (EI) e a ionização química (CI). Para compreender os dois métodos é importante entender como se processa a ionização de uma espécie na fase gasosa, quando esta é bombardeada por um feixe de elétrons.

Consideremos uma molécula A no estado gasoso em uma região onde haja um fluxo de elétrons. Se houver a colisão de um elétron com energia suficiente, poderá ocorrer a ionização:



Caso a energia do elétron seja superior à necessária para ocorrer este processo, ou houver a colisão de um segundo elétron, poderão ocorrer outros processos, gerando uma grande quantidade de espécies de fragmentos e rearranjos com uma ou mais cargas.

Na ionização química, um gás reagente é introduzido com pressão muito superior à da amostra. As moléculas deste gás serão preferencialmente ionizadas, gerando espécies que, por colisão com as moléculas da amostra, irão causar a sua ionização.

1.2 Analisador de Massas

Sendo uma partícula eletricamente carregada, o íon pode ter sua trajetória comandada por campos elétricos e/ou magnéticos externos.

Quando esta partícula é submetida a uma diferença de potencial elétrico, fica sujeita a uma força na direção deste campo e com sentido determinado por sua carga elétrica. Esta força provocará a aceleração da partícula e, ao sair desta região, terá um ganho de energia cinética igual ao produto de sua carga elétrica pela diferença de potencial. Esta é a maneira usual de remover os íons da câmara de ionização para introduzir no

- INTRODUÇÃO -

analisador de massas.

Existem vários tipos de analisadores baseados na ação de campos eletromagnéticos. Neste trabalho foram utilizados dois tipos, que são os mais comuns: Setor Magnético e Quadrupolo.

1.2.1 Setor Magnético

Quando submetido a um campo magnético, uma partícula eletricamente carregada e em movimento ortogonal à direção deste campo sofrerá a ação de uma força perpendicular a sua trajetória, que provocará a sua curvatura. Igualando a força necessária para provocar uma curvatura de raio r sobre uma partícula de massa m e velocidade v , com a força imposta por um campo magnético de intensidade B sobre uma partícula de carga z , teremos:

$$\frac{mv^2}{r} = Bzv \quad (2)$$

A velocidade final da partícula pode ser obtida igualando-se a energia potencial gerada pelo campo elétrico V no interior da câmara de ionização com a energia cinética na saída:

$$\frac{1}{2}mv^2 = Vz \quad (3)$$

De (2) e (3), obtemos a equação que governa o analisador de campo magnético:

$$\frac{m}{z} = \frac{r^2 B^2}{2V} \quad (4)$$

A figura 1 mostra um espectrômetro de massas simples utilizando um setor magnético para selecionar os íons que atingirão o detector. Para fazer esta seleção, basta fixar V e

- INTRODUÇÃO -

ajustar B ou vice-versa. As duas maneiras são possíveis, porém a varredura do campo magnético, mantendo-se o potencial de aceleração constante, é o método mais comum de se adquirir um espectro.

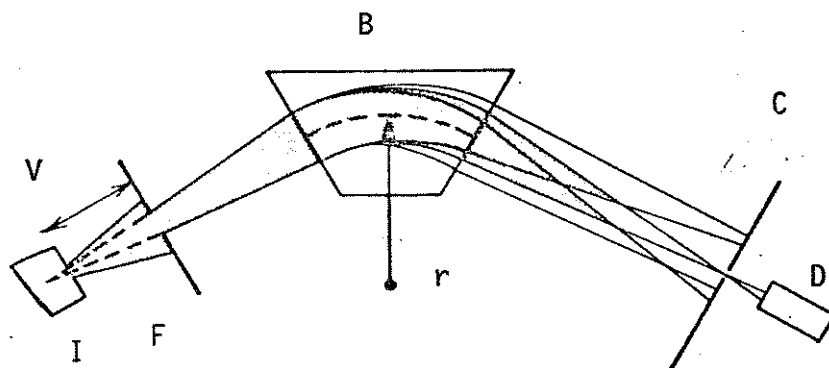


Figura 1 - Espectrômetro de massas com setor magnético. Os íons formados na câmara de ionização I, são acelerados pelo potencial V até a fenda F. Os íons que passam por esta fenda são separados de acordo com m/z pelo campo magnético B. A fenda C permite que íons dentro de uma estreita faixa de m/z atinjam o detector D [3].

A equação 3 considera que a única contribuição para a energia cinética é a diferença de potencial aplicada, porém, por estarem em fase gasosa, os íons formados possuem movimento e sua energia cinética será somada à contribuição externa. Assim, íons de mesma relação m/z possuem trajetórias diferentes. Este fato diminui a capacidade de separação do filtro magnético (resolução).

Para eliminar este efeito, em equipamentos de alta resolução é acrescentado um outro setor com a finalidade de selecionar os íons de acordo com a energia cinética. O dispositivo, posicionado entre a fonte de íons e o setor magnético, é composto por duas placas curvas e paralelas com raio R que geram um campo elétrico E perpendicular à trajetória dos íons como mostrado na figura 2. Os íons que atravessarem este dispositivo devem obedecer a equação

$$\frac{mv^2}{R} = zE \quad (5)$$

- INTRODUÇÃO -

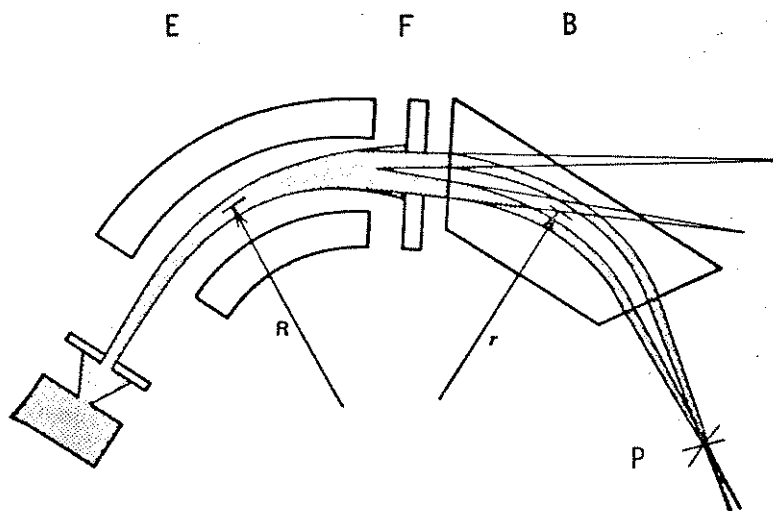


Figura 2 - Espectrômetro de massas com dupla focalização. O setor eletrostático E, juntamente com a fenda F, permite que íons dentro de uma faixa de energia cinética sejam introduzidos no setor magnético B. O duplo foco P é determinado por E e B [9].

Sendo R e E constantes, este setor eletrostático funciona como um analisador de energia cinética, permitindo selecionar, dentro de uma estreita faixa de energia, os íons que serão introduzidos no setor magnético.

Os equipamentos que possuem esta estrutura são conhecidos como de dupla focalização com engenharia Nier-Johnson. Outros equipamentos, como é o caso do MAT 311A utilizado neste trabalho, possuem os setores magnético e eletrostático invertidos.

1.2.2 Filtro Quadrupolar

Este tipo de filtro de massas possui princípio de funcionamento e características bastante diferentes do de setor magnético. O filtro é composto por quatro barras metálicas com secção circular ou parabólica precisamente alinhadas (figura 3). As barras opostas são eletricamente conectadas, onde se aplicam potenciais alternados com polaridades invertidas e os íons são introduzidos na direção do eixo z.

- INTRODUÇÃO -

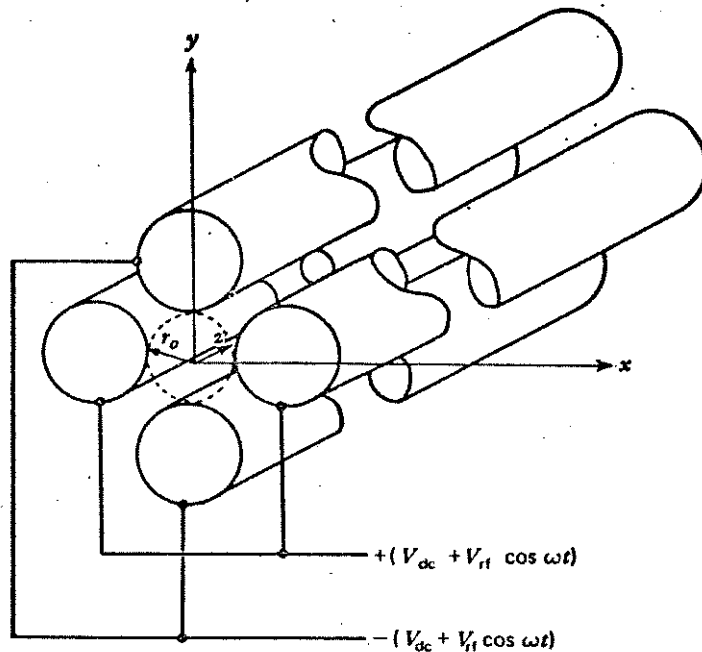


Figura 3 - Filtro Quadrupolar. Os íons são injetados na direção do eixo z. Os potenciais oscilantes aplicados nos eletrodos x e y alteram a trajetória e determinam qual a faixa de m/z atingirá o detector [3].

O princípio de funcionamento é mais complexo que para o tipo anterior e uma explicação qualitativa e quantitativa detalhada pode ser encontrada na referência 4.

O potencial aplicado no par de eletrodos do eixo x é a soma de um potencial contínuo de intensidade V_{dc} e outro alternado de amplitude V_{rf} :

$$V_x = V_{dc} + V_{rf} \cos(\omega t) \quad (6)$$

onde ω é a frequência angular do sinal alternado e t é o tempo. Os eletrodos do eixo y estão submetidos a um potencial semelhante, porém com polaridade invertida:

$$V_y = -V_{dc} - V_{rf} \cos(\omega t) \quad (7)$$

- INTRODUÇÃO -

Como a amplitude do sinal alternado é, em módulo, maior que o contínuo, durante uma parte do ciclo as polaridades se invertem. Assim, os íons, ao entrarem na região do quadrupolo, serão inicialmente atraídos por um conjunto de barras que possuam carga oposta à sua e repelidos pelo outro conjunto, tendo suas trajetórias alteradas. Instantes depois, as polaridades se invertem e, por conseguinte, as trajetórias. Isto provoca uma oscilação no movimento dos íons, que pode provocar instabilidade suficiente para fazer com que estes colidam com um dos eletrodos ou escapem lateralmente do quadrupolo. É possível demonstrar [4,5] que para selecionar íons dentro de uma estreita faixa de massas (aproximadamente um dalton) é necessário que as equações 8 e 9 sejam obedecidas:

$$a = \frac{8zV_{dc}}{mr_o^2\omega^2} \quad (8)$$

$$q = \frac{4zV_{rf}}{mr_o^2\omega^2} \quad (9)$$

onde r_o é a metade da distância entre as superfícies de eletrodos opostos e os valores de a e q devem ser o mais próximo possível de 0,2367 e 0,7060, respectivamente, para se obter o máximo de resolução. Sendo a e q constantes, nota-se que deve haver um compromisso entre V_{dc} e V_{rf} :

$$V_{dc} = \frac{a}{2q} V_{rf} \quad (10)$$

Mantendo-se a distância entre os eletrodos e a frequência constantes, a relação m/z varia linearmente com a amplitude dos potenciais V_{dc} e V_{rf} :

- INTRODUÇÃO -

$$\frac{m}{z} = \frac{8}{ar_o^2 \omega^2} V_{dc} \quad (11)$$

ou

$$\frac{m}{z} = \frac{4}{qr_o^2 \omega^2} V_{rf} \quad (12)$$

O método convencional de fazer uma varredura é variar linearmente com o tempo os potenciais aplicados aos eletrodos, enquanto o sinal do detector é registrado.

1.3 Detector

A quantidade de amostra utilizada em uma análise é pequena (da ordem de microgramas ou nanogramas). Deste total uma parcela ainda menor é ionizada e, devido à fragmentação, uma quantidade muitíssimo pequena de um determinado íon é produzida, sendo que parte destes íons são perdidos no processo de separação. Para detectar quantidades tão pequenas de partículas eletricamente carregadas é utilizado o multiplicador de elétrons.

Este dispositivo é composto por uma série de placas (dinodos) submetidas a uma diferença de potencial. Ao colidir com o primeiro dinodo, um íon remove deste, em média, dois elétrons, que caminharão em direção ao segundo dinodo, repetindo o processo. Após pouco mais de uma dezena de dinodos, a quantidade de elétrons já é o suficiente para gerar uma corrente detectável por um amplificador eletrônico convencional.

O tipo de detector utilizado nos espectrômetros deste trabalho é uma variação do modelo descrito acima (figura 4), não possuindo uma coleção de dinodos mas sim um tubo metálico, cônico e curvo. Nas extremidades deste cone é aplicada uma diferença de potencial elevada. Quando um íon atinge a parede interna do cone, o processo de remoção ocorre, formando uma cascata de elétrons em direção à outra extremidade que está conectada ao amplificador.

- INTRODUÇÃO -

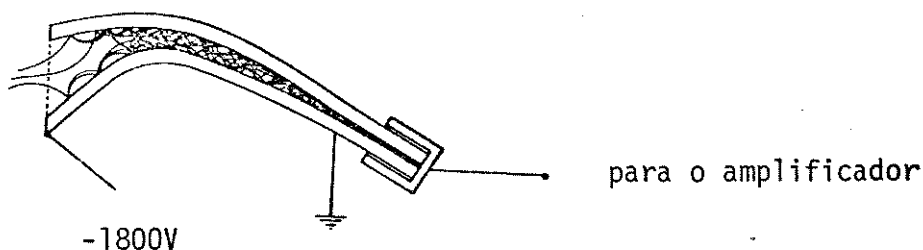


Figura 4 - Multiplicador de elétrons tipo dinodo contínuo. Na abertura é aplicada uma alta diferença de potencial negativa (aproximadamente 1800V) que atrai os cátions, provocando colisões na parede interior. Uma camada interna de óxido metálico, além de emitir elétrons após a colisão, proporciona uma dosagem de potencial elétrico pelo tubo. Como o potencial mais positivo se encontra na outra extremidade, os elétrons caminham neste sentido, provocando outras colisões.

2 Cromatografia Gasosa

Praticamente qualquer material a ser analisado é uma mistura, assim, torna-se importante separá-la em seus componentes para que sejam identificados e quantificados. A cromatografia é o método mais utilizado para esta finalidade.

Um cromatógrafo a gás é composto por injetor, coluna e detector. A amostra injetada é carregada por um gás através da coluna, que separa a mistura de acordo com a interação dos componentes com a mesma. O efluente da coluna é injetado em algum tipo de detector (ionização de chama, condutividade térmica, etc.) que permite quantificar os componentes.

O processo cromatográfico consiste no parcionamento de um soluto entre duas fases: a fase estacionária e a fase móvel gasosa. Sendo assim, a interação entre um determinado componente e a fase estacionária determina o tempo que este demorará para atravessar a coluna. Este é o tempo de retenção e depende não somente deste fator, mas de vários outros como comprimento e diâmetro interno da coluna, velocidade do gás de arraste, temperatura, espessura e composição da fase estacionária.

- INTRODUÇÃO -

O ponto crítico da interface entre cromatógrafo a gás e espectrômetro de massas é a necessidade de reduzir a pressão no interior da coluna, que é superior a 760 Torr, para menos de 10^{-4} Torr no interior do espectrômetro. Quando se usa coluna empacotada, a vazão é muito grande e para remover o gás de arraste e boa parte da amostra, pode ser utilizado um separador de jato [6]. Com colunas capilares, devido à pequena vazão, a conexão direta da saída da coluna com a câmara de ionização do espectrômetro é possível.

3 O Microcomputador IBM PC

A linha de microcomputadores IBM PC é baseada no microprocessador de 16 bits 8088 da Intel Corporation. Este é uma versão simplificada do 8086 possuindo um barramento de dados com somente 8 bits. O barramento de 20 linhas permite endereçar até 1MByte (com segmentos de 64KBytes) de posições de memória e 64KBytes de portas de entrada/saída (E/S)[7]. Deste total, estão disponíveis no *slot* de expansão somente as portas de E/S com endereços de 0200H a 03FFH. Isto corresponde a 512 portas disponíveis para expansões, sendo que muitas estão comprometidas com dispositivos como impressoras, interface serial e *disk drive* [8].

Inicialmente os microcomputadores desta linha operavam a 4,77 MHz, mas o *clock* foi aumentado, sendo que os utilizados neste trabalho operam a 8 MHz.

Os dispositivos de E/S podem ser conectados em posições de memória ou como portas de E/S. A técnica de mapeamento de memória proporciona flexibilidade de programação, pois qualquer instrução que referencie a memória pode ser usada. Por outro lado, ao dedicar parte da memória aos dispositivos de E/S, o espaço disponível ao usuário fica reduzido.

Para que não haja conflito de endereços de memórias e dispositivos internos do microcomputador com alguma nova interface, deve-se localizá-la acima do endereço da última posição

- - INTRODUÇÃO -

de memória RAM e abaixo do endereço F0000H ou entre 300H e 3FFH, para o caso de porta de E/S.

4 Bibliografia

- 1 - C.A. McDowell, "Mass Spectrometry", McGraw-Hill, New York, 1963, p. 59.
- 2 - R.W. Kiser, "Introduction to Mass Spectrometry and Its Applications", Prentice-Hall, New Jersey, 1965, p. 32.
- 3 - G.M. Message, "Practical Aspects on Gas Chromatography/Mass Spectrometry", John Wiley & Sons, New York, 1984, cap. 3.
- 4 - P.E. Miller and M.B.Denton, *J. Chem. Educ.* 63, 617 (1986).
- 5 - R.W. Kiser, "Introduction to Mass Spectrometry and Its Applications", Prentice-Hall, New Jersey, 1965, p. 86.
- 6 - R.R. Freeman, "High Resolution Gas Chromatography", Hewlett-Packard Co., 2nd Ed., 1981, p. 106.
- 7 - W.A. Dias Junior, "Microprocessadores 8086/8088: Hardware e Software", McGraw-Hill, São Paulo, 1990, p. 83.
- 8 - L.C. Eggebrecht, "Interfacing to the IBM Personal Computer", Howard W. Sams, 2nd Ed., 1990, p. 196.

CAPITULO III

INTERFACEAMENTO

Neste capítulo serão vistos os vários aspectos relacionados com os trabalhos de interfaceamento dos espectrômetros de massas e cromatógrafo aos microcomputadores para criação dos sistemas de dados. Os diagramas dos circuitos dos equipamentos comerciais utilizados não serão apresentados, assim como de cartões aproveitados integralmente.

Serão apresentados somente os programas utilizados para a aquisição básica de espectros. Os programas mais avançados para calibração, aquisição e tratamento de espectros serão discutidos no próximo capítulo.

1 Equipamentos Utilizados

O trabalho foi desenvolvido baseado em três equipamentos com características diferentes: um espectrômetro com setor magnético (MAT 311A), outros dois quadrupolares (Finnigan 1015 S/L e 4021), sendo um acoplado a um cromatógrafo (Finnigan 9600).

Para o controle dos espectrômetros Finnigan 1015 S/L e MAT 311A foi utilizado um microcomputador compatível com o IBM PC com clock de 8MHz, 256kbytes de RAM e duas unidades de disco de 5 1/4". O controle do espectrômetro 4021 e cromatógrafo ficou por conta de um microcomputador compatível com o IBM XT com clock de 8MHz, 512kbytes de RAM, uma unidade de disco de 5 1/4" e um disco winchester de 30Mbytes. Os dois microcomputadores não possuem

coprocessador matemático.

1.1 FINNIGAN 1015 S/L

O espectrômetro de massas Finnigan 1015 S/L instalado no IQ da UNICAMP é do tipo quadrupolar e originalmente trabalhava acoplado a um cromatógrafo gasoso e era controlado por um sistema de dados do mesmo fabricante. Devido às dificuldades de manutenção, tanto a interface GC/MS quanto o sistema de dados deterioraram a ponto de se tornarem inoperantes.

A primeira iniciativa no sentido de se desenvolver um sistema de dados nacional foi feita em 1984. O projeto previa a aquisição de um microcomputador NEXUS 1600 da Scopus Tecnologia, utilização do cartão contendo os conversores A/D e D/A do sistema original e desenvolvimento de software para aquisição de espectros. Apesar das limitações de velocidade e limite de m/z , o sistema funcionou satisfatoriamente [1], sendo que o microcomputador e vários componentes serviram de base para o atual sistema de dados.

Atualmente o espectrômetro possui resolução unitária na região de 1 a 100 daltons e permite a introdução de amostras sólidas e líquidas.

O circuito de DC/RF do quadrupolo é controlado por um sinal contínuo entre 0 e -10V que seleciona m/z entre 0 e 100 daltons. Este ponto de controle pode ser utilizado externamente para fazer a varredura de um espectro.

1.2 MAT 311A

Este espectrômetro de massas de duplo setor - um magnético e outro eletrostático - com engenharia reversa Nier-Johnson também está instalado no IQ da UNICAMP. O espectro pode ir acima de 600 daltons e, embora este seja um equipamento desenvolvido para atuar em alta resolução, tem sido utilizado para aquisição de espectros de baixa resolução.

O equipamento originalmente instalado não possuía um sistema digital de dados, sendo os espectros registrados em papel

- INTERFACEAMENTO -

fotossensível. Isto encarece bastante a análise, além de dificultar qualquer tratamento dos dados.

Este equipamento também não possuía originalmente um sensor para campo magnético e a relação m/z era determinada indiretamente pela corrente que atravessa a bobina. Esta determinação é muito imprecisa devido a histerese do núcleo metálico, que faz com que não haja uma relação fixa entre esta corrente e a intensidade do campo magnético.

1.3 FINNIGAN 4021

O sistema é composto por um cromatógrafo a gás modelo 9600 acoplado a um espectrômetro de massas quadrupolar. Originalmente um cartão contendo um microprocessador 8008 controlava o cromatógrafo e um sistema de dados fazia a aquisição de espectros. As constantes e intensas chuvas da região (Belém - Pará) provocam descargas elétricas que, através da rede elétrica, acabaram por danificar o cartão controlador do cromatógrafo e o computador que controlava o espectrômetro. Deste sistema de dados foi possível aproveitar o conversor D/A de 16 bits DAC16 da Analog Devices responsável pelo controle do quadrupolo.

As condições deste equipamento são bastante satisfatórias: o cromatógrafo opera com colunas empacotadas e capilares e o espectrômetro possui resolução unitária na região de 10 a 1000 daltons.

O modo de controlar externamente o quadrupolo é idêntico ao do Finnigan 1015 S/L, porém o sinal entre 0 e -10V seleciona m/z entre 0 e 1000 daltons.

2 Sistema de Dados do IQ-UNICAMP

Os espectrômetros MAT 311A e Finnigan 1015 S/L estão instalados em uma mesma sala, o que permitiu a utilização de uma única interface de aquisição de dados.

- INTERFACEAMENTO -

Cartão Interno

Interface Externa

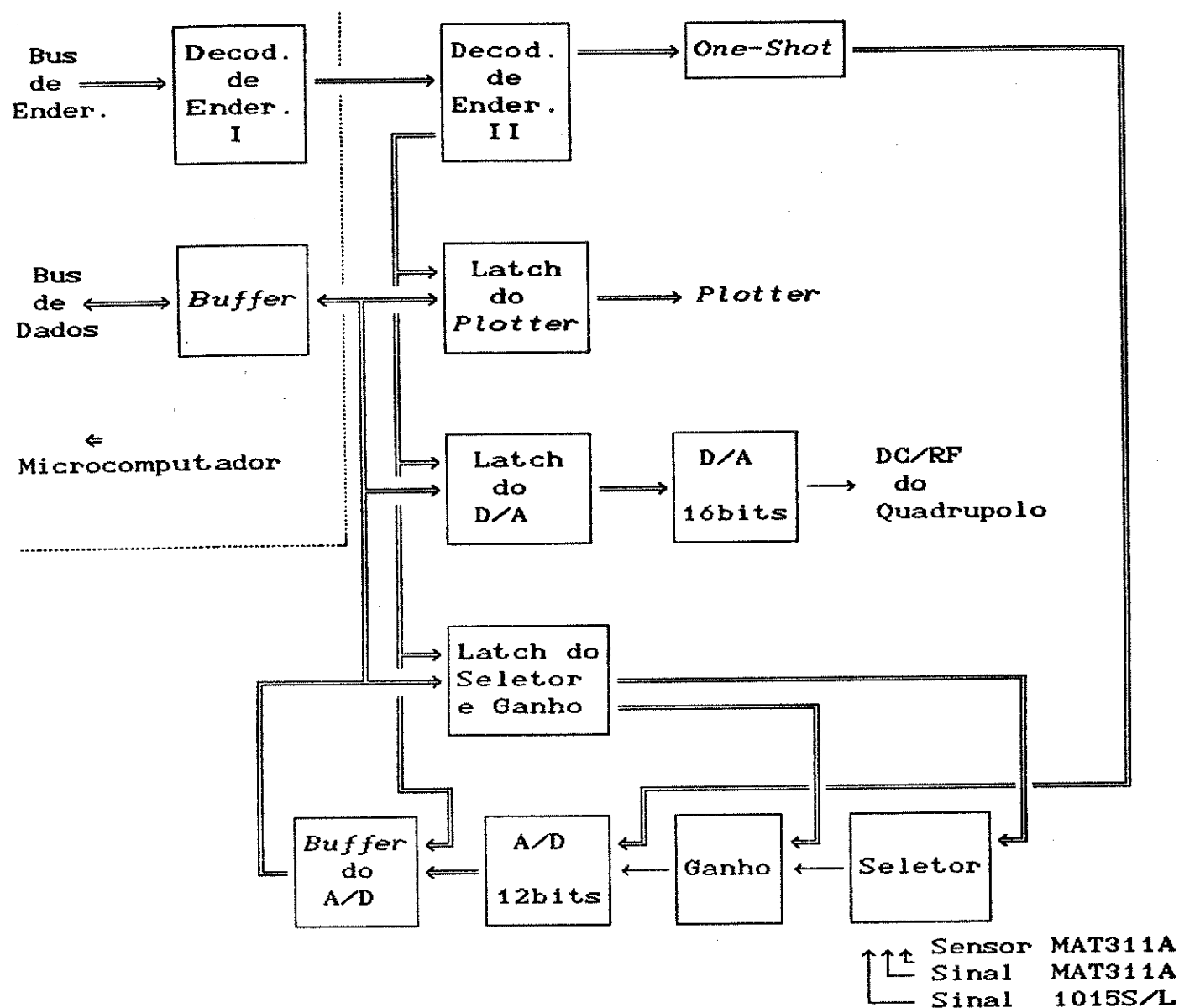


Figura 1 - Diagrama de Blocos da Interface do sistema de dados do IQ-UNICAMP. As linhas duplas representam o fluxo dos sinais digitais e as linhas simples, dos sinais analógicos.

2.1 Hardware

A figura 1 mostra o diagrama de blocos da interface. Esta é dividida em duas partes: um cartão interno ao microcomputador e um

- INTERFACEAMENTO -

conjunto de circuitos alojados em uma caixa de alumínio. O cartão interno contém parte da decodificação de endereços e um *buffer* para isolar o barramento de dados dos sinais externos. A decodificação de endereços é concluída externamente e controla vários dispositivos. Todos os circuitos externos são alimentados por uma fonte de 5V e $\pm 15V$ e a alimentação do seletor de entradas analógicas que requer -5V, é alimentado pela fonte do microcomputador.

A conversão digital/analógica não apresenta qualquer dificuldade, bastando enviar via *software* o valor desejado para o endereço do *latch*. Já o conversor analógico/digital requer uma sequência de comandos:

- ▶ Inicializar o conversor,
- ▶ Aguardar a conversão e
- ▶ Ler o valor presente na saída do conversor A/D.

O esquema da fonte de alimentação - que utiliza reguladores de tensão integrados da série 78XX e 79XX - é mostrado na figura 2. A relação dos componentes deste e dos outros circuitos se encontram no apêndice 1.

A figura 3 mostra o esquema do cartão interno. A filosofia de interfaceamento é tratar os diversos dispositivos como portas de entrada e saída (E/S). Para fazer a decodificação, são utilizadas as linhas de endereços de A0 a A9, as linhas de IOW e IOR - que são utilizadas para indicar se o sinal é de saída ou entrada - e a linha AEN que confirma se o endereço não está sendo utilizado por um processo de acesso direto a memória (DMA). O isolamento do barramento de dados é conseguido com o 74LS245.

A parte final da decodificação de endereços é conseguida através dos circuitos integrados U10 e U11, mostrados na figura 4. O temporizador U12 aumenta o tempo do pulso que controla a inicialização do conversor A/D. O *flip-flop* formado por U9C e U9D era utilizado por um integrador que filtrava o sinal de entrada do conversor A/D. Atualmente este integrador está desativado e o sinal analógico é introduzido diretamente na entrada deste

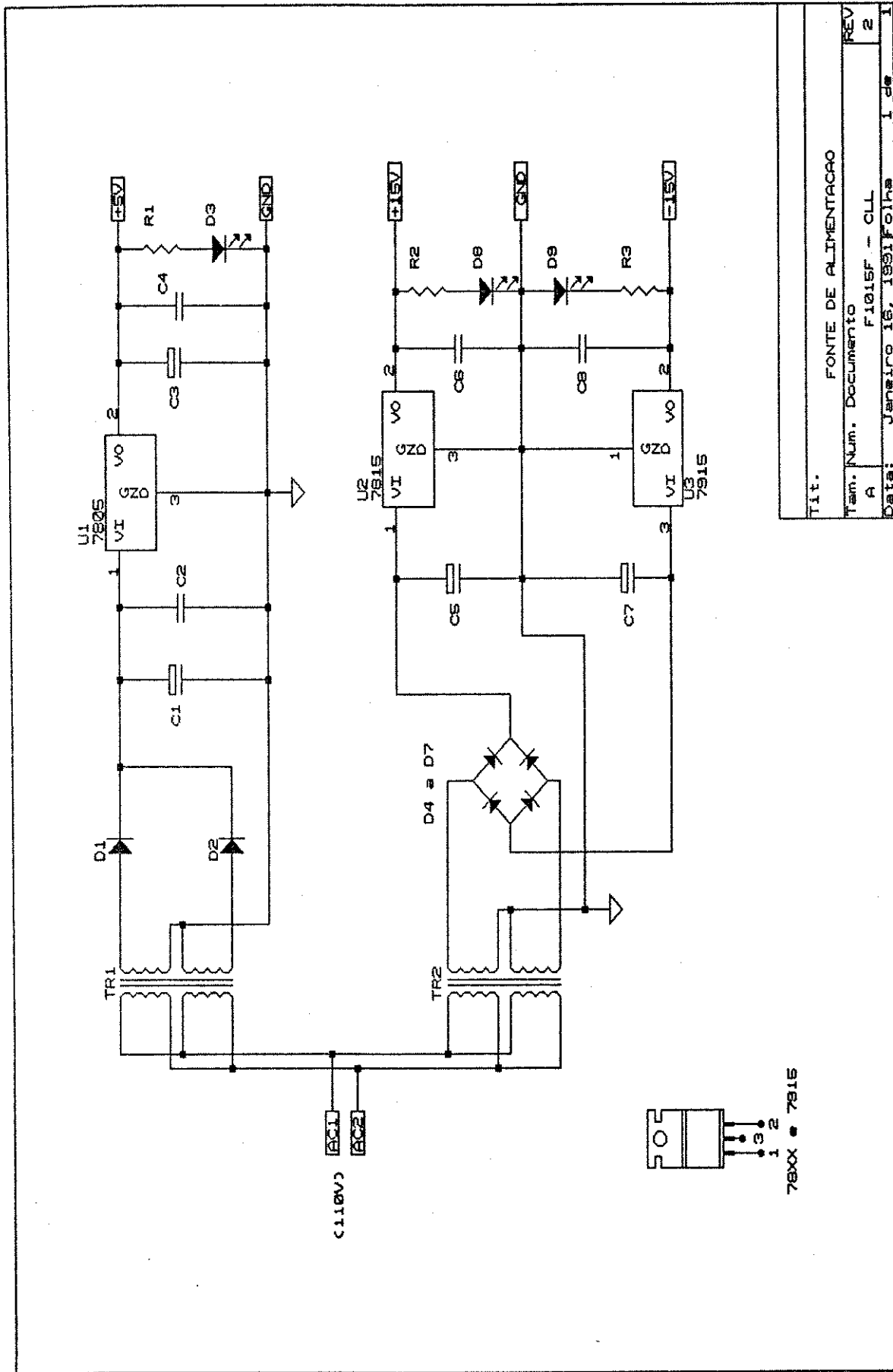


Figura 2

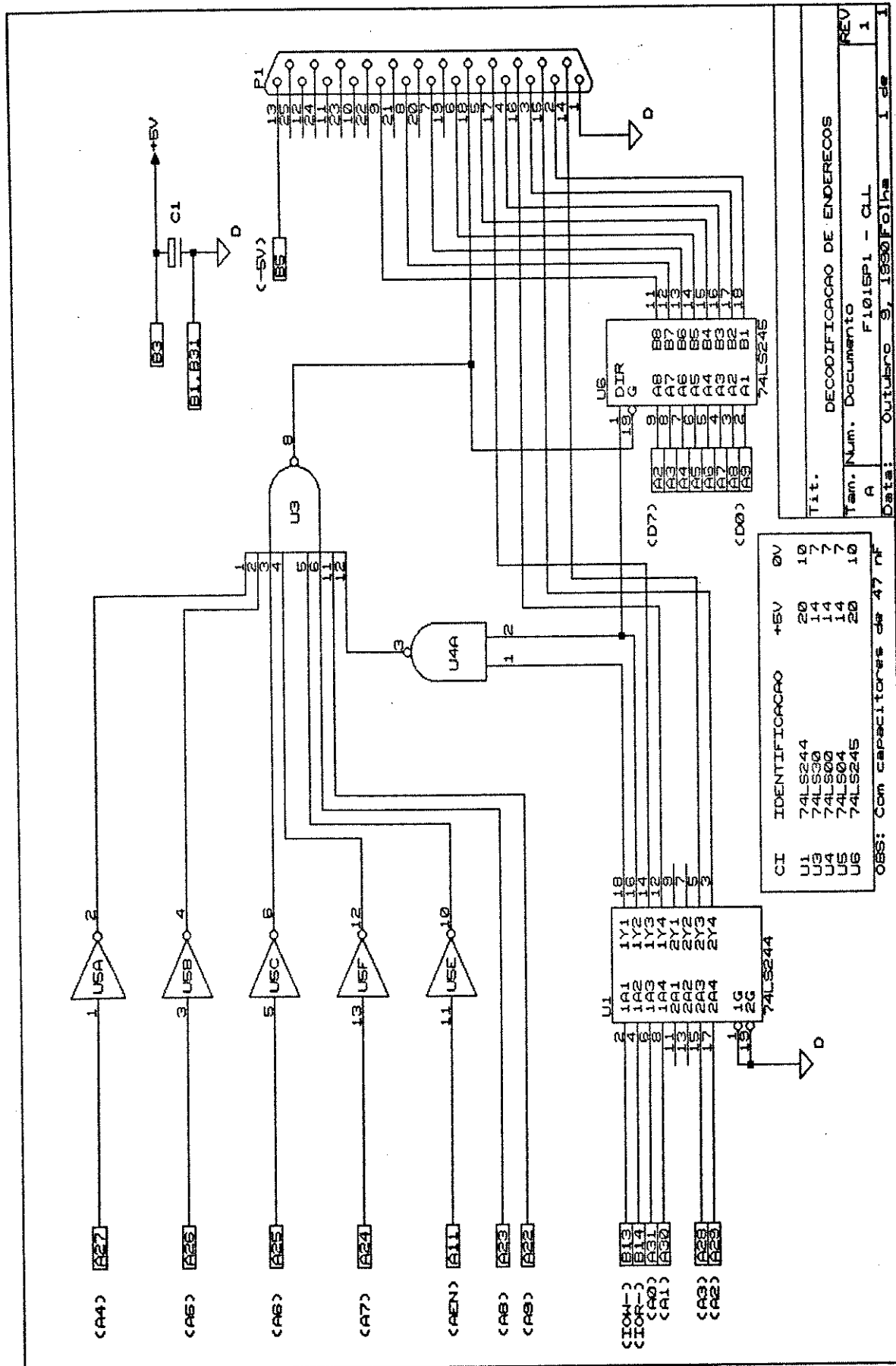


Figura 3

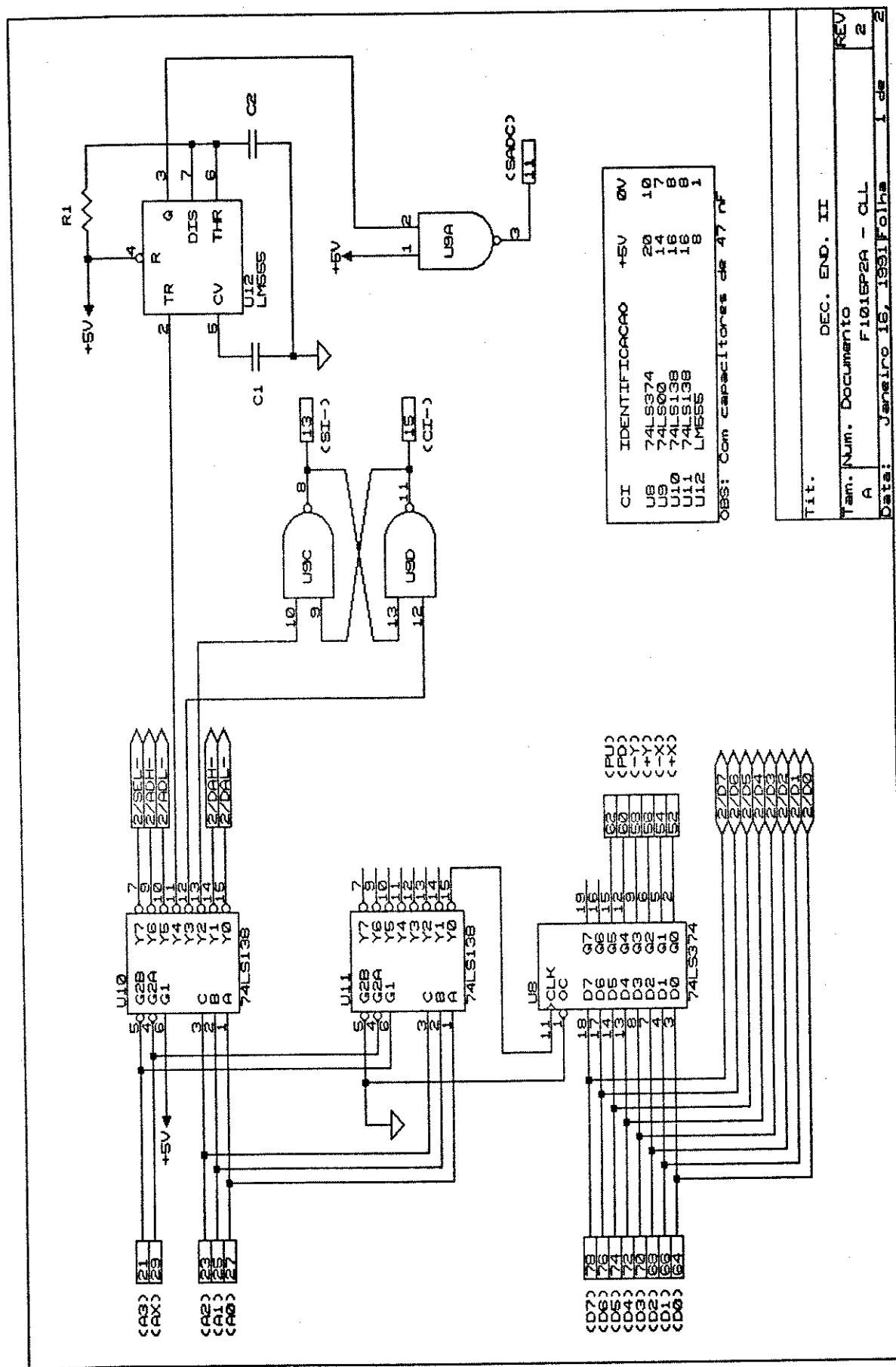


Figura 4

conversor. O *latch* U8 permite ao microcomputador controlar um *plotter* modelo 100 ZetaPlotter.

O conjunto de *buffers* e *latches* que controlam os conversores A/D e D/A é mostrado na figura 5. O acompanhamento da linha *store* (pino 50), permite ao programa determinar o final de uma conversão A/D.

A figura 6 mostra o esquema do circuito que permite a entrada de 8 sinais analógicos de modo diferencial e faz o controle de seu ganho, aproveitando a faixa dinâmica do conversor A/D. A configuração do amplificador formado por A1 e A2 é pouco comum e o potencial de saída é dado por

$$E_s = (E_+ - E_-) \left[1 + \frac{R_2(2R_1 + R_3)}{R_1 R_3} \right] \quad (1)$$

onde E_+ é o potencial no pino 3 de A2, E_- é o potencial no pino 3 de A1, $R_1 = R_{1a}$, $R_2 = R_{2a}$.

2.2 Sensor para Campo Magnético

Para controlar a corrente que alimenta a bobina geradora do campo magnético do espectrômetro MAT 311A, é utilizado um circuito apresentado esquematicamente na figura 7. O resistor é na verdade composto por um conjunto de duas bobinas de grandes dimensões em série, que melhoram a dissipação do calor e mantem constante o valor da resistência. A diferença de potencial sobre este resistor está relacionada com a corrente e assim é possível avaliar a intensidade do campo magnético. Devido à realimentação, o controle sobre a corrente é muito eficiente, porém com a histerese, o campo magnético não é precisamente controlado.

Para um acompanhamento da intensidade do campo e portanto do valor de m/z selecionado, é preciso utilizar um sensor que, posicionado no *gap* dos eletroímãs possua alguma propriedade que seja função da intensidade do campo magnético e possa ser

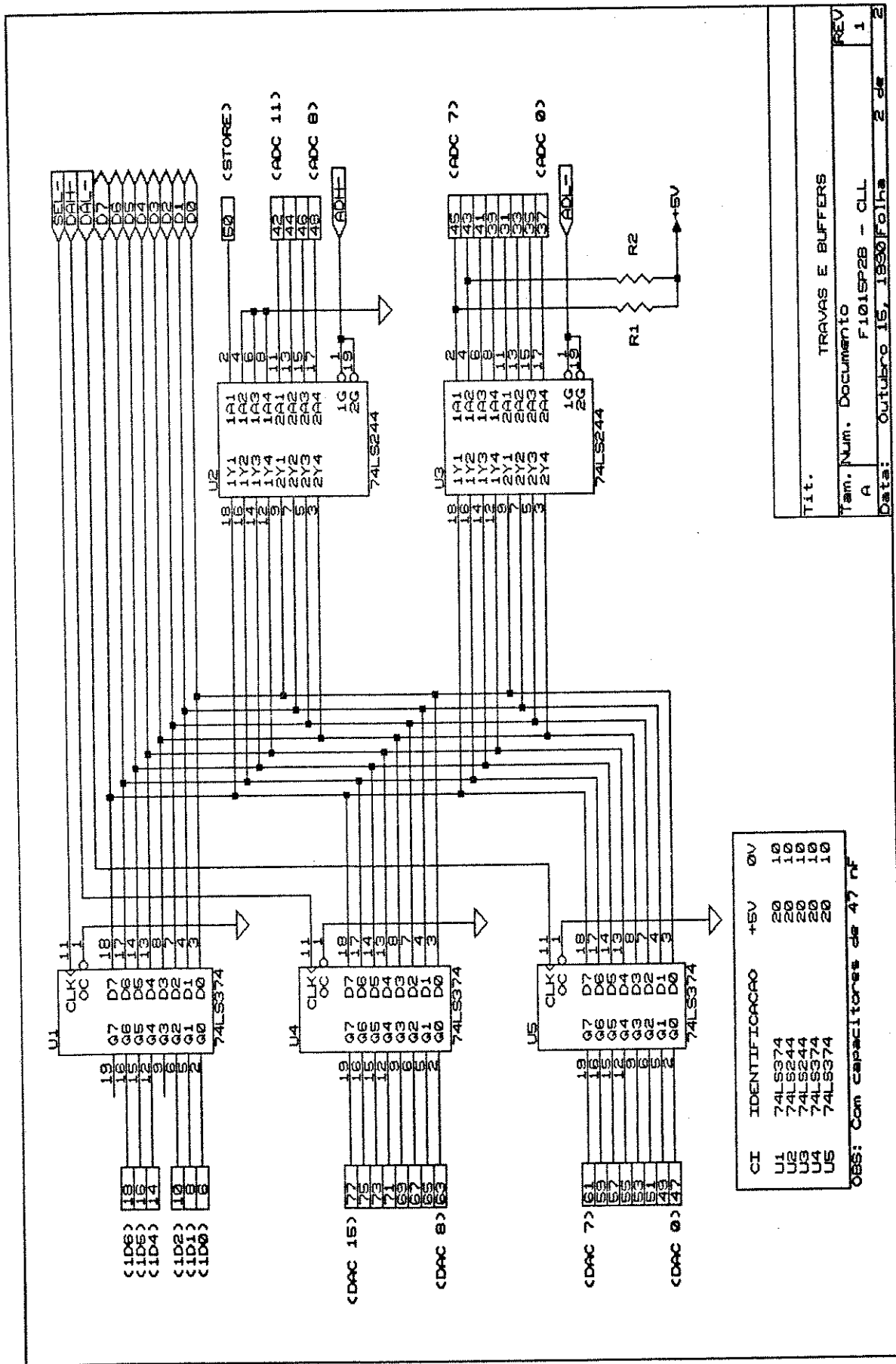


Figura 5

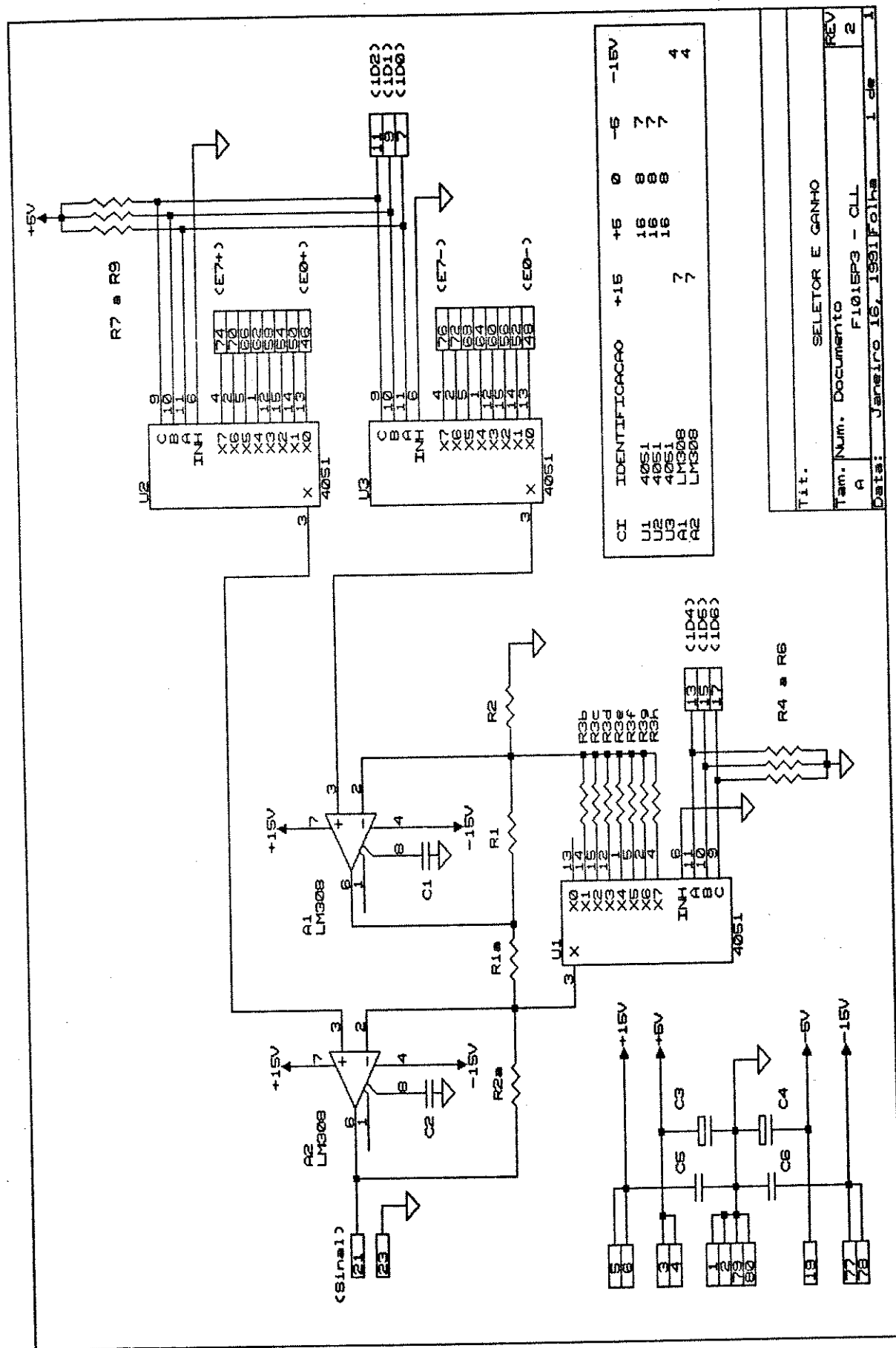


Figura 6

- INTERFACEAMENTO -

registrada eletronicamente. Os dispositivos mais utilizados com esta finalidade são baseados na magnetoresistividade ou no efeito Hall.

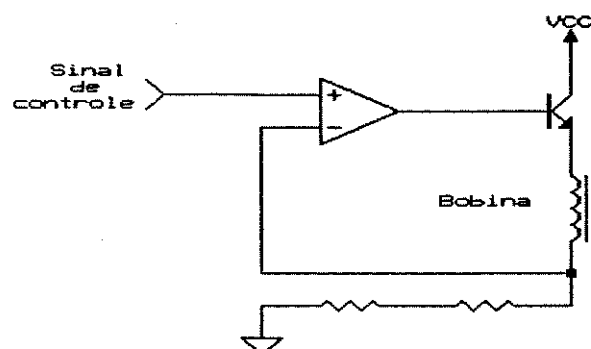


Figura 7 - Circuito de controle da corrente das bobinas do MAT311A. A estabilização é conseguida por realimentação do sinal obtido dos resistores especiais.

A magnetoresistividade se apresenta em materiais condutores quando submetidos a campos magnéticos intensos:

$$\frac{\Delta\rho}{\rho_0} \propto B_z^2 \quad (2)$$

onde ρ_0 é a resistividade do material quando o campo não é aplicado e $\Delta\rho$ é a variação da resistividade quando o material é submetido a um campo magnético B_z . Assim, com o aumento do campo, a resistividade na direção perpendicular a este aumenta [2]. Existem sensores comerciais para espectrometria de massas baseados neste princípio.

O efeito Hall está baseado na influência do campo magnético sobre a trajetória dos transportadores de carga de um material condutor ou semiconductor. Quando uma corrente atravessa um condutor que é submetido a um campo magnético (figura 8), os

- INTERFACEAMENTO -

transportadores de carga - elétrons ou "lacunas" - têm suas trajetórias alteradas de modo que a secção transversal apresenta uma variação de concentração destes transportadores. Este desequilíbrio de cargas gera uma diferença de potencial que é função da intensidade do campo magnético:

$$\mathcal{E}_y = R_H J_x H_z \quad (3)$$

onde $\mathcal{E}_y$ é o campo elétrico gerado na direção y, R_H é o coeficiente Hall, J_x é a densidade de corrente na direção x e H_z o campo magnético externo na direção z.

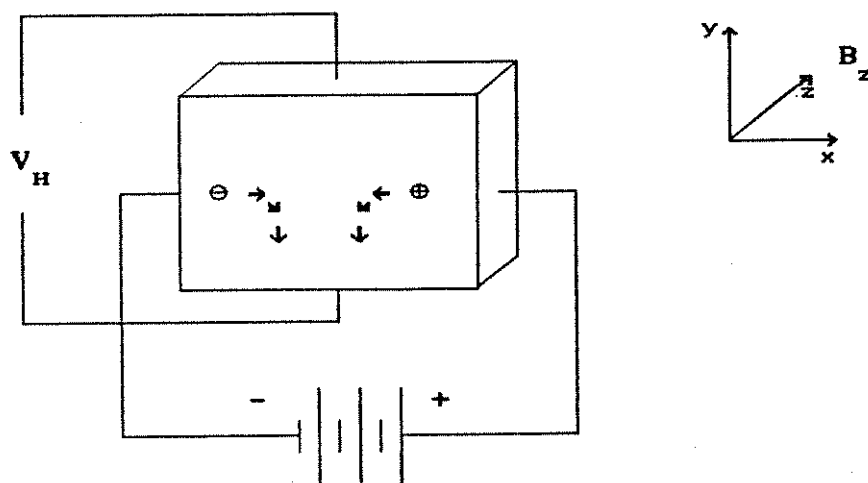


Figura 8 - Efeito Hall. A bateria conectada às extremidades do objeto condutor promove um fluxo de elétrons $\ominus$ ou lacunas $\oplus$ na direção do eixo x. O campo magnético B_z provoca a alteração da trajetória dos transportadores de carga, gerando uma diferença de potencial V_H que pode ser negativa ou positiva, dependendo do tipo de transportador de carga do material.

2.2.1 Princípio de Funcionamento

Assim como os metais, os semicondutores também apresentam o efeito Hall. Vários componentes eletrônicos semicondutores já foram utilizados como sensores. Transistores [3,4] e Diodos [5] comerciais assim como transistores MOS especiais [6] possuem suas

- INTERFACEAMENTO -

características alteradas quando submetidos a um campo magnético.

Embora seja possível fazer uma análise profunda sobre a influência do campo magnético sobre a junção PN dos dispositivos [7], os trabalhos das referências 3,4,5 procuram apenas uma relação empírica entre a intensidade do campo e a grandeza que estão medindo.

Como a temperatura possui efeito sobre as características dos semicondutores - sendo mais significativa que o efeito do campo magnético - todos os circuitos procuram compensá-la. Para isto utilizam dois componentes: um é o sensor propriamente dito, sendo submetido ao campo a ser avaliado e o outro deve estar fora da ação do campo mas a mesma temperatura que o primeiro.

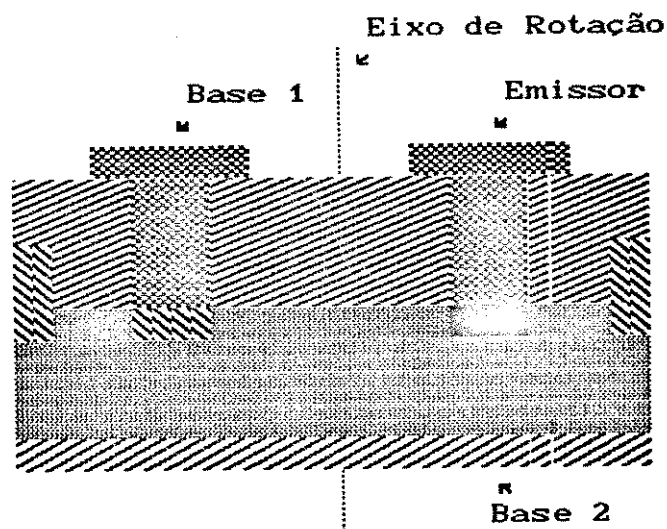


Figura 9 - Estrutura da construção anular do transistor unijunção [8]. As diversas regiões da pastilha são:

-  Silício tipo N de alta resistividade,
-  Óxido de silício,
-  Camada formada por difusão de Boro tipo P,
-  Contacto elétrico em alumínio,
-  Camada formada por difusão de fósforo tipo N e
-  Contacto elétrico em ouro.

Este trabalho está baseado no mesmo componente que Agrawal e Swami [4] utilizaram: o transistor unijunção 2N2646. Um transistor unijunção é composto basicamente por uma barra de silício tipo N

- INTERFACEAMENTO -

com dois contactos elétricos (bases 1 e 2) e um emissor tipo P implantado na lateral (figura 9).

A barra de silício é essencialmente um resistor. Como a junção PN está em posição intermediária, a barra funciona como um divisor de tensão, sendo que o potencial (V_{PNB1}) aplicado a junção é dado por

$$V_{PNB1} = \eta V_{B2B1} \quad (4)$$

onde V_{B2B1} é a diferença de potencial entre as bases e η é o fator *standoff*, que é intrínseco de cada componente. Quando a diferença de potencial emissor/base 1 (V_{EB1}) for menor que este valor acrescido da tensão direta do diodo (V_D), não haverá transporte de elétrons pela junção, pois esta estará inversamente polarizada. Para a junção ser diretamente polarizada, é necessário que V_{EB1} atinja o valor V_P dado por

$$V_P = V_D + \eta V_{B2B1} \quad (5)$$

Com o aumento da corrente elétrica entre a junção e a base 1, ocorre o aumento da condutividade. Devido a este fenômeno, a resistência entre junção e base diminui e, conseqüentemente, o potencial V_{PNB1} . Com a diminuição de V_{PNB1} , há o aumento de corrente e o processo se acentua até atingir um máximo.

A partir deste ponto, o componente entra na região conhecida como *resistência negativa*, pois com a diminuição de V_{EB1} , há um aumento de corrente pelo emissor para regenerar a situação anterior. Desta forma a junção PN não irá entrar em corte no valor de V_P , mas sim em um valor menor V_V .

A primeira etapa foi montar um circuito oscilador típico mostrado na figura 10 e instalar o transistor no *gap* dos eletroímãs do espectrômetro de massas MAT311A. O princípio de funcionamento é simples. Quando o capacitor está descarregado, o potencial no emissor é próximo de zero e portanto, menor que os

- INTERFACEAMENTO -

potenciais das bases. A medida que o capacitor se carrega, o potencial se eleva até atingir o valor de V_P quando a junção emissor/base 1 passa a conduzir, entrando na região de resistência negativa. Com a descarga do capacitor através de R_{B1} , o potencial do emissor abaixa até atingir o valor de V_V quando entra em corte. Com o fim da fuga de corrente pelo emissor, o capacitor volta a se carregar, repetindo o processo.

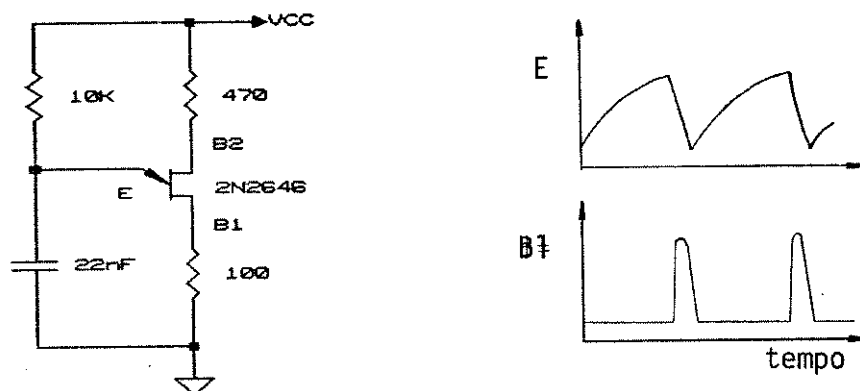


Figura 10 - Oscilador típico utilizando um transistor unijunção 2N2646. A resposta tanto no emissor como na base 1 é mostrada ao lado.

A primeira observação feita é que a posição do componente em relação às linhas de força do campo é de fundamental importância para se observar variações nas tensões V_P e V_V e, portanto, na frequência de oscilação. O transistor deve ser posicionado com os terminais perpendiculares às linhas de campo e com a rotação do componente, se atinge duas posições de maior sensibilidade. A figura 9 mostra o eixo de rotação da pastilha. A corrente de elétrons vai da base 2 para a base 1 e durante a descarga do capacitor, também para o emissor.

Comparando a sensibilidade deste circuito com a sensibilidade de uma simples junção PN, observa-se uma vantagem razoável: quando

- INTERFACEAMENTO -

submetido a mesma intensidade de campo, o desvio de V_p do oscilador é aproximadamente 5 vezes superior ao desvio da tensão direta de um diodo. Por outro lado o coeficiente térmico também é maior: aproximadamente $8\text{mV}/^\circ\text{C}$ contra $2\text{mV}/^\circ\text{C}$ de um diodo de silício comum. Isto se deve a influência da resistência interbases do transistor unijunção na resposta do oscilador. Sob este ponto de vista, a utilização do transistor unijunção não parece tão vantajosa, porém, do modo como o circuito para compensação de temperatura foi idealizado, o desempenho global melhora consideravelmente.

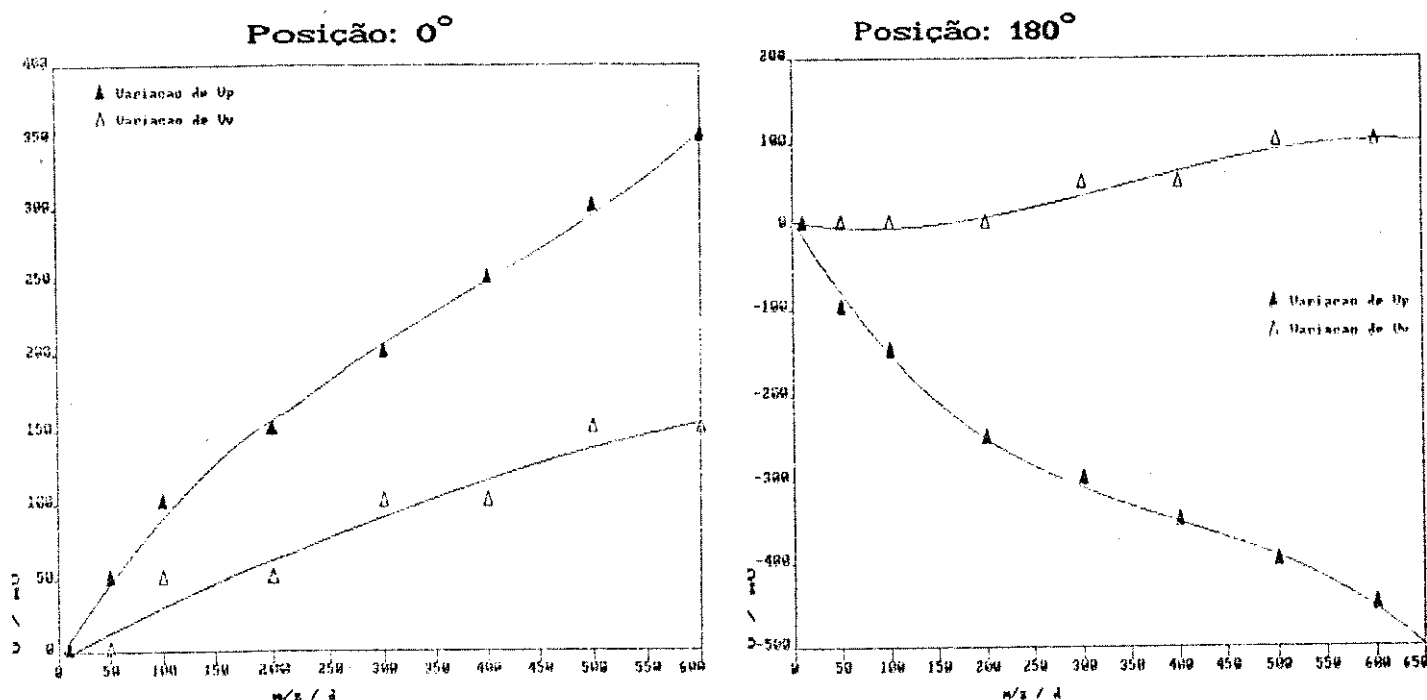


Figura 11 - Variação de V_p e V_v (em milivolts) em função do valor de m/z (em daltons). Os diagramas correspondem às medidas efetuadas para os dois máximos de sensibilidade defasados em 180° graus.

Aproveitando-se do fato da resposta de V_p ter dois máximos de sensibilidade em função da rotação (figura 11), sendo um positivo e outro negativo, o circuito apresentado na figura 12 foi projetado. Os dois transistores dos osciladores 1 e 2 são posicionados no *gap* do eletroímã de forma a darem o máximo de

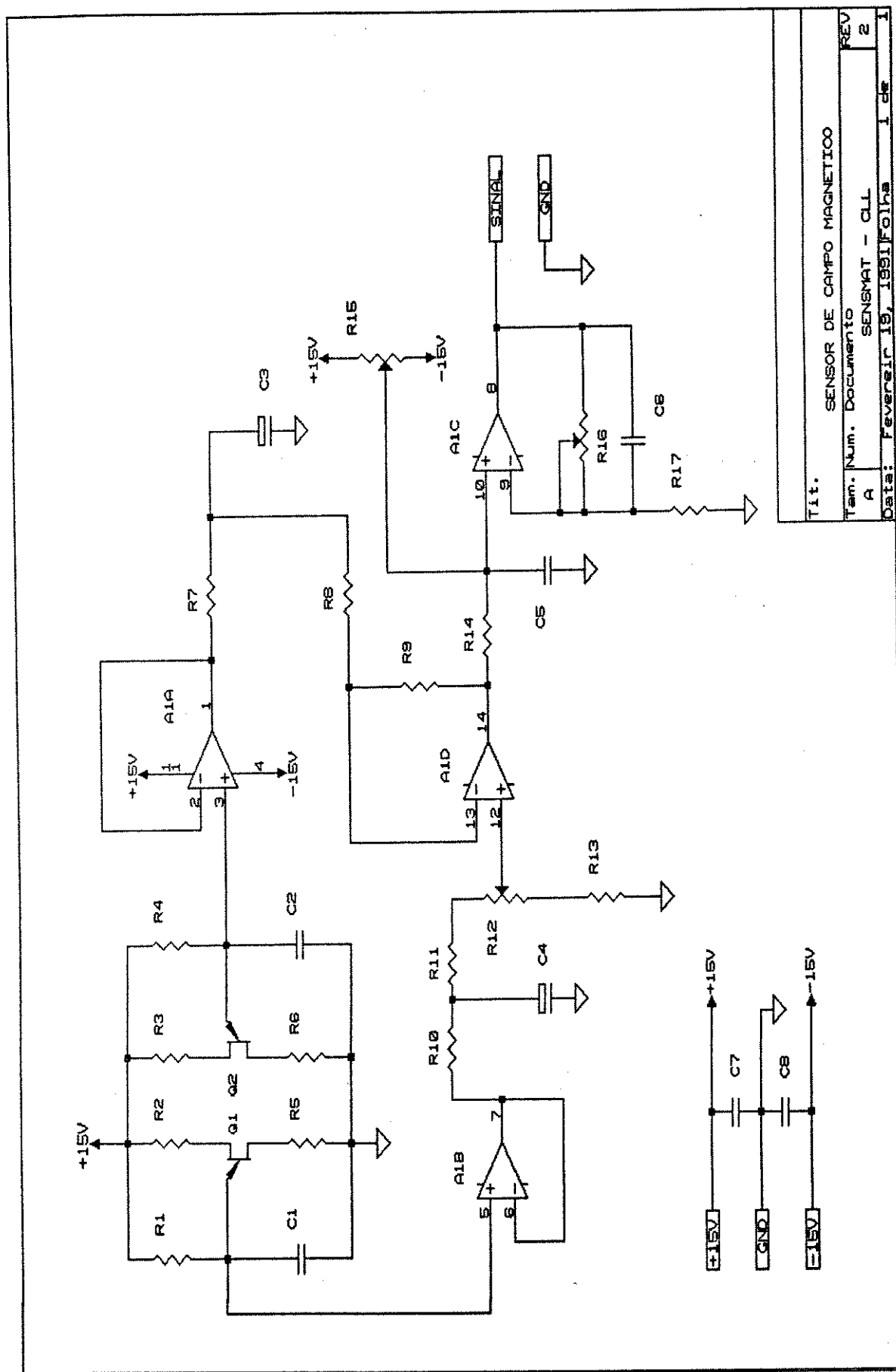


Figura 12

- INTERFACEAMENTO -

sensibilidade, porém com polaridades contrárias. Os buffers A1A e A1B isolam os osciladores do filtro RC, que fornecem o valor médio da amplitude de oscilação. Através do amplificador diferencial A1D, os valores médios são subtraídos além de dar um ganho de aproximadamente dois. O amplificador A1C desconta a tensão de *offset*, dá um ganho de até 5 vezes e filtra novamente o sinal. Como os transistores estão submetidos ao mesmo ambiente, os desvios com a temperatura são praticamente os mesmos e se anulam após a subtração dos sinais. Já a resposta em função do campo aplicado é praticamente dobrada após a etapa de subtração, pois os sinais têm polaridades contrárias.

Para posicionar os transistores firmemente no espectrômetro e permitir a rotação do mesmo, foram construídos dois suportes em nylon mostrado na figura 13. Como o invólucro metálico do transistor 2N2646 é um bom isolante do campo magnético [9], este foi removido após preencher o seu interior com resina epóxi para proteger os frágeis filamentos internos.

2.2.2 Interpretação do Efeito do Campo sobre V_p e V_v

A análise do efeito do campo magnético sobre o transistor unijunção é complexa. A medida da resistência interbases em função do campo magnético mostra que a pastilha de silício sofre um considerável efeito da magnetoresistividade. Foi observado que a variação da resistência independe do ângulo de rotação da pastilha, o que era de se esperar. Como o efeito Hall sobre a junção PN pode variar com o ângulo de rotação, não alterando, porém, a polaridade de $\partial V_D / \partial H$, não há uma explicação direta para o comportamento de V_p (figura 11).

A equação 5 mostra que V_p é função de V_D e η . Como o efeito sobre V_D não justifica a intensidade e inversão da polaridade da resposta, o efeito do campo magnético sobre η é o mais importante. Observando a estrutura da pastilha mostrada na figura 9, pode-se supor que os máximos de intensidade de resposta ocorram quando o campo aplicado é perpendicular ao plano do papel. A trajetória dos

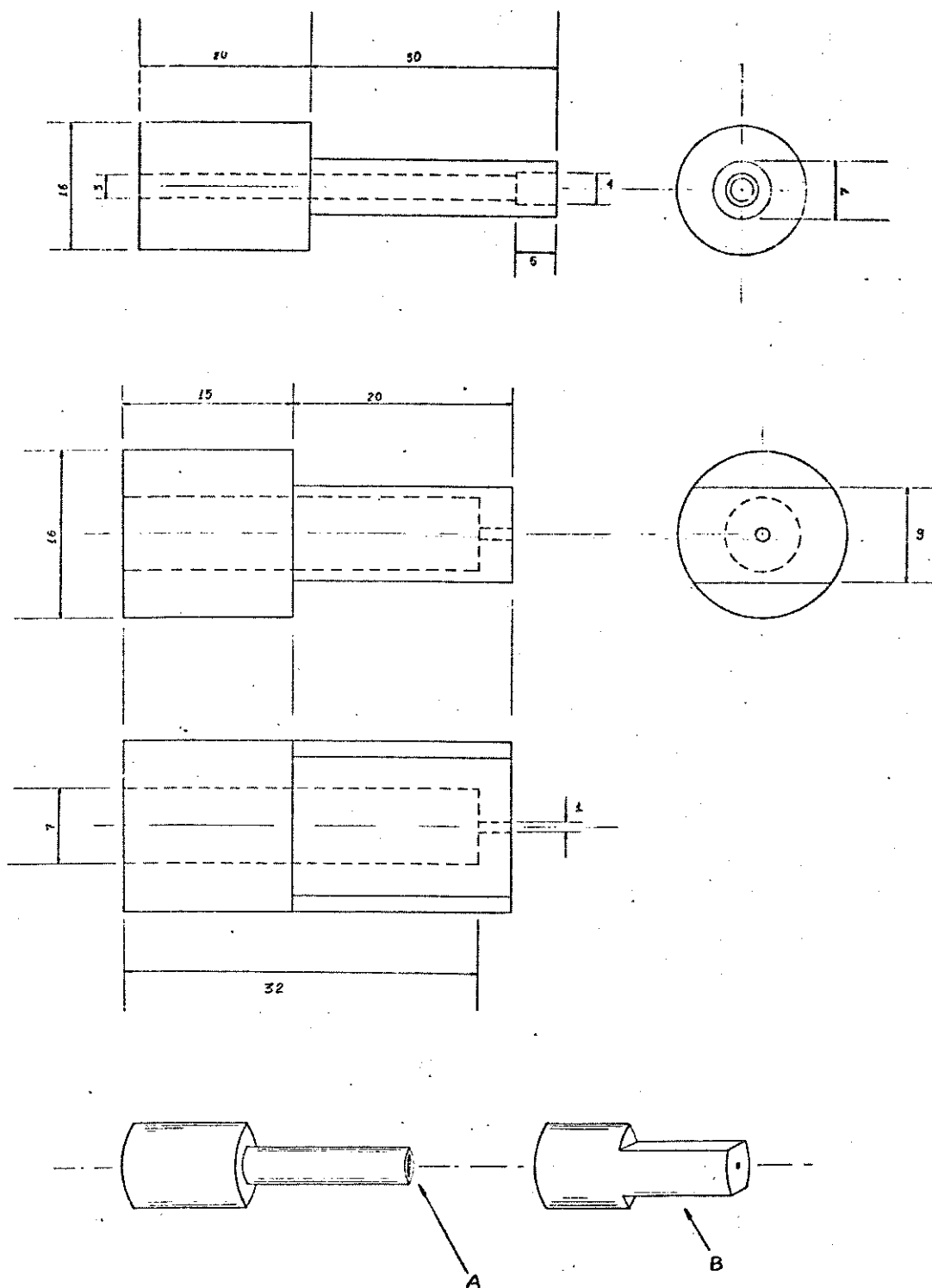


Figura 13 - Suporte para o sensor de campo magnético. As medidas são dadas em milímetros. O transistor unijunção é colocado no compartimento A. Com o encaixe B, o conjunto é posicionado no eletroímã.

- INTERFACEAMENTO -

elétrons da base 2 para a base 1 é alterada, podendo se aproximar ou distanciar do emissor, dependendo do sentido do campo. Certamente isto altera a relação dos potenciais entre junção/base 1 e interbases e, portanto, o η do transistor. Este efeito depende do sentido do campo magnético aplicado.

O comportamento de V_v em função do campo está de acordo com o esperado para o caso de se considerar o componente como um diodo de silício. Segundo Bansal e Parshad [3] a corrente direta de um diodo diminui com o aumento do campo, assim, V_D deve aumentar. Como antes de atingir o valor de V_v a junção PN está conduzindo, a corrente de elétrons entre emissor e base 1 deve ser a mais importante e, assim, o efeito da variação da trajetória interbases não é significativo.

Uma interpretação quantitativa de todo o processo somente seria possível com o conhecimento exato da estrutura da pastilha, seu posicionamento em relação ao campo e um estudo preciso da variação das características do transistor em função do campo magnético.

2.2.3 Avaliação do Sensor

A primeira etapa de ajustes do circuito foi determinar os coeficientes térmicos dos osciladores para dimensionar o valor de resistência do *trimpot* na entrada de A_3 . Este resistor é responsável pela compensação da diferença de CT dos osciladores. Para fazer este ajuste, os eixos centrais do sensor - contendo os transistores - são introduzidos em outro suporte feito também em nylon (figura 14). Este suporte possui um enrolamento externo com fio de níquel-cromo responsável pelo aquecimento do conjunto.

Devido a amplificação do sinal, seria esperado um CT para o circuito total da ordem de $40\text{mV}/^\circ\text{C}$. O melhor ajuste conseguido diminui o CT para $3\text{mV}/^\circ\text{C}$. Como a resposta do circuito dá em média 10mV de intervalo entre duas unidades de massa, uma variação de 1°C deve causar um erro menor que 0,5 daltons. Na prática, se observou que, devido a refrigeração da sala e do eletroímã, a

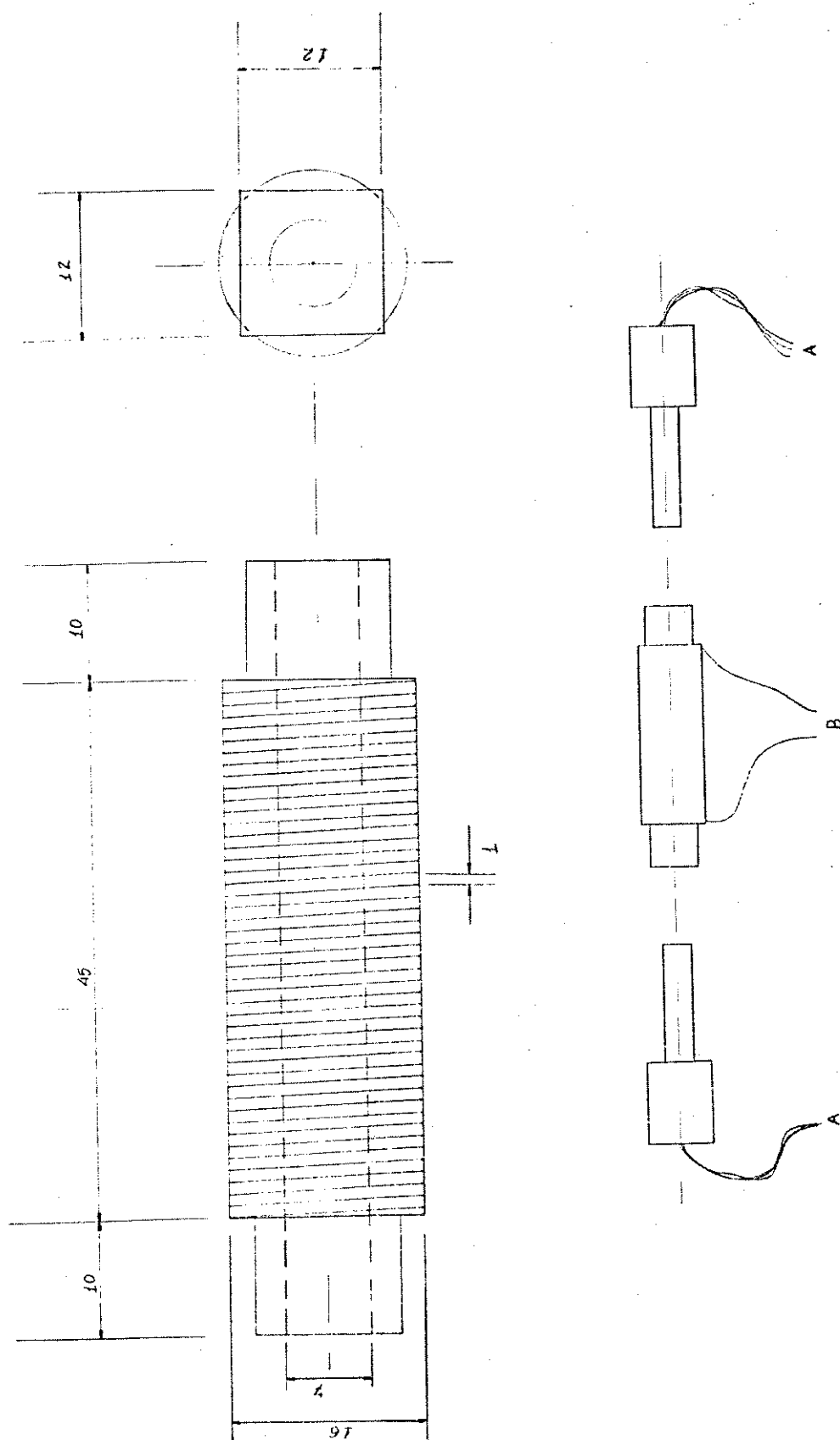


Figura 14 - Calibrador para o sensor de campo magnético. As medidas são dadas em milímetros. Os terminais do enrolamento de aquecimento (B) são conectados a uma fonte ajustável de tensão, aquecendo igualmente os transistores que estão conectados aos osciladores pelos terminais (A).

- INTERFACEAMENTO -

variação de temperatura é bem menor que 1°C .

O fato dos dois transistores serem posicionados próximos diminui um inconveniente presente nos outros circuitos [3,4,5]: como o sensor de referência não pode ser submetido ao campo magnético, a distância entre os sensores deve ser considerável, o que implica em temperaturas ambientes diferentes.

Procedimento	Leitura para 78d
Primeira leitura	724mV
Varredura até 5d	724mV
Varredura até 600d	724mV
Intervalo de 10min	726mV
Varredura até 600d	725mV
Varredura até 5d	725mV
Varredura até 600d	726mV
Varredura até 1d	726mV
Intervalo de 20min	724mV
Varredura até 1d	726mV

Tabela 1 - Teste de estabilidade do sensor para campo magnético. Para verificar a independência da histerese, foram feitas varreduras de 78d aos valores indicados e após o retorno ao pico em 78d, foram feitas as leituras. O valor médio é 725mV e a estimativa do desvio padrão é 0,94mV.

O desempenho estático do sensor foi avaliado introduzindo uma amostra de benzeno no espectrômetro de massas e acompanhando o sinal do circuito com um multímetro de quatro dígitos e meio. O circuito foi calibrado de forma a responder de 0 a 1,3V para a região de 20 a 150 daltons. A resposta inicial para o pico do íon molecular em 78d foi 724mV. A tabela 1 mostra os resultados obtidos após uma série de procedimentos para verificar a estabilidade do sensor. Se for considerado que o intervalo entre 77 e 78d corresponde a 8mV, pode-se verificar que, para um

- INTERFACEAMENTO -

intervalo de 95% de confiança, a flutuação do sinal ($\pm 1,88\text{mV}$) é menor que $\pm 0,25d$.

2.3 Software

Como foram configuradas como portas de E/S, as funções da interface são acessadas por instruções específicas para esta finalidade. Na linguagem Pascal, na qual foram desenvolvidos todos os programas do sistema de dados, isto é conseguido através dos vetores **Port** e **PortW**. A tabela 2 mostra os índices do vetor **Port** - que são iguais aos endereços de E/S - utilizados e suas respectivas funções.

Porta	Função
300H	Byte menos significativo do conversor D/A
301H	Byte mais significativo do conversor D/A
302H	Inicia integrador (desativado)
303H	Limpa integrador (desativado)
304H	Inicia conversor A/D
305H	Byte menos significativo do conversor A/D
306H	Byte mais significativo do conversor A/D
307H	bit 0 a 3 - Seletor de entradas bit 4 a 7 - Ganho de entrada

Tabela 2 - Endereços das portas utilizadas pela interface do sistema de dados do IQ-UNICAMP.

O conversor D/A é atualizado usando-se o vetor **PortW**, pois assim os dois endereços consecutivos são acessados em um intervalo de tempo bem menor, além de simplificar a programação.

A conversão A/D é inicializada com uma instrução de leitura ou escritura no endereço 304H. Como a conversão é mais lenta que os processos do microcomputador, é necessário o acompanhamento do bit 7 da porta 306H, que se mantém em nível lógico alto enquanto a conversão não termina. Após o fim da conversão, o valor fica disponível para leitura a qualquer momento, perdendo-se somente quando uma nova conversão for iniciada.

Como a seleção de canal de entrada e ganho é feita em um

- INTERFACEAMENTO -

único endereço, é necessário manter um byte de memória com a imagem da última informação enviada a porta. Este valor armazenado deve ser manipulado toda vez que for desejado acessar a referida porta, permitindo trabalhar individualmente cada bit. A tabela 3 lista os dispositivos conectados aos canais de entrada e a tabela 4 os valores nominais de ganho.

Canal	Identificação
0	Campo Magnético do MAT311A
1	Multiplicador de elétrons do MAT311A
2	Multiplicador de elétrons do 1015S/L

Tabela 3 - Lista dos canais de entrada. Os canais de 3 a 7 estão disponíveis.

Valor enviado	Ganho Aproximado
0	2
1	4
2	6
3	12
4	22
5	42
6	80
7	163

Tabela 4 - Correspondência entre o valor enviado para os quatro bits mais significativos da porta 307H e o ganho para o sinal de entrada.

Para visualizar os sinais adquiridos pelo conversor A/D foi criado o programa MONITOR. O canal, assim como o ganho podem ser selecionados a qualquer instante e o conjunto de dados armazenados em um vetor é apresentado ao final de cada sequência de aquisição. O programa é útil para avaliar as condições do sinal de entrada, assim como selecionar o valor de ganho mais adequado para ser utilizado em outros programas.

- INTERFACEAMENTO -

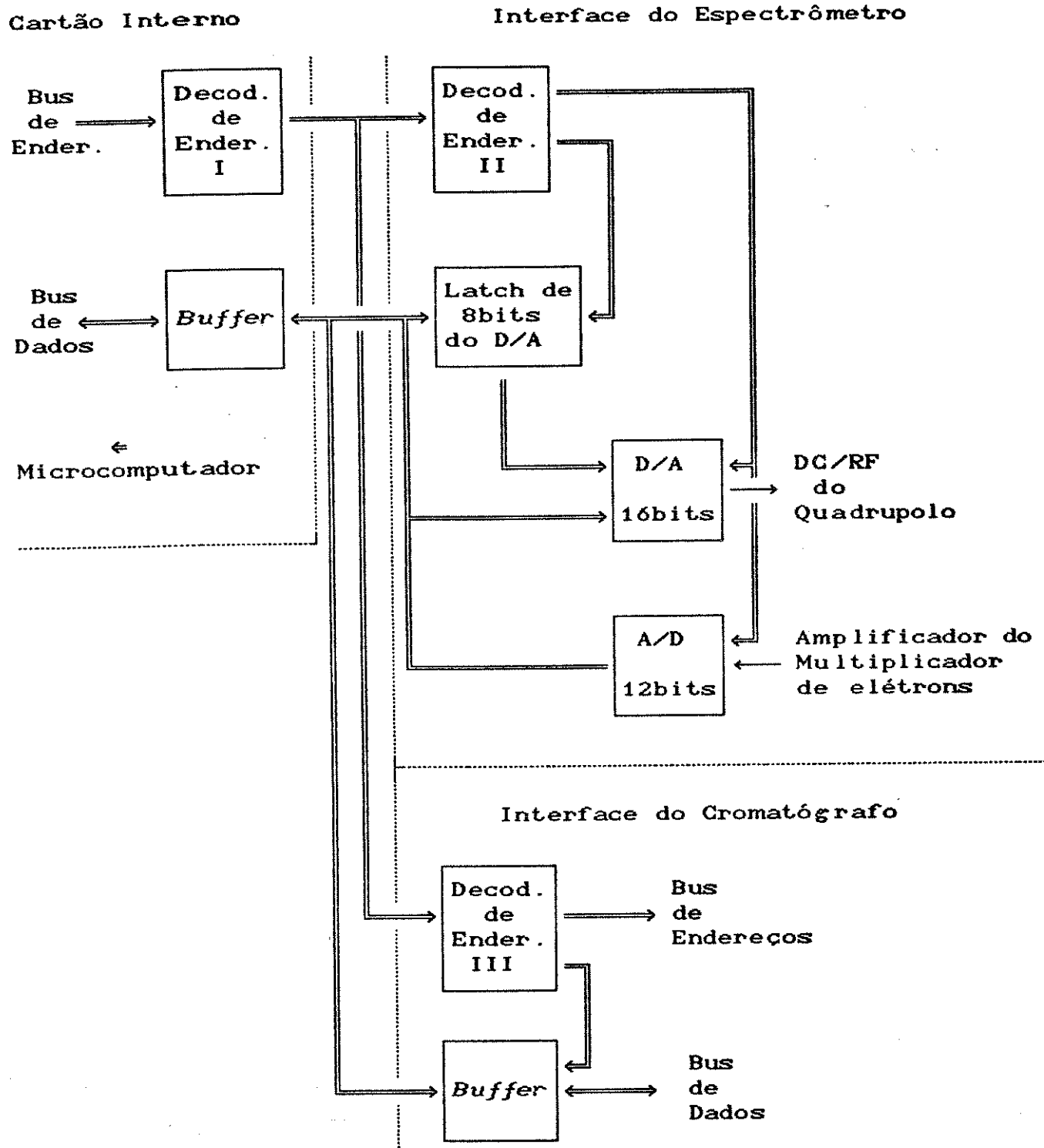


Figura 15 - Diagrama de Blocos da Interface do sistema de dados do CCEN-UFPa. As linhas duplas representam o fluxo dos sinais digitais e as linhas simples, dos sinais analógicos.

3 Sistema de Dados do CCEN-UFPA

Este sistema é mais recente e, embora possua características bastante diferentes do outro sistema, o princípio de funcionamento é o mesmo. Os dispositivos são tratados como portas de E/S e a transmissão dos dados é feita de modo paralelo.

3.1 Hardware

O diagrama de blocos da interface é mostrado na figura 15. O sistema é composto por 3 módulos: um cartão interno ao microcomputador com a decodificação parcial de endereços, um cartão controlador que substitui o microprocessador do cromatógrafo e um módulo com conversores A/D e D/A para aquisição de dados do espectrômetro de massas. Este último módulo requer fonte de alimentação própria mostrada na figura 16.

A primeira parte da decodificação de endereços é feita no interior do microcomputador. Como pode ser observado na figura 17, o princípio de funcionamento deste cartão é semelhante ao utilizado pelo sistema de dados da UNICAMP.

O controle do cromatógrafo - mostrado na figura 18 - é bastante simples utilizando somente 3 circuitos integrados e poucos componentes passivos. A finalidade dos *buffers* U1 e U3 é proteger e isolar os sinais do microcomputador e cromatógrafo. O CI U2 faz a decodificação final dos endereços. Desta forma o processamento que era executado por um microprocessador 8008, passou para o 8088 do microcomputador.

A aquisição de dados do Espectrômetro de Massas é mais complexa devido a necessidade da interface analógica/digital. A figura 19 mostra a decodificação final de endereços que controlam os conversores A/D e D/A.

O cartão do conversor D/A possui um *latch* de 16 bits que devem ser atualizados simultaneamente. Como o *bus* de dados do microcomputador é de 8 bits, este é enviado em duas vezes: o *byte* menos significativo é enviado na primeira etapa e armazenado em U2, o *latch* do D/A é liberada para entrada de dados juntamente com

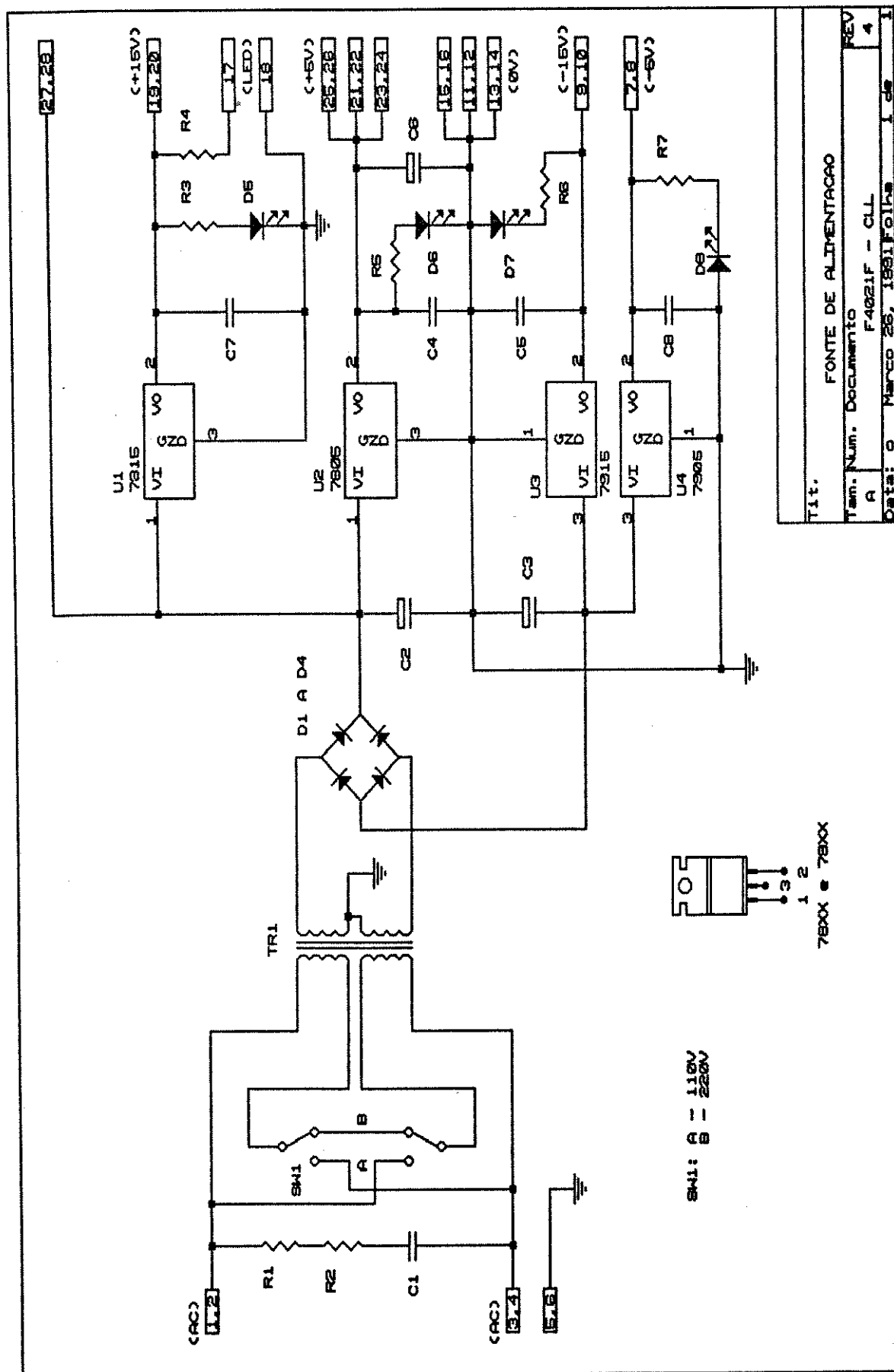


Figura 10

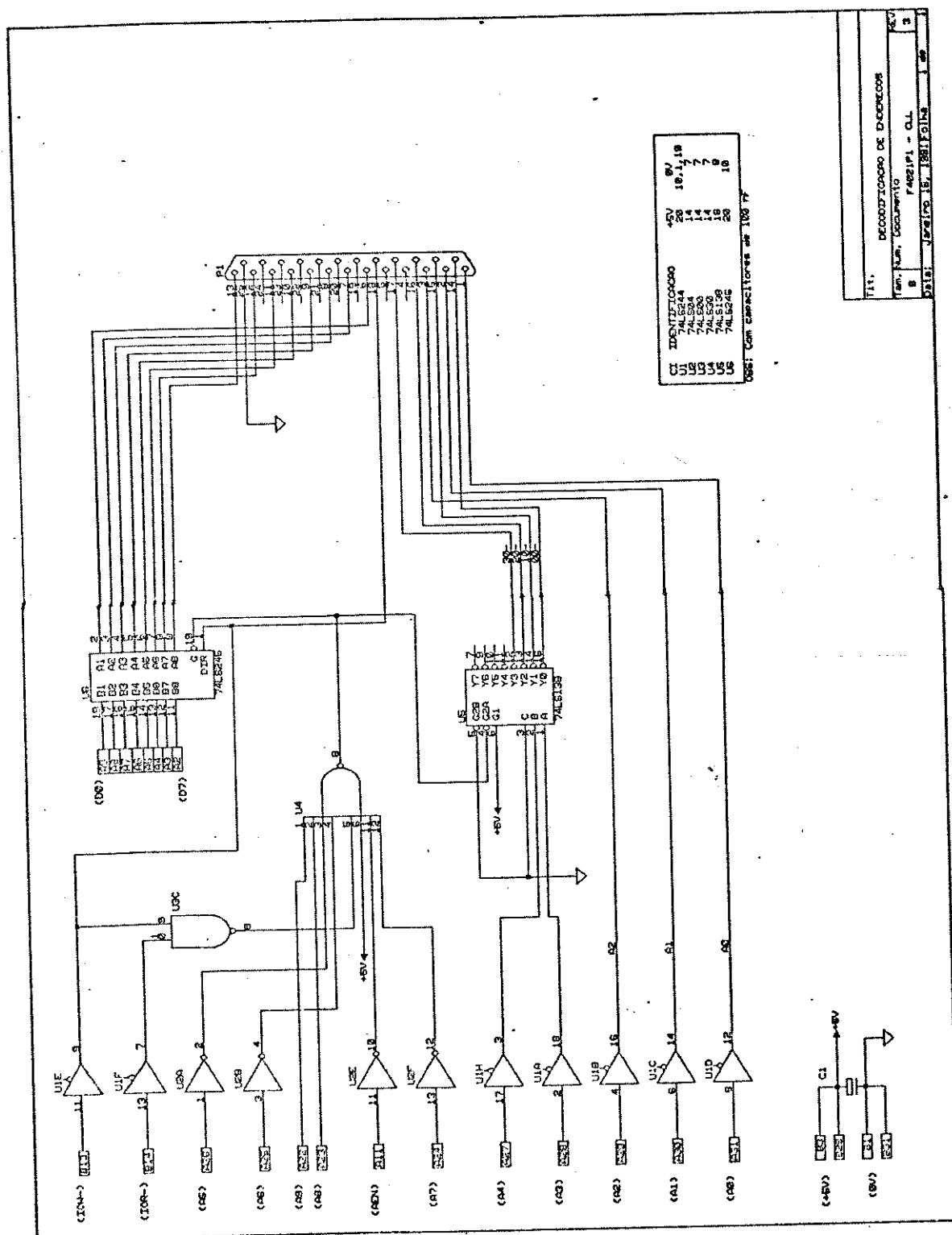


Figura 17

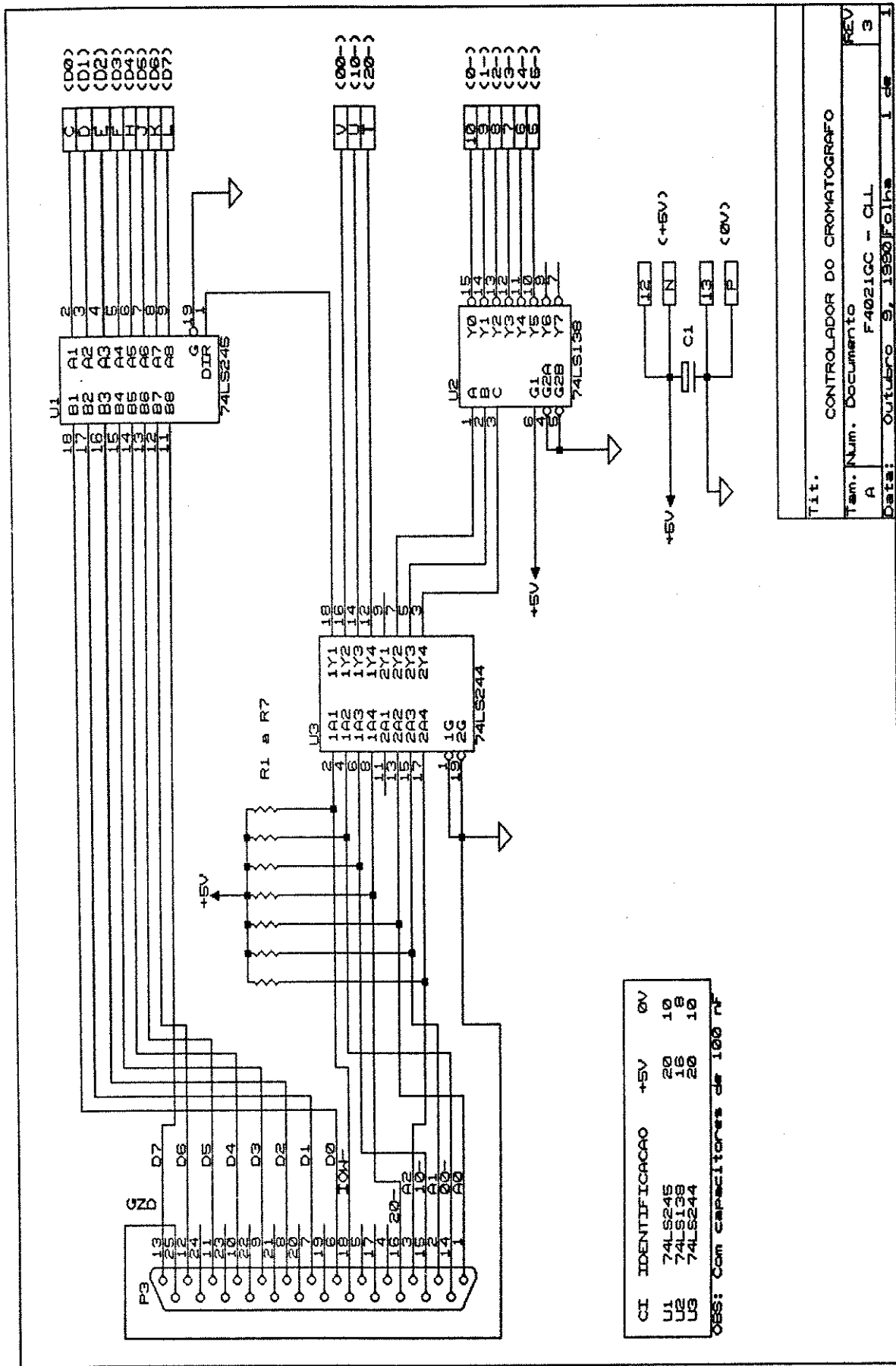


Figura 18

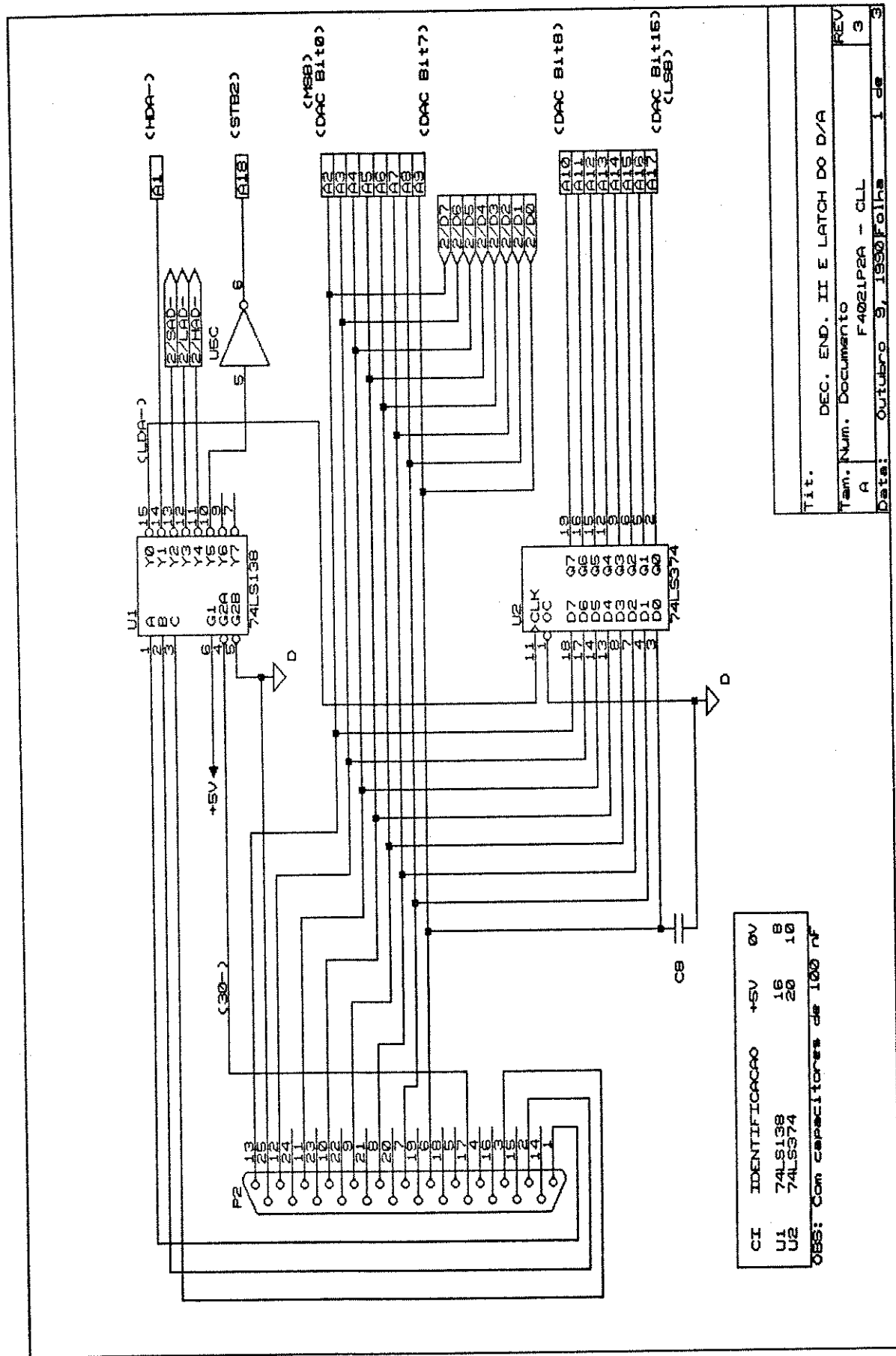


Figura 19

Tit.		DEC. END. II E LATCH DO D/A	
Tam. Num.	Documento	F4021P2A - CLL	
A		3	
Date:	Outubro 9, 1990	Folha	1 de 3

o envio do byte mais significativo. Após a conversão, um sinal entre 0 e -10V é gerado correspondendo respectivamente ao valor enviado entre 0 e 65535.

3.2 Conversor Analógico/Digital

O cartão do conversor A/D do antigo sistema de dados não foi aproveitado, pois o circuito integrado responsável pela conversão propriamente dita foi danificado e não estava disponível no mercado, assim como outro conversor de 12 bits com velocidade adequada. Resolvemos desenvolver um novo conversor A/D utilizando componentes de fácil aquisição no mercado nacional.

3.2.1 Princípio de Funcionamento

Algumas características específicas do processo de aquisição de dados de GC/MS foram levadas em consideração no desenvolvimento deste modelo.

Um espectro de massas possui sinais sempre positivos, assim o novo conversor foi desenvolvido na forma unipolar, embora seja possível somar uma tensão de referência na entrada tornando-o bipolar.

Devido a necessidade de grande amplificação do sinal correspondente aos íons que atingem o multiplicador de elétrons, o nível de ruído é bastante intenso. Assim, quando se deseja uma medida precisa deste sinal, deve-se integrá-lo durante um intervalo de tempo razoável afim de eliminar este ruído. Nem sempre isto é possível, principalmente durante a aquisição de espectros no processo cromatográfico.

Desta forma, os espectros possuem intensidades pouco precisas e, a princípio, não deveriam requerer um conversor de alta resolução. Porém, sempre são utilizados conversores com pelo menos 12 bits. Isto se torna necessário pois a intensidade do espectro varia consideravelmente durante o cromatograma, além de possuir picos de baixa intensidade que, às vezes, podem ser importantes na elucidação da estrutura.

- INTERFACEAMENTO -

Outro aspecto importante é que, para o sistema criado, não é fundamental que o conversor possua extrema estabilidade térmica, pois, além de trabalhar em ambiente com temperatura controlada, o espectro é normalizado com relação ao pico mais intenso. Assim, a variação de temperatura será pequena, pois o intervalo de tempo é de no máximo alguns segundos. A única situação onde o intervalo de tempo pode chegar a minutos ou horas, é quando se utiliza curva de calibração para determinação quantitativa de concentração.

Desta forma procuramos desenvolver um conversor unipolar, com 12 bits de resolução, razoável precisão, boa linearidade, velocidade compatível com o antigo conversor, razoável estabilidade térmica e que utilizasse componentes de fácil aquisição.

Existem diversos métodos de conversão A/D [9] sendo que os mais comuns estão baseados em um dos princípios abaixo:

► O sinal analógico é integrado através de um capacitor e seu valor é quantificado pela contagem do tempo de carga ou descarga.

► Um conversor D/A gera um sinal que é comparado analogicamente com a entrada desconhecida, permitindo saber qual destes é mais intenso. Assim, após várias conversões D/A, é possível determinar o valor digital que mais se aproxima do valor desta entrada.

Na primeira categoria estão os conversores de dupla rampa, que são bastante estáveis e precisos, embora sejam lentos. Na segunda categoria estão os conversores por aproximações sucessivas que podem atingir alta velocidade de conversão, possuindo como desvantagem a necessidade de componentes de alta precisão.

Devido a necessidade de velocidade superior a aquela do conversor de dupla rampa e indisponibilidade de resistores de alta precisão no mercado nacional, optamos por um novo modelo.

O método de conversão está baseado na subtração de uma tensão de referência do valor de entrada. Se o resultado for positivo, significando que o sinal de entrada é o de maior valor, é gerado um bit 1 e somente a diferença entre os sinais passa para a

próxima etapa. No caso do resultado ser negativo, um bit 0 é gerado e o sinal de entrada passa integralmente para a próxima etapa. Deste sinal, é subtraído agora um novo valor de tensão de referência, que é igual a metade do valor da etapa anterior. O processo se repete gerando um bit 0 ou 1, alimentando a próxima etapa com o mesmo sinal ou a diferença. O número de etapas é igual ao número de bits do conversor, que são obtidos serialmente do mais para o menos significativo.

Na primeira etapa, a entrada é comparada com a tensão de referência, na segunda, com metade e na etapa N com $1/2^{N-1}$ vezes a tensão de referência. Assim, o fundo da escala é dado por

$$V_{fs} = \frac{2^N - 1}{2^{(N-1)}} V_{rf} \quad (6)$$

onde N é o número de bits do conversor e V_{rf} é a tensão de referência. Quando N é igual a 12, o fundo de escala é 1,9995 vezes a tensão de referência.

A implementação do algoritmo de conversão necessita de um número de módulos analógicos igual ao número de bits do conversor. Porém, foi possível utilizar somente um módulo, que trabalha de modo cíclico. Ou seja, o resultado de uma etapa é introduzido na entrada do mesmo módulo. O diagrama de blocos deste módulo pode ser visto na figura 20.

Inicialmente o sinal é armazenado em um *Sample & Hold*, enquanto o outro é zerado. A seguir a saída do S&H alimenta dois amplificadores operacionais simultaneamente. Um destes faz a subtração da tensão de referência e o resultado é que irá selecionar, através da chave CMOS, qual dos sinais irá realimentar o módulo. Este novo sinal é armazenado no S&H que inicialmente estava zerado. Na etapa seguinte o processo se repete, alternando-se as funções dos S&H. Ao dar ganho dois, os amplificadores permitem que na próxima etapa a tensão de referência não seja a metade da etapa anterior, mas de mesmo

- INTERFACEAMENTO -

valor.

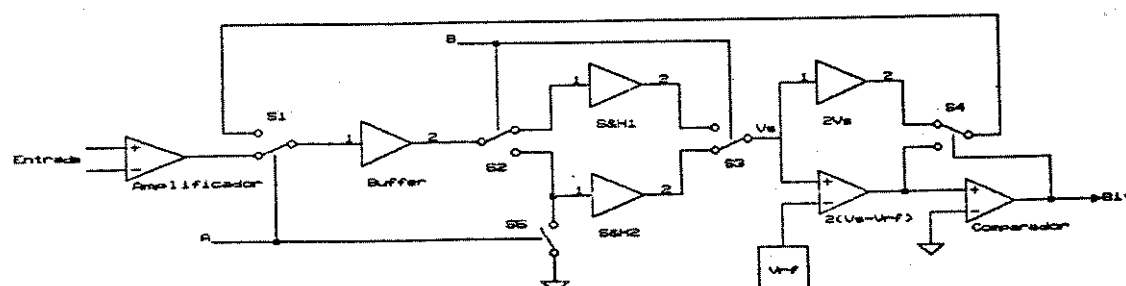


Figura 20 - Diagrama de blocos do módulo analógico do conversor A/D. As chaves CMOS S1 e S5 são acionadas (A) no início da conversão, de modo que S5 fique aberta e S1 permita a realimentação vinda de S4. As chaves S2 e S3 são acionadas juntas (B), de modo que os S&H trabalhem alternadamente. A chave S4 é controlada pelo comparador, realimentando o módulo com o dobro do sinal de entrada ou o dobro da diferença entre este e a tensão de referência.

Para controlar este módulo, é necessário um outro módulo de controle. Este possui um oscilador que determina a frequência de trabalho (*clock*). O sinal de *clock* é dividido gerando os sinais de controle das chaves CMOS do módulo analógico.

O diagrama do módulo analógico é apresentado na figura 21. O amplificador operacional A1 permite a entrada diferencial e D1 protege este contra sobretensão. Para servir de *buffer* que alimenta os capacitores C1 e C2 dos S&H, foi utilizado A2. Os trimpots R8 e R9 permitem que as tensões de *offset* (V_{os}) de A2, A3 e A4 sejam compensadas. O amplificador A5 possui ganho 2 e V_{os} é compensada por R13.

O amplificador A6 não necessita de ajuste de V_{os} , pois este é compensado no ajuste da tensão de referência através de R21. Do modo como foi configurado, o ganho para a entrada não inversora é

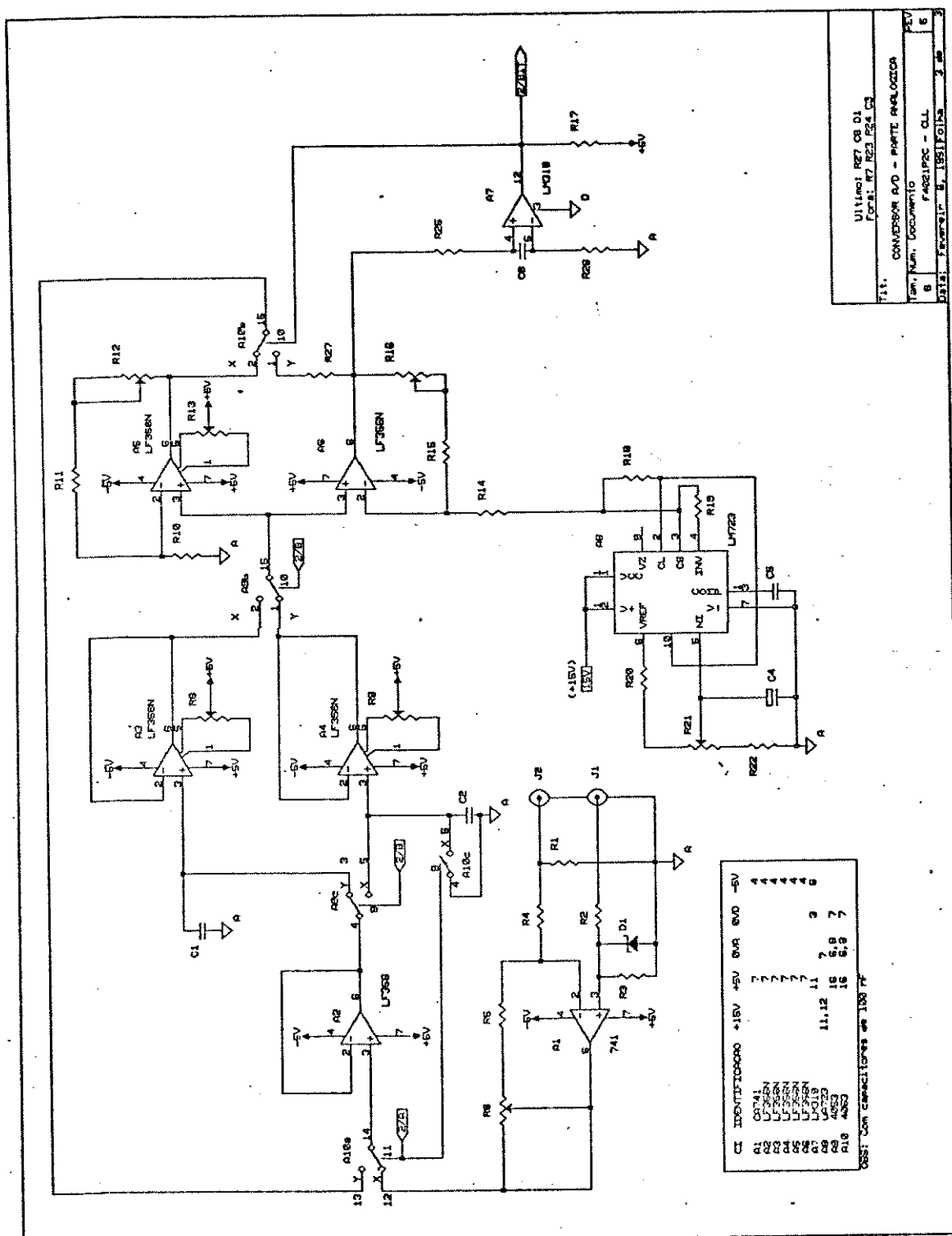


Figura 21

- INTERFACEAMENTO -

dois, enquanto a tensão de referência possui ganho um. Assim, o valor de tensão de referência na saída de A8 deve ser o dobro do valor desejado.

O comparador A7 possui um filtro RC na entrada afim de eliminar a interferência gerada pelo chaveamento de A10b.

Na figura 22 é apresentado o diagrama do módulo digital ou de controle do conversor. O conjunto formado pelas portas NAND U7A, U7B e pelo inversor U5B funciona como um oscilador que é disparado pelo *flip-flop* U6B e C no início da conversão. O contador U8 alimentado pelo sinal de *clock* gera os sinais que controlam as chaves CMOS, o contador de bits e o *shift-register*.

O contador U9 determina o número de bits do conversor através dos pinos 15, 1, 10 e 9. Ao atingir o valor estipulado de bits, é gerado um sinal em B0 (pino 13) que, através do inversor U5A, desativa o *flip-flop*, interrompendo o processo. Neste instante, o *shift-register* formado por U10 e U11 contem o valor convertido, que pode ser acessado pelo microcomputador através dos *buffers* U3 e U4.

O diagrama de temporização dos principais pontos deste circuito pode ser observado na figura 23. Após a inicialização do conversor, os S&H alternam suas funções e podem ser observados pulsos nas transições. A saída de A6 (pino 6) determina o estado lógico do sinal Bit. Os *shift-registers* são atualizados na transição baixo-alto do sinal do pino 3 de U8. O atraso de 0,4 μ s entre estes dois sinais é o suficiente para registrar o sinal Bit. Como ocorrem treze transições baixo-alto, são registrados 13 bits sendo que o primeiro é sempre zero e desprezado. Os 12 bits restantes são válidos na composição do valor convertido. No exemplo da figura 23, foi injetado um sinal de 0,975V e a tensão de referência ajustada para 1,500V. O valor convertido é 010100110011 em binário, ou seja, 1331 em decimal.

3.2.2 Avaliação do Conversor A/D

O conversor A/D, assim como qualquer circuito eletrônico,

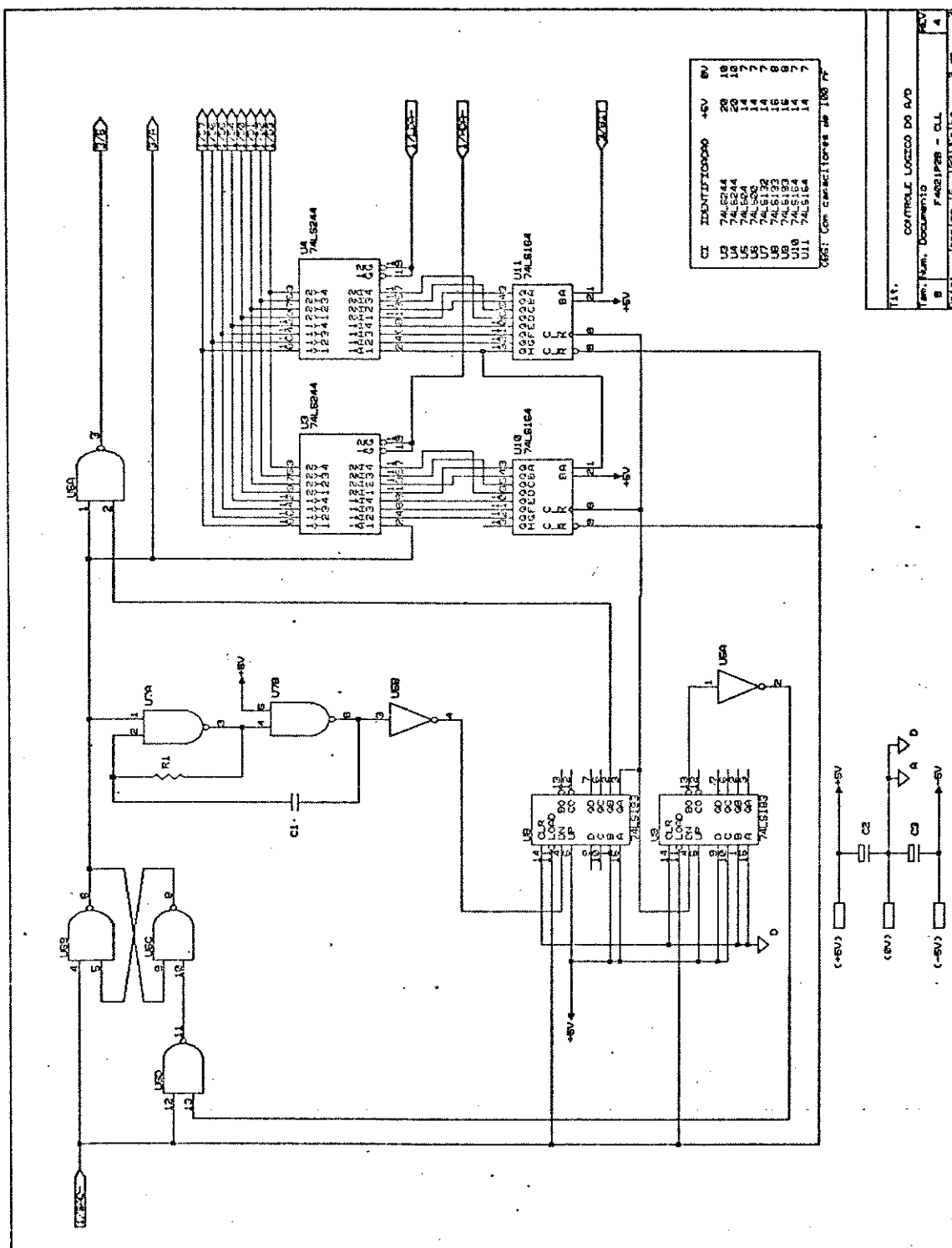


Figura 22

pode ter seu desempenho avaliado a partir de experimentos práticos e/ou cálculos matemáticos e estatísticos. Vários aspectos [11] devem ser analisados e considerados nesta avaliação: precisão, velocidade, custo, tamanho, consumo, etc. Destes, os três primeiros são os mais importantes neste caso.

O custo é difícil de ser avaliado, pois depende de fatores econômicos e de mercado e uma avaliação rígida feita no momento pode ser totalmente contrariada em um curto espaço de tempo. Um exemplo que mostra esta instabilidade, é que um simples soquete pode ter custo superior ao circuito integrado que nele será instalado. O importante a ser observado é que os componentes são em sua maioria de baixo custo e fácil aquisição. Os componentes principais em termos de custo são os amplificadores operacionais LF 356N.

Como foi dimensionado, o conversor trabalha com etapas de aproximadamente $5,6 \mu s$, o que resulta em um total de $70 \mu s$ para conversão de 12 bits. Este tempo é razoável apesar de ser aproximadamente 3 vezes superior ao do conversor original ADC12QM da Analog Devices. Existe um compromisso entre tempo de conversão e precisão, determinado principalmente pelos S&H: Os capacitores C1 e C2 podem ser de valor menor, diminuindo o tempo de amostragem, porém a precisão será sacrificada devido às descargas das capacitâncias internas da chave A9c. Se for desejada maior velocidade, o *buffer* A2 deverá ser substituído por um com maior capacidade de corrente de saída e a chave A9c, por outro modelo com menor resistência quando ativada e menor capacitância gatilho/porta.

- INTERFACEAMENTO -

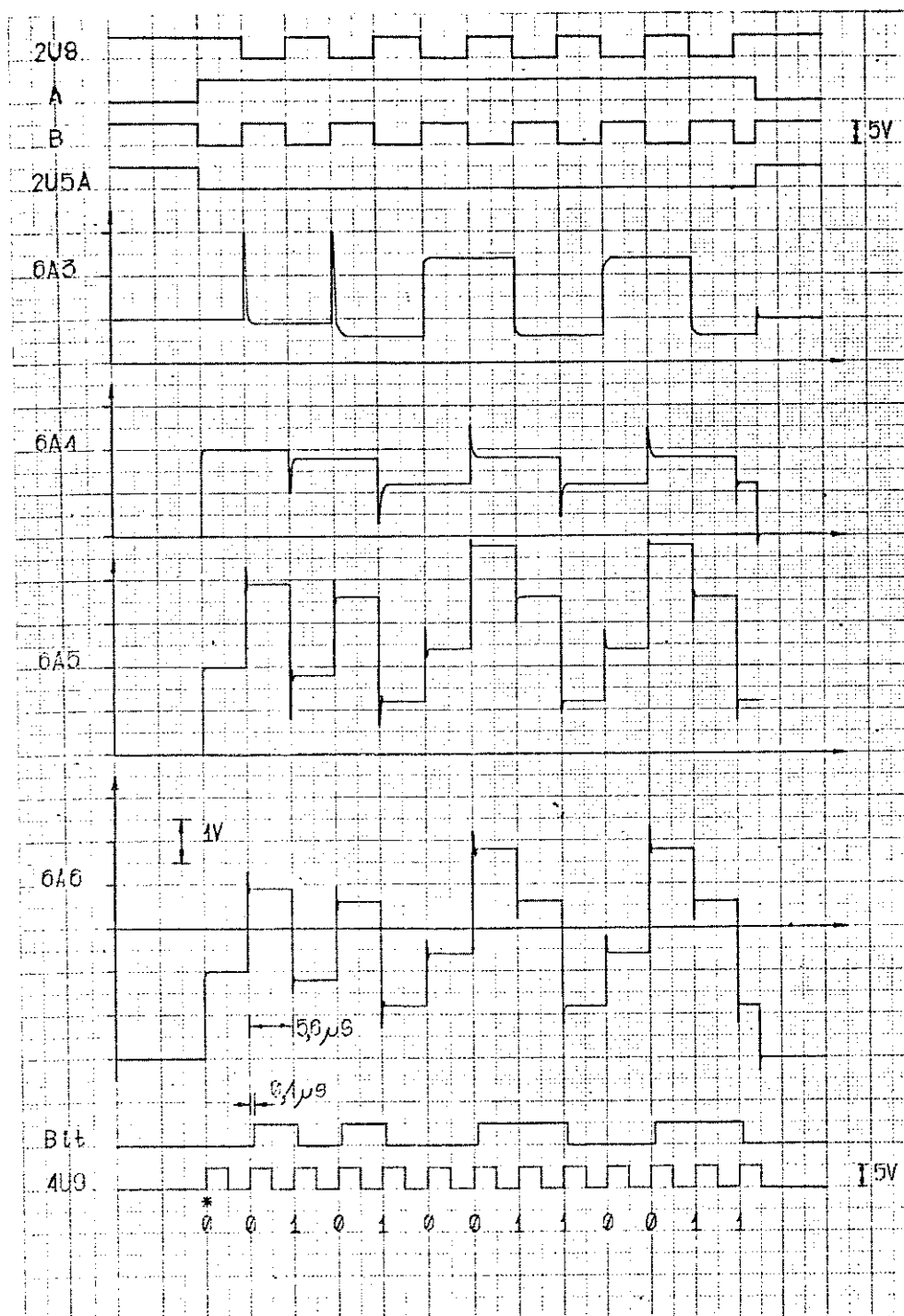


Figura 23 - Temporização do conversor A/D. São apresentados sinais digitais e analógicos de vários pontos dos circuitos das figuras 21 e 22. Os sinais A, B e Bit correspondem às conexões de mesma nomenclatura entre estas figuras e os demais são denotados pelo número do pino e identificação do circuito integrado. Neste exemplo, foi injetado na entrada um sinal de 0,975V e para obter $V_{ref}=1,500V$, a tensão no pino 3 de AB foi ajustada para 3,000V. O bit zero gerado no início (*) é sempre desprezado.

Por ser configurado com resolução de 12 bits, o erro de quantificação - metade do bit menos significativo - é de 1/8192 do fundo de escala. Porém, esta não é a única fonte de erro que determinará a precisão da conversão. Segundo Hoeschele [10], a precisão de um conversor é função da soma de erros de elementos individuais do sistema. Os erros determinados, aleatórios e sistemáticos são considerados na composição do erro global:

$$\varepsilon_t = \sum_{i=1}^m \varepsilon_{di} + \left[\sum_{j=1}^n \varepsilon_{aj}^2 \right]^{1/2} + \left[\sum_{k=1}^r \varepsilon_{sk}^2 \right]^{1/2} \quad (7)$$

onde ε_t é o erro total, ε_{di} é o erro determinado da causa i , ε_{aj} é o erro aleatório da causa j e ε_{sk} é o erro sistemático da causa k .

Com o esquema da figura 21, pode-se identificar as possíveis fontes de erro. A análise é feita a partir da chave CMOS A10a, pois o amplificador de entrada A1 não faz parte do conversor A/D.

A principal fonte de erro sistemático é o coeficiente térmico dos diversos componentes. Hoeschele faz suas análises para um intervalo de $\pm 30^\circ\text{C}$. Devido às condições de trabalho prevista para este circuito, descritas no tópico anterior, acreditamos que um intervalo de $\pm 3^\circ\text{C}$ seja mais realista. Assim, estas duas situações são consideradas.

As principais fontes de erros determinados seriam os amplificadores A5 e A6, pois os ganhos são estipulados por resistores. Porém, com a utilização dos resistores *trimpots* multivoltas R12 e R16, o ganho pode ser precisamente determinado.

A tabela 5 mostra o resumo das principais fontes de erros e o efeito total destes na precisão do conversor. O tratamento é diferente do dado por Hoeschele, pois são consideradas os desvios absolutos. A última linha apresenta o erro relativo tomando o fundo de escala como 4 volts.

- INTERFACEAMENTO -

Fonte de Erro	Aleatório	Sistemático	
		$\pm 3^{\circ}\text{C}$	$\pm 30^{\circ}\text{C}$
Driver A_2			
$\Delta V_{os}/\Delta T = 5\mu\text{V}/^{\circ}\text{C}$		$\pm 0,015$	$\pm 0,150$
Sample & Hold			
$C_H = 4,7\text{nF}$			
$R_{ON} = 120\Omega$			
$I_B = 10\text{pA}$			
$I_L = 20\text{pA}$			
$t_A = 6\mu\text{s}$			
$C_{ios} = 0,2\text{pF}$			
$\Delta V_{max} = 5\text{V}$			
Amostragem	$\pm 0,040$		
Chaveamento	$\pm 0,213$		
Buffers A_3 e A_4			
$\Delta V_{os}/\Delta T = 5\mu\text{V}/^{\circ}\text{C}$		$\pm 0,015$	$\pm 0,150$
Tensão de Referência			
$V_{rf} = 2\text{V}$			
$\text{Reg}_{IN} = 0,01\%\text{V}/^{\circ}\text{C}$	$\pm 0,200$		
$\text{TCV}_O = 0,002\%\text{V}/^{\circ}\text{C}$		$\pm 0,120$	$\pm 1,200$
Amplificadores A_5 e A_6			
$\Delta V_{os}/\Delta T = 5\mu\text{V}/^{\circ}\text{C}$		$\pm 0,015$	$\pm 0,150$
Total para um ciclo			
	$\pm 0,295$	$\pm 0,123$	$\pm 1,228$
Total para 12 bits			
	$\pm 1,022$	$\pm 0,426$	$\pm 4,254$
Quantificação			
$V_{fs} = 4\text{V}$			
$1/2 \text{ LSB}$	$\pm 0,488$		
Erro Total Absoluto			
		$\pm 1,559$	$\pm 5,387$
Erro Total Relativo			
		$\pm 0,039\%$	$\pm 0,13\%$

Tabela 5 - Resumo dos cálculos de erros para o conversor A/D. Foram tomadas como fontes de informações as referências 12, 13 e 14. Todos os valores, com exceção da última linha, são dados em milivolts.

- INTERFACEAMENTO -

Como a constante de tempo RC do S&H é pequena ($0,56\mu\text{s}$), o tempo de cada etapa do conversor poderia ser reduzido de $5,6\mu\text{s}$ para 3 ou $4\mu\text{s}$ sem comprometer significativamente a precisão do conversor.

Geralmente, a precisão de um conversor é dado como 1/2 bit menos significativo (LSB). A tabela 6 permite comparar a precisão do conversor com conversores de 8 a 12 bits. Para o caso de considerar um intervalo de temperatura de $\pm 30^\circ\text{C}$, o conversor pode ser visto como de precisão entre 8 e 9 bits. Considerando um intervalo de $\pm 3^\circ\text{C}$, a precisão está entre 10 e 11 bits. Em qualquer um dos dois casos, o desempenho do conversor é satisfatório.

Número de bits	Precisão para $\pm 1/2\text{LSB}$
8	$\pm 0,20\%$
$\Delta T \pm 30^\circ\text{C}$	$\pm 0,13\%$
9	$\pm 0,098\%$
10	$\pm 0,049\%$
$\Delta T \pm 3^\circ\text{C}$	$\pm 0,039\%$
11	$\pm 0,024\%$
12	$\pm 0,012\%$

Tabela 6 - Comparação da precisão do conversor A/D a partir dos resultados da tabela 5. Como se pode observar, a precisão do conversor está entre 8 e 11 bits, dependendo das condições de temperatura ambiente.

3.3 Software

A tabela 7 mostra os endereços utilizados nesta interface. Existem vários endereços livres, que podem eventualmente ser utilizados caso uma expansão das funções seja desejada.

A aquisição de dados do espectrômetro de massas é feita de forma semelhante a do Finnigan 1015S/L. Já o controle das funções do cromatógrafo é feito de modo diferente. As grandezas analógicas são controladas através de conversores D/A, que fornecem o valor desejado a um circuito que, comparando com o sinal de sensores, controlam as etapas de potência. Com excessão do conversor do

- INTERFACEAMENTO -

aquecimento da coluna, que é de 12 bits, os demais são de 8 bits. Através das portas 301H e 302H, que são unicamente de leitura, é possível conhecer o *status* dos diversos dispositivos.

Os bits mais significativos da porta 30AH permitem controlar solenóides responsáveis por janelas que refrigeram o forno (bits 4 e 5), além de outro que divide o fluxo do injetor (bit 7). A divisão do fluxo do separador é conseguido através do bit 0 da porta 30CH.

Porta	Função
301H	Status da temperatura da coluna e vazão das válvulas do cromatógrafo
302H	Status da temperatura do injetor e detector do cromatógrafo
309H	Byte menos significativo do D/A que determina a temperatura da coluna
30AH	Byte mais significativo do D/A que determina a temperatura da coluna, controle de ventilação do forno e <i>split</i> do injetor
30CH	D/A que controla o fluxo da válvula A e <i>split</i> do separador
30DH	D/A que controla o fluxo da válvula B
310H	D/A que controla a temperatura do injetor
311H	D/A que controla a temperatura do detector A
312H	D/A que controla a temperatura do separador
318H	Byte menos significativo do D/A do espectrômetro
319H	Byte mais significativo do D/A do espectrômetro
31AH	Inicia o conversor A/D do espectrômetro
31BH	Byte menos significativo do A/D do espectrômetro
31CH	Byte mais significativo do A/D do espectrômetro

Tabela 7 - Endereços utilizados pela interface do CCEN-UFFA.

O arquivo de inclusão Interfac.CLL reúne todos os procedimentos e funções necessários para controlar e adquirir dados do cromatógrafo e espectrômetro. Este arquivo pode ser incluído em qualquer programa para controle, calibração ou

aquisição de dados.

4 Bibliografia

- 1 - C.L. do Lago e C. Kascheres, *Química Nova* 11, 377 (1988).
- 2 - S.M. Sze, "Physics of Semiconductor Devices", 2nd Ed., John Wiley & Sons, New York, 1981, p. 35.
- 3 - S.K. Bansal and R.Parshad, *Indian J. Pure Appl. Phys.* 7, 210 (1969).
- 4 - S.L. Agrawal and R. Swami, *Eletr. Int.* 22, 176 (1980).
- 5 - A.A. Khan, *Indian J. Pure Appl. Phys.* 10, 81 (1972).
- 6 - G.R.M. Rao, "MOSFET Magnetic Sensor and Application to Integrated Circuits", thesis, Southern Methodist Univ., Dallas, Tex., 1971.
- 7 - M. Garfinkel and W.E. Engeler, *J. Appl. Phys.* 36, 1877 (1965).
- 8 - J.D. Lenk, "Handbook for Transistors", Prentice-Hall, New Jersey, 1976, p. 185.
- 9 - C.A. Harper, "Handbook of Eletronic Packaging", McGraw-Hill, New York, 1969, p. 10-35.
- 10 - D.F. Hoeschele, "Analog-to-Digital / Digital-to-Analog Conversion Techniques", John Wiley & Sons, New York, 1968.
- 11 - H. Schmid, "Eletronic Analog/Digital Conversions", Van Nostrand, New York, 1970, p. 485.

- INTERFACEAMENTO -

- 12 - Signetics Co., "Analog Data Manual", Sunnyvale, Ca-USA, 1981, p. 30.
- 13 - Motorola Inc., "Semiconductor Data Library", Vol 6/Series B, 1976, p. 4-64.
- 14 - National Semiconductor Co., "Logic Databook", Santa Clara, Ca-USA, 1984, p. 5-152.

CAPÍTULO IV

PROGRAMAS PARA OS SISTEMAS DE DADOS

Os sistemas de dados desenvolvidos neste trabalho possuem os mesmos princípios básicos: é necessária uma curva de calibração entre o sinal digital e o sinal analógico referente a relação m/z , os espectros adquiridos podem ser apresentados em vídeo, impressos ou armazenados para tratamento posterior. Neste capítulo serão apresentados os programas para calibração, aquisição e tratamento de espectros. Todos os programas foram desenvolvidos em Turbo Pascal versão 3.01 da Borland e as listagens se encontram no apêndice 2.

1 Programas para o Finnigan 1015S/L

A figura 1 mostra os programas, o fluxo de dados e arquivos utilizados para aquisição de espectros do Finnigan 1015S/L. O programa de calibração CAL100 gera um conjunto de dados de m/z em função do valor enviado para o conversor D/A, que permite ao programa AJUSTE gerar o polinômio de calibração. Este polinômio permite ao programa COL100 traduzir o valor de m/z desejado no valor correspondente a ser enviado para o conversor D/A. Os arquivos de espectros gerados por este último programa pode ser impresso pelo programa MD2 ou utilizado em qualquer outro programa desenvolvido para tratamento específico (capítulos V e VI).

- PROGRAMAS PARA OS SISTEMAS DE DADOS -

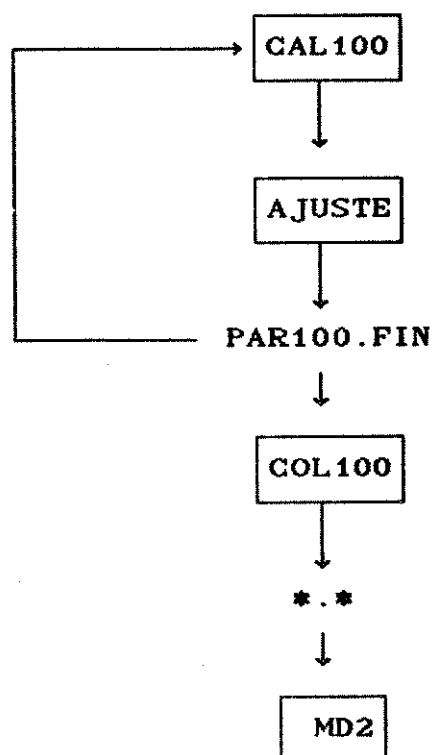


Figura 1 - Programas, arquivos e o fluxo de dados do sistema de aquisição de espectros do Finnigan 1015S/L. A moldura indica um programa. O símbolo * significa que o arquivo pode possuir qualquer conjunto de códigos ASCII no nome ou complemento, seguindo a regra para o DOS.

1.1 O Programa CAL100

Este programa é responsável por gerar o conjunto de dados que serão utilizados na calibração da conversão de m/z em valor para D/A. Uma vez posicionado o espectrômetro na faixa de 0 a 100 daltons, o conversor D/A de 16 bits pode selecionar valores de m/z nesta faixa recebendo valores entre 0 e 65535, respectivamente. Desta forma, é possível variar o valor de m/z com incrementos de 0,0015 daltons. Como a resolução do espectrômetro é unitária, não é necessário um incremento tão pequeno para fazer a varredura. O programa permite selecionar o incremento, sendo o valor *default* 0,05d e mínimo 0,01d.

Para evitar que os picos do *background* estejam presentes no espectro de calibração, este é subtraído. Para isto é

primeiramente feita a varredura de um espectro sem introduzir a amostra. O espectro arquivado em um vetor, é posteriormente subtraído do espectro da amostra de calibração.

Após a aquisição do espectro de calibração e determinada a linha base, os picos são localizados automaticamente. Para isto o vetor é percorrido e se forem encontrados pelo menos 8 sinais consecutivos acima da linha base, o conjunto será considerado como um pico. O valor de D/A é determinado pelo centróide do conjunto:

$$p = \frac{\sum_{j=i}^f j \cdot S_j}{\sum_{j=i}^f S_j} \quad (1)$$

onde p é o índice do vetor correspondente ao centróide, i e f são os índices inicial e final, respectivamente, da região considerada como um pico e S é o sinal adquirido do conversor A/D. Como o índice do vetor está relacionado com o valor de m/z, é possível através da antiga curva de calibração determinar o valor de D/A a que corresponde. Para determinar a m/z referente a este valor, pode-se utilizar o valor correspondente a p aproximado para a m/z inteira mais próxima ou, no caso da calibração anterior não apresentar a relação correta com a m/z do pico, a determinação deve ser feita manualmente.

O programa permite ainda acrescentar ou eliminar manualmente um pico do conjunto encontrado antes da calibração. Isto é necessário pois em alguns casos podem ocorrer interferências no momento de aquisição, gerando picos falsos ou impedindo a identificação correta.

A figura 2 mostra a tela apresentada pelo programa onde um pico com m/z atual de 216,65d foi localizado. Pela calibração automática, o pico é aproximado para 217d. É possível observar que dois outros picos de baixa intensidade -um acima e outro abaixo do principal- foram determinados erroneamente e, portanto, devem ser

eliminados.

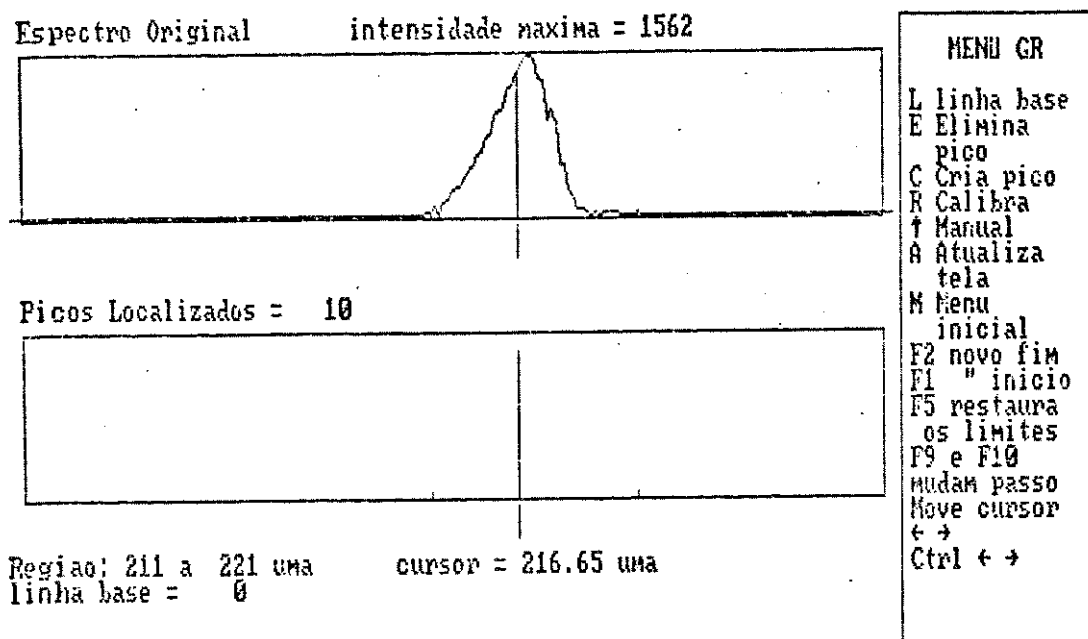


Figura 2 - Tela apresentada pelo programa CAL100. O cursor se encontra sob o maior pico localizado. Devido ao ruído, foram determinados dois outros picos que não pertencem ao espectro da amostra.

1.2 O programa AJUSTE

Este programa permite, através do método de mínimos quadrados, ajustar um polinômio de grau n por um conjunto de pontos fornecidos pelo operador ou por um arquivo em ASCII.

O polinômio procurado deve minimizar a equação

$$D = \sum_{i=1}^N \left(y_i - \sum_{g=0}^K a_g x_i^g \right)^2 \quad (2)$$

onde D é a distância euclidiana entre o ponto experimental (x_i, y_i) e o ponto pertencente ao polinômio para o mesmo valor de abscissa, N é o número de pontos, K é o grau do polinômio e a_g é o coeficiente

g do polinômio. Para tanto as derivadas parciais com relação ao coeficientes devem ser zero. Assim, é gerado o sistema de equações

$$\sum_{g=0}^K a_g \left(\sum_{i=1}^n x_i^{g+g} \right) = \sum_{i=1}^n y_i x_i^g \quad \text{Para } G \text{ de } 0 \text{ a } K \quad (3)$$

A solução deste sistema, fornece o polinômio com o grau K estipulado pelo operador.

Embora a princípio, a resposta do quadrupolo deva ser linear com o sinal do conversor D/A, empiricamente se determinou que um polinômio de grau superior é mais adequado. Isto ocorre devido a falta de linearidade do conversor D/A e/ou do circuito de controle do quadrupolo.

1.3 O Programa COL100

Uma vez de posse da curva de calibração, é possível fazer a varredura do espectro de modo mais rápido. Isto é possível pois, em baixa resolução, os picos ocorrem geralmente em intervalos inteiros de m/z . No caso de se formar um íon com peso molecular ímpar e duplamente carregado, o valor de m/z será intermediário entre dois valores inteiros. Se forem desprezados estes casos pouco comuns, a varredura poderá ser feita com saltos de 1 dalton.

A aquisição de um espectro com o programa COL100 é feito enviando ao D/A o valor inicial de m/z para estabilizar o quadrupolo. A partir daí, o processo de aquisição se inicia:

► O conversor D/A é atualizado,

► um intervalo de 2ms é aguardado para estabilizar no novo valor de m/z e

► são adquiridos um ou mais valores do conversor A/D, armazenando a soma em uma posição do vetor de espectro.

Do espectro da amostra é subtraído o espectro de *background*, que elimina os picos dos gases residuais e a linha base do sistema eletrônico. Para avaliar o espectro, é possível graficá-lo no vídeo e se for desejado, pode ser arquivado em disco.

1.4 O Programa MD2

Este programa permite gerar tabelas e gráficos referentes aos espectros armazenados em disquete. O arquivo deve ser do tipo ASCII contendo um par de valores por linha: o primeiro é um valor real referente a m/z e o segundo, um valor inteiro referente a altura do pico. É permitido também introduzir manualmente -através do teclado- um conjunto de picos referente a um espectro.

Após graficar o espectro no vídeo, é possível destacar os principais picos de cada *cluster* automaticamente, ampliar ou reduzir tanto a região de m/z quanto a intensidade do espectro. A figura 3 mostra o registro de um espectro de benzeno.

A tabela dos picos apresenta duas colunas referentes ao valor de m/z dos picos. A primeira coluna dá o valor sem aproximações, enquanto a segunda faz a aproximação para valores com resolução de 0,5d. A figura 4 mostra a tabela dos picos para o espectro apresentado na figura 3.

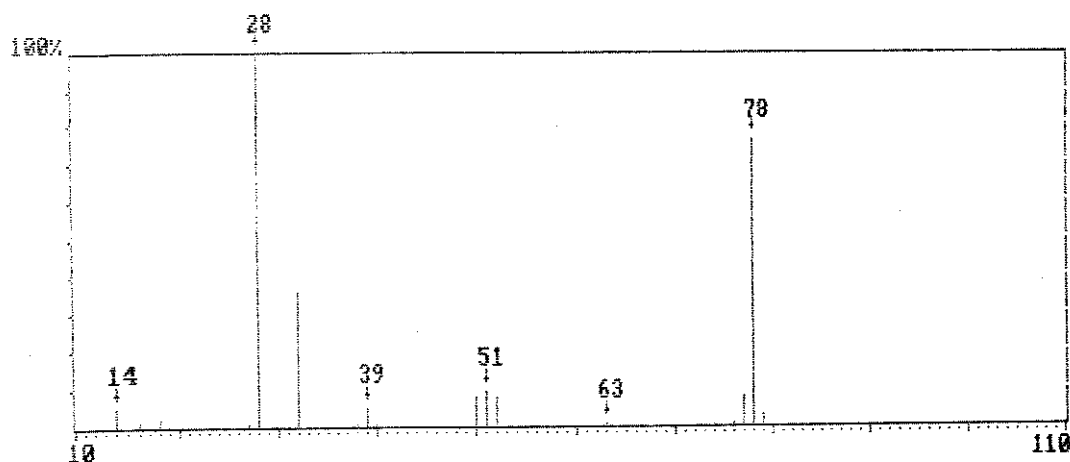


Figura 3 - Espectro de benzeno. O pico de m/z 92d não foi numerado devido a proximidade de 4d de diferença em relação ao pico 28d.

- PROGRAMAS PARA OS SISTEMAS DE DADOS -

Espectro de Massas

Arquivo: benzeno

Relacao dos picos

m/z	m/z	Intensidade(em %)	m/z	m/z	Intensidade(em %)	m/z	m/z	Intensidade(em %)
14.02	14.0	5.30	36.97	37.0	0.20	63.01	63.0	0.50
15.98	16.0	1.10	38.01	38.0	0.80	74.01	74.0	1.00
17.99	18.0	2.20	39.00	39.0	5.30	75.98	76.0	0.40
26.00	26.0	0.10	39.93	40.0	0.80	77.02	77.0	7.70
27.04	27.0	0.50	48.99	49.0	0.10	77.97	78.0	75.40
28.00	28.0	100.00	49.99	50.0	7.90	79.00	79.0	2.50
29.03	29.0	0.20	51.00	51.0	9.20			
31.98	32.0	35.50	52.01	52.0	8.10			

Figura 4 - Tabela dos picos do espectro de benzeno apresentado na figura 3.

2 Programas para o MAT311A

A figura 5 mostra o conjunto de programas e arquivos para aquisição de espectros do MAT311A. O programa MATCOL coleta um espectro de forma contínua e gera um arquivo que será utilizado por MATTRATN para localizar, determinar m/z e normalizar o espectro. Este último programa também pode fornecer o espectro sem determinar m/z dos picos, permitindo ao operador fazer a calibração através do programa AJUSTE. Para tabelar e graficar o espectro, também é utilizado o programa MD2.

2.1 O programa MATCOL

Como o espectrômetro MAT311A não permite o controle da varredura, o método de aquisição de um espectro consiste em fazer a leitura do sinal do sensor de campo magnético e do multiplicador de elétrons simultaneamente. Assim, são armazenados estes dois valores através da leitura do conversor A/D, ocupando no total 4

- PROGRAMAS PARA OS SISTEMAS DE DADOS -

bytes por ponto (cada valor lido do conversor A/D ocupa dois bytes). Como o espectro é armazenado em um segmento de memória de 64kbytes, o programa permite a aquisição de 15000 pontos, sendo que cada posição corresponde na verdade a média de 4 leituras do conversor A/D.

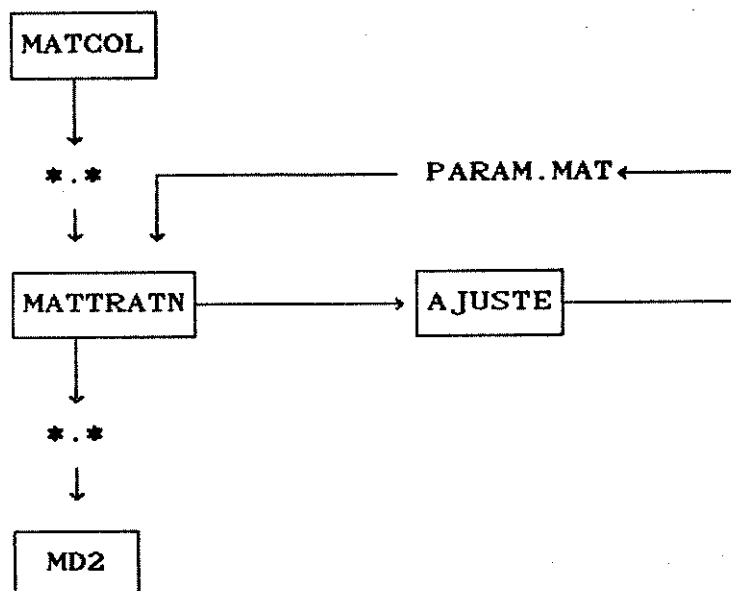


Figura 5 - Programas, arquivos e o fluxo de dados do sistema de aquisição de espectros do MAT311A. A moldura indica um programa. O símbolo * significa que o arquivo pode possuir qualquer conjunto de códigos ASCII no nome ou complemento, seguindo a regra para o DOS.

Há um compromisso entre quatro variáveis: velocidade de aquisição, memória consumida, formato do pico e quantidade de amostra. Este compromisso pode ser visualizado tomando como base, por exemplo, o formato do pico. Para caracterizar um pico é importante que se disponha do maior número de pontos com o menor nível de ruído possível. Como a aquisição de dados é contínua, mesmo quando não há picos, o primeiro requisito tende a causar um excesso de consumo de memória. Com o acúmulo de leituras para fornecer um valor médio, o nível de ruído é minimizado, porém a velocidade de aquisição diminui. À medida que a velocidade

diminui, torna-se necessário que se disponha de maior quantidade de amostra para que não diminua a pressão na câmara de ionização durante a aquisição. O ponto de equilíbrio deste compromisso foi determinado empiricamente com resultados satisfatórios.

2.2 O programa MATTRATN

Este programa faz o tratamento *off line* do espectro adquirido por MATCOL, fornecendo a listagem dos picos para o programa MD2. O "coração" do programa está na subrotina *TrataDados* que por sua vez executa quatro procedimentos:

►*LimpaRampa*: Este procedimento inicialmente toma o vetor referente ao sinal do campo magnético em regiões de 11 dados obtendo médias. Estes valores médios de cada região são utilizados na obtenção de um polinômio de quarto grau que descreve o comportamento do sinal em função do tempo. Para a próxima etapa, os dados experimentais são abandonados e em seus lugares são utilizados os valores determinados por este polinômio.

►*EncPicos*: Para detectar a presença de um pico, é necessário que o sinal supere um valor de limiar que é determinado pelo desvio padrão da linha base. O valor *default* para este limiar é 2,576 vezes o valor do desvio padrão e pode ser alterado pelo operador. Se três ou mais sinais consecutivos superarem o valor de limiar, é considerada a existência de um pico. Com este procedimento, a probabilidade de que o ruído cause uma falsa determinação de um pico é inferior a 1 em 500. Ou seja, deve ocorrer, em média, a cada 500 espectros. Por outro lado, há um compromisso entre esta margem de segurança e o limite de detecção.

Devido à baixa resolução do espectrômetro, pode ocorrer que dois ou mais picos fiquem compreendidos em uma mesma região. Assim, é utilizada a primeira derivada para localizar o início e o fim de cada pico. Cada pico é caracterizado pela área e o centróide.

►*Traduz*: Através da curva de calibração, que é também um polinômio do quarto grau, o valor do centróide é convertido em

- PROGRAMAS PARA OS SISTEMAS DE DADOS -

valor de m/z . Como a primeira derivada é sensível ao ruído, pode ocorrer que, no procedimento anterior, um pico seja desmembrado em dois ou mais. Assim, os picos com diferença de m/z inferior a 0,3d são aglutinados em um só.

►*Normaliza*: Como última etapa, a intensidade dos picos é normalizada de modo que o mais intenso corresponda a 100.

3 Programas para o Finnigan 4021

Este é o sistema de dados mais recente e, portanto, absorveu todas as técnicas aplicadas aos sistemas anteriores além de possuir recursos adicionais. O desempenho global é superior não devido apenas ao *software* ou *hardware* desenvolvido, mas principalmente ao bom funcionamento do espectrômetro e cromatógrafo.

A figura 6 mostra o conjunto de programas e arquivos deste sistema de dados. O programa CAL4021 utiliza o arquivo FC43.CAL que contém o espectro do perfluorotributilamina, uma substância utilizada tradicionalmente para calibração de espectrômetros de massas e cujo nome comercial é FC43. Os parâmetros de calibração gerados são arquivados em F4021.CAL para serem utilizados pelo programa GCMS. Além destes parâmetros, é necessária uma série de outros dados sobre o controle do cromatógrafo e espectrômetro que estão nos arquivos com terminação PAR. Este programa gera o arquivo GC.SCN - que contém o último cromatograma efetuado - e os arquivos com terminação SCN e INF, que contém cromatogramas e condições de análises salvos pelo operador. O programa TRATA permite recuperar estes dois últimos arquivos para analisar o cromatograma. Caso seja desejado transferir para a impressora um dos espectros do cromatograma, é utilizado o programa GRAF que trabalha a partir dos arquivos de espectro SCANx, onde x é o número do espectro dentro do cromatograma.

- PROGRAMAS PARA OS SISTEMAS DE DADOS -

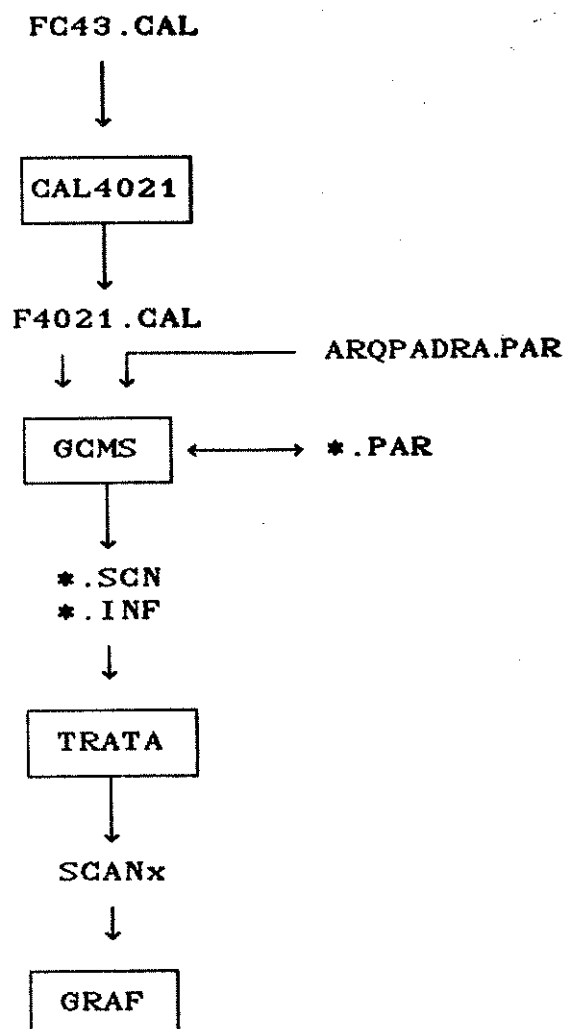


Figura 6 - Programas, arquivos e o fluxo de dados do sistema de aquisição de espectros do Finnigan 4021. A moldura indica um programa. O símbolo * significa que o arquivo pode possuir qualquer conjunto de códigos ASCII no nome, seguindo a regra para o DOS. Os arquivos SCANx possuem a terminação de acordo com o número deste dentro do cromatograma (x).

3.1 O Programa CAL4021

Este programa é mais eficiente que seu equivalente CAL100, pois varre, localiza os picos e gera a curva de calibração automaticamente. Ao contrário deste, o CAL4021 não arquiva o espectro todo para posterior análise, mas desconta a linha base e localiza os picos em tempo real. Este procedimento permite fazer incrementos menores pois o consumo de memória é baixo. Neste caso

- PROGRAMAS PARA OS SISTEMAS DE DADOS -

também é utilizado o centróide do pico, porém a intensidade é dada pela integral.

Para identificar corretamente os picos do espectro de calibração é utilizado o espectro padrão armazenado em disco no arquivo FC43.CAL. Este arquivo possui os principais picos do perfluorotributilamina a 70eV. Como o desvio que ocorre algum tempo após feito a última calibração é pequeno, a localização é facilitada. Identificados os valores de m/z pela antiga curva de calibração, um pico do padrão é comparado com todo o espectro adquirido. Quando for encontrado um pico com desvio inferior a 0,5d, a associação é feita. Se não houver correspondência entre um pico do padrão e nenhum dos picos da amostra, é impresso no vídeo uma mensagem alertando o operador.

Após identificar o valor de m/z correto de cada pico é feita a calibração e, se for confirmado pelo operador, o arquivo F4021.CAL é atualizado. Embora possa gerar polinômios de calibração de maior grau, foi determinado que basta uma reta para se obter ótimos resultados.

3.2 O programa GCMS

Este é o programa mais complexo do sistema de dados, pois deve controlar o cromatógrafo além de fazer a aquisição de espectros. Existe uma grande quantidade de parâmetros que devem ser estipulados pelo operador e para não se tornar repetitivo, é possível arquivar em disco todas as condições em que foi realizada uma análise. Além dos arquivos com terminação PAR gerados pelo operador, existe o arquivo ARQPADRA.PAR contendo um conjunto de condições padrões, que serve como ponto de partida para o programa.

Existem três grupos de condições de análise: um referente ao programa do cromatógrafo, outro referente ao programa do espectrômetro de massas e outro com os recursos da tela gráfica apresentada durante a aquisição, permitindo ao operador avaliar o cromatograma em tempo real.

- PROGRAMAS PARA OS SISTEMAS DE DADOS -

As condições cromatográficas são as seguintes:

- ▶ Fluxo do gás de arraste - se for utiliza coluna empacotada;
- ▶ Temperatura do injetor e separador;
- ▶ Temperatura da coluna quando não está sendo corrido um cromatograma;
- ▶ Tempo de equilíbrio - que é o tempo decorrido após as condições estipuladas serem atingidas para, então, iniciar o cromatograma.

É possível programar rampas de aquecimento da coluna e controlar as válvulas de divisão de fluxo no injetor e no separador. Os programas podem ser feitos com até 10 etapas. Cada etapa de aquecimento é caracterizada por:

- ▶ Temperatura inicial da coluna,
- ▶ Intervalo de tempo sob a temperatura inicial,
- ▶ Velocidade de aquecimento,
- ▶ Temperatura final após a rampa de aquecimento e
- ▶ Intervalo de tempo sob a temperatura final.

Se uma nova etapa for desejada, as condições iniciais de temperatura e tempo são idênticas às condições finais da etapa anterior e o operador estipula a velocidade de aquecimento e condições finais. Ao final de cada etapa, o tempo é apresentado e ao final do programa, é apresentado o tempo total previsto para o cromatograma.

O programa das válvulas é simples e cada etapa é caracterizada pela identificação da válvula - se do injetor ou separador - se esta deve ser ativada ou desativada e o tempo decorrido do início do cromatograma em que isto deve ocorrer.

Para compreender os parâmetros referentes ao espectrômetro, é necessário uma explicação do modo como é feita a aquisição de espectros. A varredura é feita do valor de m/z final para o inicial em etapas de um dalton. A cada valor de m/z é considerado uma margem em torno do valor central com intervalos de 0,1d. Uma vez calculado o valor de m/z inicial para localização de um pico a aquisição é iniciada:

- PROGRAMAS PARA OS SISTEMAS DE DADOS -

►O conversor D/A é atualizado;

►São feitas uma ou mais aquisições através do A/D, guardando-se o valor em uma posição de vetor após a subtração da linha base;

►O valor de m/z é diminuído em $0,1d$ e o processo se repete até atingir o final da margem.

É computada a soma das intensidades dos sinais e se esta for maior que um valor de limiar, é considerada a existência do pico. A intensidade do pico é a soma das intensidades para toda a margem considerada e o valor de m/z - com resolução de $0,05d$ - é dado pelo centróide. Todos os picos encontrados são arquivados em disco, juntamente com o tempo em que o espectro foi iniciado.

A linha base é computada antes de iniciar o cromatograma. Para isto é selecionado o valor de m/z igual a $10d$, para o qual são feitas 500 aquisições do conversor A/D. A linha base é o valor médio e variância é utilizada na determinação do limiar de detecção de um pico.

A condições de análise do espectrômetro são as seguintes:

►"Janela" de m/z - é a margem em torno do valor central para localização de um pico. São permitidos valores simétricos e assimétricos entre $-0,4$ e $+0,4d$, com resolução de $0,1d$. Como a varredura é feita com incrementos de $0,1d$, a margem escolhida influencia no tempo de aquisição do espectro.

►Número de desvios padrões da linha base para a formação do limiar de detecção. O limiar é dado por

$$L = n \cdot \sigma \cdot \sqrt{10(j_f - j_i) + 1} \quad (4)$$

onde n é o número de desvios, σ é o desvio padrão e j_i e j_f são os valores inicial e final da margem. Esta equação é derivada da soma das varianças (σ^2) para cada intervalo dentro da margem de m/z . O número de desvios padrões fornecido pelo operador é que determinará a confiança na determinação de um pico. Por exemplo, se o número de desvios padrões for um, a probabilidade de que o

- PROGRAMAS PARA OS SISTEMAS DE DADOS -

sinal exceda $\pm\sigma$ é 0,3174, porém, como um pico será detectado no caso de exceder $\pm\sigma$, a probabilidade é 0,1587. Isto quer dizer que de cada 100 valores de m/z que na verdade não apresentem picos, em aproximadamente 16 vezes ele será detectado. Para que isto seja válido é considerado que a média não se altera, ou seja, a linha base deve ser constante.

Assim como para o cromatógrafo, é possível programar o modo de aquisição de espectros do espectrômetro de massas. Os parâmetros são:

▶Valor de m/z inicial.

▶Valor de m/z final.

▶Momento inicial para aquisição. No caso do primeiro programa, que é obrigatório, o tempo é zero, ou seja, no início do cromatograma.

▶Número de conversões (na base 2), que deve ser um valor inteiro entre 0 e 4. Como internamente o número de conversões é determinado por este valor como potência de dois, este intervalo corresponde de 1 a 16 conversões A/D. Deve-se tomar cuidado ao utilizar valores diferentes durante o mesmo cromatograma, pois com a diminuição no número de conversões, há um aumento do nível de ruído e, portanto, o limiar para a detecção de picos pode se tornar incorreto.

O recurso de permitir a alteração da faixa de m/z durante a aquisição do cromatograma é bastante interessante. Em geral, os componentes de maior peso molecular têm maior tempo de retenção. Como ao aumentar o tempo de retenção, há um alargamento do pico cromatográfico, é possível aumentar a faixa de m/z - o que resulta em maior tempo de aquisição de um espectro - e ainda assim possuir um número razoável de pontos para caracterizar um pico cromatográfico.

O melhor desempenho do sistema, portanto, dependerá da prática do operador. Um procedimento razoável, no caso de se dispor de amostra suficiente, seria correr um cromatograma com ampla faixa de m/z para verificar o limite necessário em função do

tempo.

Os programas de aquecimento e controle de válvulas, assim como o programa do espectrômetro são compilados antes de iniciar a aquisição do cromatograma, gerando uma tabela contendo o cronograma de trabalho, que é consultado e, se for o caso, executado ao final de cada espectro.

O último grupo de condições se refere ao modo de apresentação gráfica durante a aquisição do cromatograma. Estão disponíveis três janelas, sendo que cada uma pode fazer um registro diferente durante o cromatograma:

►Cromatograma de uma região de m/z . Com este tipo de registro tem-se praticamente o cromatograma de corrente iônica total (TIC).

►Cromatograma de um íon, onde somente um valor de m/z escolhido pelo operador é acompanhado. O tempo consumido para gerar este cromatograma é bem menor que no caso anterior, pois não é necessário fazer a soma de um conjunto de picos.

►Espectro de uma região de m/z . Embora consuma tempo considerável, é bastante útil, pois permite observar se a intensidade do sinal do espectro é razoável, além de verificar se os limites de m/z foram corretamente estipulados.

►Janela desativada. Neste caso nada é processado e portanto, o tempo gasto é zero. Em casos onde exista a necessidade de fazer o maior número possível de aquisições, este é o melhor procedimento.

Seguindo a idéia de otimizar a aquisição do cromatograma, após determinadas as melhores condições através de um ou mais cromatogramas prévios, um novo é corrido desativando as janelas.

Se for desejado, o cromatograma pode ser armazenado em disco, gerando dois arquivos: um com terminação SCN contendo o cromatograma e outro com terminação INF, contendo a identificação da análise e as condições em que foram operados o cromatógrafo e o espectrômetro. O arquivo .SCN é na verdade apenas o arquivo GC.SCN renomeado. Este é composto por registros de 4 bytes preenchidos como mostrado na figura 7.

- PROGRAMAS PARA OS SISTEMAS DE DADOS -

	Campo m		Campo h	
	minuto	hora	centésimo	segundo
1º espectro	numscan		np	
	m/z x 20 do pico 1		área do pico 1	
	≈	≈	≈	≈
	m/z x 20 do pico np		área do pico np	
2º espectro	minuto	hora	centésimo	segundo
	numscan		np	
	m/z x 20 do pico 1		área do pico 1	
	≈	≈	≈	≈
	m/z x 20 do pico np		área do pico np	
	≈	≈	≈	≈

Figura 7 - Formato do arquivo do cromatograma. O primeiro registro é utilizado para armazenar o tempo no início da varredura do espectro. No segundo registro vai o número do espectro (numscan) e o número de picos (np), ambos ocupando 2 bytes cada. A partir daí, são armazenados os picos, sendo que o valor de m/z é multiplicado por 20, dando uma resolução de 0,05d.

3.3 O Programa TRATA

Este programa utiliza os arquivos gerados por GCMS para fornecer a listagem das condições de análise, informações gráficas sobre o cromatograma e os espectros de massas. Exemplos de relatório de condições de análise de diclorometano e ciclohexano podem ser vistos nas figuras 8 e 9.

Como não há espaço suficiente na memória para conter todo o cromatograma, o arquivo deste é inicialmente percorrido do início ao fim obtendo informações que preenchem um vetor de registros com três campos:

► Campo início: contém o número do registro onde o espectro se

- PROGRAMAS PARA OS SISTEMAS DE DADOS -

inicia. O valor da variável para o primeiro espectro é zero, para o segundo é dois mais o número de picos e assim por diante.

UFPA

Finnigan GC/MS

CLL

Amostra:

Operador:

Data: 07/07/90

CROMATOGRAFO

Vazao do gas de arraste: 1 ml/min

Temperatura do injetor: 330 oC

Temperatura do forno: 70 oC

Temperatura do separador: 330 oC

Temperatura inicial oC	Tempo inicial min	Velocidade oC/min	Temperatura final oC	Tempo final min	Tempo total min
70	3.00	0.10	88	5.00	188.00
					----- 188.00

Injetor/Separador	Ativa/Desativa	inicio (min)
I		0.20

ESPECTROMETRO

Janela de m/z: -0.4 d a 0.4 d

Linha base: 31

Desvios padroes da linha base: 1.000

Limiar: 21

m/z inicial d	m/z final d	inicio da aquisicao min	Numero de Conversoes 2 [^]
10	100	0.00	3

Figura 8 - Listagem das condições de análise de CH₂Cl₂.

- PROGRAMAS PARA OS SISTEMAS DE DADOS -

UFPA Finnigan GC/MS CLL

Amostra: hexano
 Operador: mh
 Data: 23/10/90

CROMATOGRAFO
 Vazao do gas de arraste: 1 ml/min
 Temperatura do injetor: 100 oC
 Temperatura do forno: 60 oC
 Temperatura do separador: 100 oC

Temperatura inicial oC	Tempo inicial min	Velocidade oC/min	Temperatura final oC	Tempo final min	Tempo total min
50	1.00	10.00	70	1.00	4.00
					----- 4.00

Injetor/Separador	Ativa/Desativa	inicio (min)
I	A	0.20

ESPECTROMETRO
 Janela de m/z: -0.4 d a 0.4 d
 Linha base: 30
 Desvios padroes da linha base: 1.000
 Limiar: 19

m/z inicial d	m/z final d	inicio da aquisicao min	Numero de Conversoes 2^
35	100	0.00	1

Figura 9 - Listagem das condições de análise de ciclohexano.

- PROGRAMAS PARA OS SISTEMAS DE DADOS -

►Campo tempo: contém o tempo decorrido até o início da aquisição do espectro.

►Campo tic: contém a soma das intensidades dos picos do espectro. Este valor corresponde a corrente iônica total (TIC).

Em uma segunda etapa o arquivo é novamente percorrido, obtendo-se o menor e o maior valor de m/z presente em todo o cromatograma.

De posse destes dados, a subrotina para tratamento gráfico permite manipular o cromatograma. As figuras 10 e 11 mostram exemplos de telas gráficas apresentadas para cromatogramas de diclorometano e ciclohexano - com respectivos relatórios nas figuras 8 e 9.

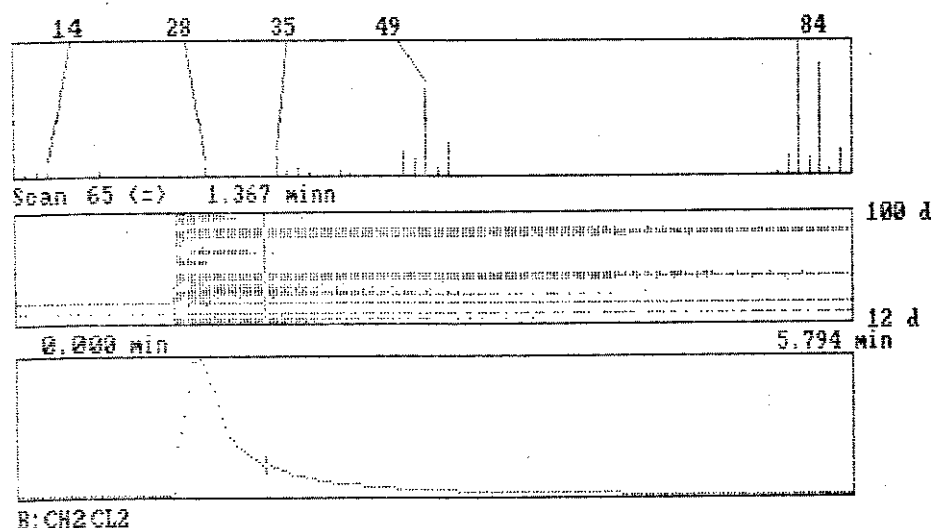


Figura 10 - Tela apresentada para o cromatograma de CH_2Cl_2 .

As três janelas apresentam visões de posições diferentes em relação ao cromatograma. A janela inferior apresenta o cromatograma de TIC em função do tempo. É o modo convencional de apresentação e é obtido através do campo tic do vetor preenchido no início. A janela do meio mostra a presença de picos de massa em função do tempo (mapa). Este cromatograma tem a mesma escala de tempo que a janela inferior. Caso seja desejado observar um

- PROGRAMAS PARA OS SISTEMAS DE DADOS -

espectro isoladamente, este pode ser selecionado por um cursor e apresentado na janela superior.

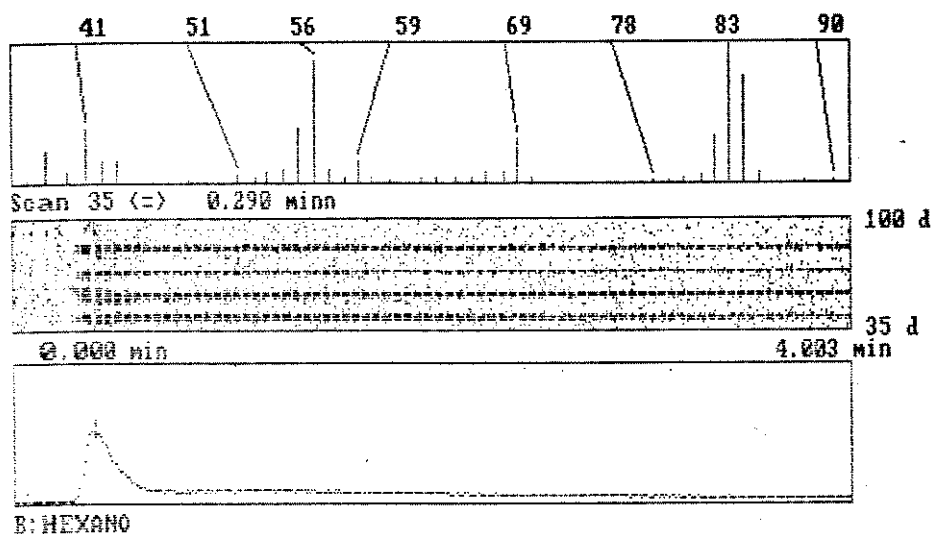


Figura 11 - Tela apresentada para o cromatograma de ciclohexano.

Na figura 10, o mapa mostra poucos picos antes da entrada do diclorometano no espectrômetro, sendo que os picos com m/z baixo, correspondentes ao gás residual, está presente em todo o cromatograma. É possível observar também que, mesmo depois de aproximadamente 6 minutos, ainda existia amostra na coluna. No início do pico, é identificado um conjunto de picos com m/z próximo de 100d que desaparece após algum tempo. Isto indica a presença de um contaminante, pois o íon molecular do diclorometano é 84d. A figura 11 mostra uma grande dispersão de pontos por todo o cromatograma, que certamente não pertencem a nenhum componente, sendo resultante do ruído. Existem duas possibilidades para a ocorrência destes picos espúrios: o número de conversões neste caso é 2 - para a análise do diclorometano é 8 - ou a linha base do espectrômetro não permaneceu constante - não foi aguardado o período de aquecimento do equipamento após ser ligado.

Além do cursor que permite caminhar sobre o cromatograma,

- PROGRAMAS PARA OS SISTEMAS DE DADOS -

existem outros recursos disponíveis:

►Pode ser feito o zoom do eixo de intensidade do cromatograma de TIC ou do eixo de tempo.

►O espectro sob o cursor pode ser salvo em disco, gerando arquivos SCANx.

►A tela pode ser impressa.

►Pode ser feito o cromatograma de ions. Este é um importante recurso que permite observação de até 10 valores de m/z simultaneamente em até 3 janelas. Um exemplo deste recurso pode ser visto na figura 12, que corresponde a mesma amostra apresentada na figura 10. Se a figura 10 deixa dúvidas quando a presença de contaminantes, neste caso é possível observar claramente que os picos em 83d e 84d, assim como em 61d e 62d não pertencem ao mesmo componente. Também é possível observar que o pico em 83d e 97d, por apresentarem perfis semelhantes, devem pertencer ao mesmo componente. Por fim, o pico em 49d parece ser um fragmento comum, pois seu perfil é semelhante a soma de 83d e 84d.

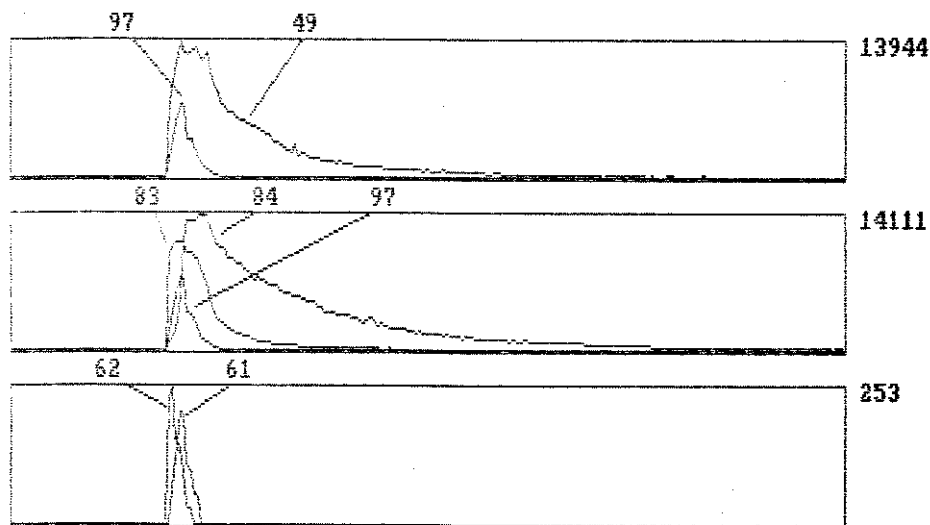


Figura 12 - Cromatograma de Ions de CH_2Cl_2 .

►Podem ser localizados os picos cromatográficos. Embora a

- PROGRAMAS PARA OS SISTEMAS DE DADOS -

maioria dos métodos para localizar picos estejam baseados em análise da primeira e segunda derivadas [1,2], é utilizado neste programa o centróide. Devido ao nível de ruído considerável e pequeno número de pontos, seria necessário um algoritmo eficiente, fazendo filtragem digital antes da análise de derivadas. Assim, foi proposto que este não seria o objetivo primordial do sistema criado.

O algoritmo desenvolvido calcula a média e o desvio padrão para os primeiros 30 pontos do cromatograma, que devem estar no tempo morto do cromatograma. A média é tomada como linha base e o desvio padrão como o limiar de detecção de um pico. O método não é eficiente, pois é necessário que o sinal retorne a linha base para separar dois picos, o que nem sempre é possível.

4 Bibliografia

- 1 - A.N. Papas, *CRC Crit. Rev. Anal. Chem.* 20, 359 (1989).
- 2 - F. Augusto, "Desenvolvimento e Aplicação de Software para Cromatografia Gasosa", tese de mestrado, Instituto de Química - UNICAMP, 1990.

CAPÍTULO V

ANÁLISE DE PADRÕES ISOTÓPICOS

A confirmação de fórmulas moleculares por espectrometria de massas é feita pela medida precisa de m/z do íon molecular, o que só é possível com aparelhos de alta resolução. Neste método, se aproveita o fato dos prótons e neutrons possuírem massas um pouco diferentes. Assim, por exemplo, é possível diferenciar com medidas precisas de massa um átomo de nitrogênio de um grupo CH_2 que grosseiramente apresentam massa 14.

Uma outra maneira de fazer a confirmação utiliza a intensidade dos picos dos vários isótopos de um íon. O método é razoavelmente mais complexo que o primeiro, pois envolve um número muito maior de cálculos. Entretanto, como a maioria dos espectrômetros de massas em uso são do tipo quadrupolar com resolução unitária, este método torna-se atrativo.

As intensidades dos picos de um espectro de massas não correspondem rigorosamente a abundância dos isótopos a que se referem [1]. Isto é devido a efeitos cinéticos e de equilíbrio isotópicos, *background*, discriminação do aparelho, etc. Apesar destes fatores que interferem na análise, vários autores [1,2,3,4] desenvolveram programas que geram os padrões isotópicos de uma fórmula conhecida para fazer a comparação visual entre estes e o padrão experimental.

Outra técnica matemática utilizada por Brauman [5] e por

- ANÁLISE DE PADRÕES ISOTÓPICOS -

McLaughlin e Rozett [6] é a redução do número de isótopos dos elementos, gerando espectros simulados. Isto é útil principalmente em estudos com organometálicos, pois alguns metais podem possuir até onze isótopos, dificultando a análise visual.

Todos estes métodos apresentam o inconveniente de deixar para o analista a tarefa de fazer a comparação visual, o que pode acarretar falhas devido ao cansaço e predisposição a indicar um resultado esperado. Assim, Bell [7] desenvolveu um programa mais sofisticado que gera todas as combinações possíveis de elementos entre C, H, N, O, S, F, Cl, Br e I que resultem em massa igual a do íon molecular e testa se as intensidades do padrão gerado estão dentro da margem de erro experimental. Este método apresenta o inconveniente de ser sensível a margem de erro que é imposta pelo operador. A listagem impressa pelo programa indica simplesmente se o padrão isotópico calculado passa pelo teste de margem de erro ou não.

O programa ELANAL desenvolvido por Kavanagh [8] apresenta uma solução mais razoável fornecendo um parâmetro indicador da semelhança entre o padrão experimental e o calculado. O parâmetro Q dado por

$$Q = \sum_{i=1}^n (I_{ri} - I_{ci})^2 \quad (1)$$

onde I_{ri} e I_{ci} são as intensidades dos n picos do padrão experimental e calculado, respectivamente, é tanto menor quanto maior a semelhança entre ambos.

Em todos estes métodos, é importante que a intensidade do conjunto dos picos do padrão seja alta para diminuir a possibilidade de erro devido a imprecisão dos dados experimentais. Este é um grande inconveniente, pois, a 70eV, é muito comum que o íon molecular seja pouco intenso. Para contornar este problema, Tenhosaari [9] utiliza o conjunto de várias análises dos diversos fragmentos além do íon molecular. Este procedimento é bastante

- ANÁLISE DE PADRÕES ISOTÓPICOS -

razoável, porém a análise de fragmentos é mais complexa pois pode ocorrer superposição de padrões de fragmentos, o que, se não for considerado, pode comprometer totalmente o resultado.

Palmer e Enke [10] desenvolveram programas para determinação de fórmula molecular empregando dados de espectros adquiridos em espectrômetros MS/MS. Estes equipamentos permitem eliminar parcialmente o problema de superposição de padrões.

1 Fundamentos e Implementação do Programa AC4

O programa para análise de padrões isotópicos desenvolvido neste trabalho [11] possui alguns pontos em comum com os descritos nos trabalhos de Bell [7], Kavanagh [8] e Tenhosaari [9]. Dados o conjunto de picos experimentais e as condições de análise, são gerados os padrões isotópicos de todas as composições contendo C, H, N, O, Si, P, S, F, Cl, Br e/ou I com massa compatível com este conjunto. Cada um destes padrões é comparado e uma listagem com os dez mais semelhantes é apresentada. Para a análise de fragmentos, são utilizados dois recursos que permitem eliminar o efeito do íon contaminante.

1.1 Condições de Análise

Além do conjunto dos picos do padrão isotópico, é necessário que algumas condições sejam estabelecidas pelo operador. Estas condições dizem respeito ao modo como foi adquirido o espectro e outras limitam o número de átomos de cada elemento, agilizando a análise.

Tendo conhecimento do modo como foi adquirido o espectro - equipamento, sensibilidade utilizada, modo de registro e tempo de aquisição - o operador estabelece as margens permitidas de erro de m/z e de intensidade para que as composições que serão geradas sejam consideradas compatíveis. Os valores iniciais são: $\pm 0,3d$ para m/z e erro relativo de $\pm 10\%$ da intensidade do pico, sendo que o menor valor de erro absoluto é $\pm 0,2$ (considerando o pico base como 100).

- ANÁLISE DE PADRÕES ISOTÓPICOS -

De posse do conjunto dos picos experimentais, o programa gera os números mínimo e máximo de cada elemento, permitindo que estes limites sejam alterados posteriormente pelo operador. Os limites são estabelecidos pelas seguintes considerações:

►Se tem-se a informação que a amostra é orgânica, é óbvio que o número mínimo de átomos de carbono é um. Assim, composições elementares que não contenham carbono não são consideradas.

►Pode-se afirmar que

$$m_1 \geq M_{X1} N_X \quad (2)$$

onde N_X é o número máximo de átomos do elemento X, m_1 é o valor de m/z do primeiro pico do padrão e M_{X1} é a massa do isótopo mais leve do elemento X. Se for considerada a situação onde a amostra é composta praticamente por este único elemento X, da equação 2 tem-se que

$$N_X = \text{Int} \left[\frac{m_1}{M_{X1}} \right] \quad (3)$$

►Para amostras orgânicas, a equação 2 pode ser reescrita como

$$m_1 \geq M_{X1} N_X + 12N_C \quad (4)$$

Considerando que existem no máximo quatro heteroátomos para cada átomo de carbono ($N_X = 4N_C$), a equação 3 pode ser reescrita como

$$N_X = \text{Int} \left[\frac{m_1}{M_{X1} + 3} \right] \quad (5)$$

►Hidrogênio e halogênios são tratados separadamente do item anterior. Devido a valência um, o número máximo de átomos é aquele

- ANÁLISE DE PADRÕES ISOTÓPICOS -

correspondente a cadeia saturada ($2N_C + 2$). Juntamente com a equação 4, tem-se que

$$N_X = \text{Int} \left[\frac{m_1 + 12}{M_{X_1} + 6} \right] \quad (6)$$

►Devido a grande abundância do isótopo ^{81}Br que equivale a 98% de ^{79}Br , o número máximo deste elemento é dado por

$$N_{\text{Br}} = \text{Int} \left[\frac{m_n - m_1}{2} \right] \quad (7)$$

onde m_n é a relação m/z do último pico do padrão.

►Quando um elemento estiver sujeito a mais de uma das restrições anteriores, o menor valor de número máximo de átomos será aceito.

1.2 Geração de Padrões Isotópicos

Após serem estabelecidas todas as condições de análise, o programa passa a gerar todas as composições possíveis dentro dos limites mínimo e máximo de cada elemento. O programa varia um a um o número de átomos, totaliza a massa para a composição utilizando o isótopo mais leve de cada elemento e compara com m_1 : se a composição apresentar massa superior a m_1 , é descartada; se for igual, é aceita; se for inferior, é complementada com átomos de carbono e hidrogênio dentro dos seus limites.

Para as composições aceitas são gerados os padrões isotópicos da mesma forma que Bell [7]. Cada elemento é considerado separadamente simplificando o cálculo de abundâncias isotópicas e em uma segunda etapa a contribuição para a massa e abundância de todos os elementos são computadas. Como a abundância de um isótopo depende da probabilidade de ocorrência, os cálculos usados para gerar o padrão isotópico são baseados em tratamentos

- ANÁLISE DE PADRÕES ISOTÓPICOS -

probabilísticos.

Os elementos fluor, fósforo e iodo são monoisotópicos e, portanto, não são necessários cálculos. Devido a baixa abundância natural do deutério (0,015% de ^2H), o hidrogênio é também considerado monoisotópico.

Para os elementos carbono, oxigênio e nitrogênio são utilizados dois isótopos. Devido a baixa intensidade do segundo isótopo mais abundante, são considerados no máximo três picos para o conjunto dos átomos de um elemento. Esta aproximação é válida desde que o número de átomos não seja elevado. As intensidades dos três íons são dadas por:

$$I_1 = A_1 \cdot a \quad (8)$$

$$I_2 = A_2 \cdot a \cdot A_1^{a-1} \quad (9)$$

$$I_3 = 1 - I_1 - I_2 \quad (10)$$

onde I_i é a intensidade do pico i , A_j é a abundância natural do isótopo j e a é o número de átomos do elemento.

A massa M em daltons associada com cada um destes picos é dada por

$$M_{i+1} = m_1 \cdot (a-i) + m_2 \cdot i \quad \text{para } i=0 \text{ até } 2 \quad (11)$$

onde m_j é a massa do isótopo j .

Não é possível limitar o número máximo de picos para cloro e bromo devido a alta abundância natural do segundo isótopo (^{37}Cl é 32,5% de ^{35}Cl). Para estes elementos, as contribuições para intensidade e massa são dadas por:

$$I_{i+1} = A_1^{(a-i)} \cdot A_2^i \frac{a!}{(a-i)!i!} \quad \text{para } i=0 \text{ até } a \quad (12)$$

$$M_{i+1} = m_1 \cdot (a-i) + m_2 \cdot i \quad (13)$$

- ANÁLISE DE PADRÕES ISOTÓPICOS -

Os cálculos para silício e enxofre, que apresentam três isótopos significativos, são análogos a aqueles para cloro e bromo:

$$I_{1+i+j} = A_1^{(a-i-j)} \cdot A_2^j \cdot A_3^i \frac{a!}{(a-i-j)!j!i!} \quad (14)$$

$$M_{1+i+j} = m_1 \cdot (a-i-j) + m_2 \cdot j + m_3 \cdot i \quad (15)$$

para $i=0$ até a e para $j=0$ até $a-i$.

Depois de estabelecidos os padrões isotópicos para os conjuntos de átomos de cada elemento, a próxima etapa é combiná-los para gerar o padrão para a composição elementar. A massa para cada combinação é dada por

$$M = \sum M_{ix} \quad (16)$$

onde M_{ix} é a contribuição para a massa do pico i do elemento X e a soma é estendida para todos os elementos presentes na composição. Todas as combinações são geradas variando i de 1 até o último pico do padrão do elemento X .

A probabilidade de ocorrência simultânea de condições é o produto das probabilidades individuais de ocorrência. Assim, a intensidade de cada pico será o produto das intensidades individuais, isto é,

$$I = \prod I_{ix} \quad (17)$$

onde I_{ix} é a contribuição para a intensidade do pico i para o elemento X e o produto é estendido para todos os elementos da composição. Como a probabilidade de ocorrências alternadas é a soma das probabilidades individuais, as intensidades das combinações de mesma massa são somadas.

O exemplo a seguir mostra como o padrão isotópico é calculado

- ANÁLISE DE PADRÕES ISOTÓPICOS -

para o dissulfeto de carbono (CS_2). Usando as equações 14 e 15, é obtido o padrão para dois átomos de enxofre. Este é combinado com o de um carbono. O resultado, apresentado na tabela 1, mostra que duas combinações resultam em massa 77. Isto também ocorre para 78, 79 e 80. Na última etapa, as intensidades para as combinações que resultam em massas iguais são adicionadas, obtendo a intensidade total daquele pico. O padrão isotópico resultante é mostrado na tabela 2.

C \ S ₂	64	65	66	67	68
	0,90307	0,01444	0,07988	0,00064	0,00176
12	76	77	78	79	80
0,98890	0,89305	0,01428	0,07899	0,00063	0,00174
13	77	78	79	80	81
0,01110	0,01002	0,00016	0,00089	0,00001	0,00002

Tabela 1 - Combinação de padrões isotópicos de um átomo de carbono com dois átomos de enxofre. O valor superior é o valor aproximado de massa e o inferior é a abundância. O padrão resultante é obtido pela soma das abundâncias de combinações de mesma massa.

massa	intensidade
76	0,89305
77	0,02430
78	0,07915
79	0,00152
80	0,00175
81	0,00002

Tabela 2 - Padrão calculado através dos dados da tabela 1 para a molécula de CS_2 .

1.3 Comparação de Padrões Isotópicos

O conjunto de picos pode ser representado graficamente como um ponto em um sistema de coordenadas. Um conjunto de dois picos pode ser localizado em um espaço bidimensional, três picos, no

espaço tridimensional e assim por diante.

O conjunto de picos do padrão isotópico é normalizado para que a soma das intensidades seja um:

$$\sum_{i=1}^N I_i = 1 \quad (18)$$

onde I_i é a intensidade do pico i de um total de N picos do conjunto.

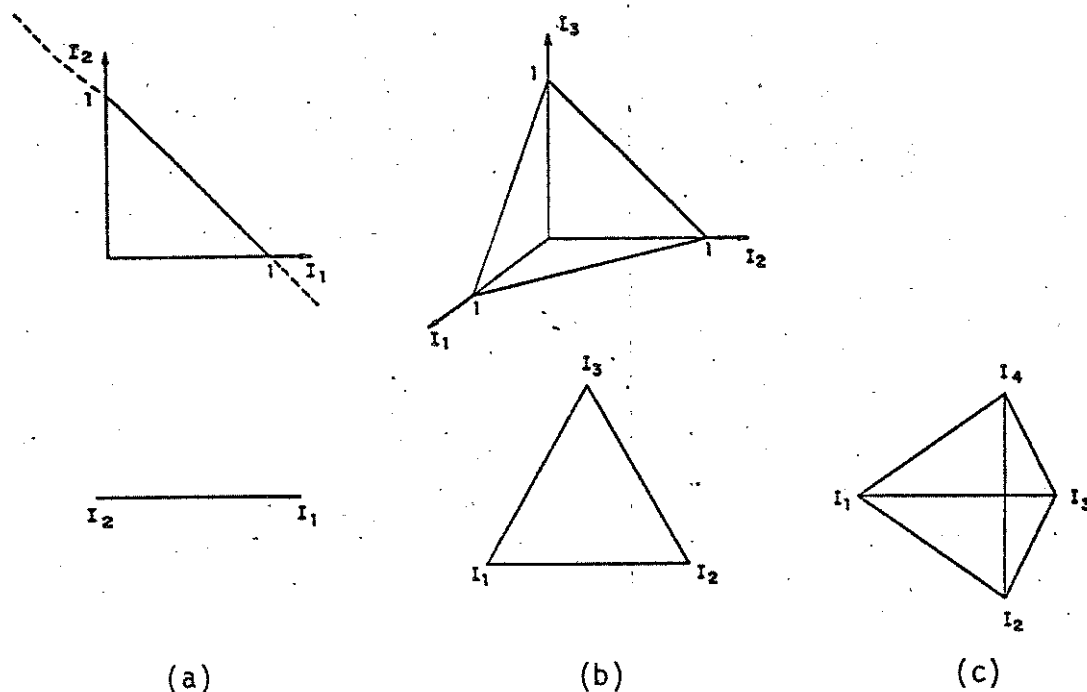


Figura 1 - Representação gráfica para conjuntos de picos. Um conjunto de dois picos pode ser representado como um ponto sobre um segmento de reta em (a). Um conjunto de três picos pode ser representado como um ponto sobre o triângulo em (b). Não é possível visualizar o sistema de coordenadas para um conjunto de quatro picos, porém, este pode ser localizado como um ponto no interior do tetraedro em (c).

Isto leva à reta $I_1 + I_2 = 1$ e ao plano $I_1 + I_2 + I_3 = 1$, para N igual a dois e três, respectivamente. Outra restrição é que não existem picos com intensidades negativas, isto é, $I_i \geq 0$. Desta forma, para dois picos, o intervalo permitido é o segmento de reta mostrado na

- ANÁLISE DE PADRÕES ISOTÓPICOS -

figura 1a e para três picos, o triângulo mostrado na figura 1b. Para quatro picos, não é possível visualizar o eixo de coordenadas, mas as restrições formam um tetraedro (figura 1c). Assim, para N picos, estas restrições produzem um politopo (N-1)-simplex, que é o equivalente do triângulo em duas dimensões e do tetraedro em quatro dimensões. É importante notar que existe um vértice do politopo para cada pico do padrão.

Uma vez calculado o padrão isotópico para uma composição, a diferença ou distância Euclidiana entre o conjunto experimental e calculado pode ser determinada por

$$D = \left[\sum_{i=1}^N (I_{ri} - I_{ci})^2 \right]^{\frac{1}{2}} \quad (19)$$

onde N é o número de picos experimentais, I_{ri} é a intensidade experimental e I_{ci} é a intensidade calculada do pico i. Este valor é utilizado como parâmetro de comparação: quanto menor a distância, mais semelhantes são os padrões isotópicos.

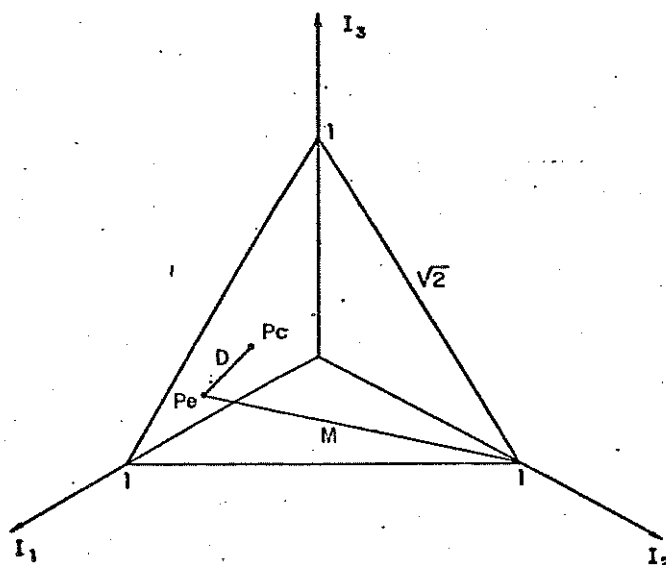


Figura 2 - Representação gráfica para um conjunto de três picos onde Pe é o conjunto de picos experimental e Pc é um conjunto calculado. O segmento D é a distância entre Pe e Pc e M é a distância máxima.

Devido ao conjunto de restrições, a diferença máxima entre dois pontos é a raiz quadrada de dois, que é a distância entre dois pontos extremos ou vértices do politopo (figura 2). Como o padrão experimental nunca está posicionado no vértice, a distância máxima será sempre menor que este valor.

Dado o conjunto de picos experimentais, um dos vértices será o ponto que representa o padrão isotópico mais diferente. Assim, a distância entre um conjunto experimental e o vértice j é dada por

$$D_j = \left[1 - 2I_{rj} + \sum_{i=1}^N I_{ri}^2 \right]^{\frac{1}{2}} \quad (20)$$

Pode-se observar facilmente que D_j é máxima quando I_{rj} é mínima. Portanto, para determinar a distância máxima basta localizar o pico experimental de menor intensidade e calcular a distância entre o conjunto experimental e o vértice correspondente a este pico.

O parâmetro de comparação é então definido como

$$d = 100 D/M \quad (21)$$

onde d é a distância relativa, D é a distância absoluta e M é a distância máxima ou $\text{MAX}(D_j)$.

1.4 Análise de Misturas de A com A-1

Quando ocorre a fragmentação de um íon com perda de hidrogênio radicalar, é formado um novo íon com massa um dalton abaixo da massa do íon original. Devido a baixa abundância natural do deutério, o hidrogênio é praticamente monoisotópico e, portanto, o padrão isotópico do íon resultante é praticamente o mesmo do original. O exemplo mostrado na figura 3 pode ser descrito pelas equações:

- ANÁLISE DE PADRÕES ISOTÓPICOS -

$$c_1 = b_1 \quad (22)$$

$$c_i = b_i + a_{i-1} \quad \text{para } i=2 \text{ até } n-1 \quad (23)$$

$$c_N = a_{N-1} \quad (24)$$

onde a_i é o pico i do padrão A, b_i é o pico i do padrão A-1, c_i é o pico i do padrão isotópico da mistura resultante com N picos. Os picos do padrão A-1 podem ser substituído por

$$b_i = x a_i \quad (25)$$

onde x é a relação entre A-1 e A. Destas considerações, o seguinte polinômio é obtido:

$$\sum_{i=1}^N c_i (-x)^{i-1} = 0 \quad (26)$$

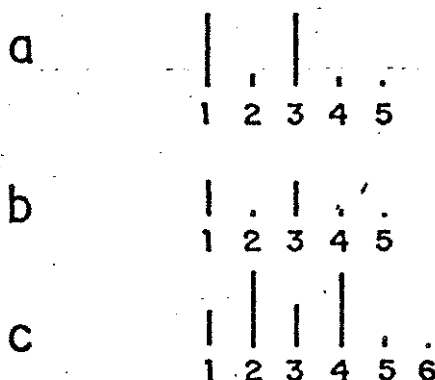


Figura 3 - Padrão isotópico do íon molecular M (a) e do fragmento M-1 (b) de $C_9H_9O_2Br$. O padrão (c) corresponde a região do íon molecular se M-1 tiver intensidade 50% de M.

Assim, o fator x é uma das raízes do polinômio. Na prática o valor de x deve estar entre 0,01 e 100. A determinação das raízes é feita pelo método iterativo de Birge-Vieta [12] e todos os valores encontrados dentro do intervalo acima são empregados na

análise.

A análise do padrão isotópico é feita normalmente adicionando-se uma nova restrição: o número mínimo de átomos de hidrogênio é um.

1.5 Eliminação de Padrões Contaminantes

Algumas vezes, formam-se íons de diferentes composições, mas com massas iguais ou próximas. Isto acarreta a sobreposição de seus padrões isotópicos, podendo inviabilizar a análise. Para estes casos foi incluída uma subrotina capaz de, uma vez conhecida a composição elementar de um dos íons, fornecer o padrão isotópico do outro.

Com a composição elementar do contaminante conhecida, o programa gera seu padrão isotópico e tenta subtrair-lo do padrão experimental. Esta eliminação é feita comparando pico a pico para verificar que proporção de contaminante pode ser subtraída sem gerar picos com intensidades negativas.

Para ser possível esta eliminação, é necessário considerar que existe ao menos um pico que seja exclusivo do contaminante. A proporção do contaminante na amostra é a menor relação (g_i) entre as intensidades dos picos experimental (I_{ri}) e contaminante (I_{ci}):

$$g_i = \frac{I_{ri}}{I_{ci}} \quad (27)$$

O fator g_i é calculado para cada pico do contaminante e é escolhido o de menor valor, pois assim fica garantida a não formação de picos negativos.

Analisando uma mistura simulada de cloreto de metileno e ciclohexano na proporção de 1:1 é possível verificar como o método se processa. A tabela 3 mostra o padrão isotópico na região do íon molecular das duas substâncias e da mistura, além dos valores de g para duas situações: Na primeira, o ciclohexano é o contaminante e na segunda, o cloreto de metileno é o contaminante.

- ANÁLISE DE PADRÕES ISOTÓPICOS -

Como a condição de que haja um pico exclusivo do contaminante não é respeitada no primeiro caso, os valores de g são sempre maiores que o valor ideal (0,5). Isto provoca a eliminação excessiva, zerando o pico de m/z 85. Isto não ocorre no segundo caso pois o cloreto de metileno possui três picos exclusivos: 87, 88 e 89. Neste segundo caso o valor de g mínimo é 0,50002 e portanto, bastante próximo do ideal.

m/z d	mistura 1:1	C_6H_{12}		CH_2Cl_2	
		intens.	g	intens.	g
84	0,75148	0,93522	0,80353	0,56774	1,32363
85	0,03467	0,06298 *	0,55053	0,00637	5,44270
86	0,18245	0,00179	101,927	0,36311	0,50247
87	0,00204			0,00408	0,50053
88	0,02903			0,05806 *	0,50002
89	0,00033			0,00065	0,50769

Tabela 3 - Mistura simulada de cloreto de metileno e ciclohexano na proporção de 1:1. Os picos marcados com * possuem o valor de g mínimo.

Se, durante a aquisição do espectro, a linha base não fosse corretamente determinada, seria possível ocorrer uma situação onde ouvesse a diminuição das intensidades. Se a intensidade do pico de m/z 89 caísse para 0,00030, o valor de g seria por volta de 0,46 enquanto o pico de m/z 88 continuaria indicando o valor de 0,50. É por este motivo que picos de pequena intensidade não são considerados na determinação de g .

2 Parte Experimental

O programa foi escrito em Turbo Pascal 3.0 da Borland Inc. e aproximadamente cem análises foram feitas usando duas fontes de dados: espectros em papel fotossensível adquiridos do espectrômetro MAT311A e espectros do "Atlas of Mass Spectral Data" [13]. Foram feitas análises com os limites de números de átomos estipulados pelo programa e outras com limites estipulados pelo operador - sempre duas vezes o número de átomos da composição

- ANÁLISE DE PADRÕES ISOTÓPICOS -

elementar correta. A figura 4 mostra um exemplo de relatório de análise exatamente como é fornecido pelo programa.

3 Avaliação do Programa

A tabela 4 sumariza os resultados de análises para onze amostras: as nove primeiras foram obtidas da referência 13 e as duas últimas de espectros do espectrômetro MAT311A. Para fins de comparação, as cinco primeiras amostras são as mesmas utilizadas em estudo prévio por Kavanagh [8]. Para ambos os programas, as composições elementares corretas ocupam aproximadamente as mesmas posições nas listas dos relatórios.

Em geral, a imposição de limites dos átomos pelo operador resulta em melhores resultados. Por exemplo, na análise de $C_6H_{14}OS_2$ (análise 6) com limites *default*, a composição correta ocorre em sexto lugar e o tempo de análise é de 1187 segundos. Com limites estipulados pelo operador, que geralmente conhece o histórico da amostra, para o dobro de átomos de cada elemento, a composição correta aparece em terceira posição após 8 segundos (análise 7). Na análise de 1-iodobutano os resultados são mais drásticos. Neste caso, a composição correta não aparece entre os dez mais semelhantes quando são utilizados os limites *default* (análise 8). Entretanto, com a interferência do operador nos limites (análise 9), a composição correta ocupa o primeiro lugar. Assim, impondo limites mais realistas baseado em conhecimento prévio da amostra, o resultado será mais preciso e o tempo de análise menor.

Outro fator que afeta o resultado é a precisão das medidas de intensidades. Quando são utilizadas mais de uma escala de um espectro registrado em papel fotográfico, as intensidades devem ser corrigidas pois a relação entre as escalas não é exatamente dez. Por exemplo, sem correção, a composição elementar de $C_6H_7O_2N_2SBr$ ocorre em quarto lugar (análise 10) e com correção, em primeira posição (análise 11). Quando é utilizado um espectro registrado por um sistema de dados, o principal problema é

- ANÁLISE DE PADRÕES ISOTÓPICOS -

determinar corretamente a linha base e o limite de detecção de picos.

Relatorio de Analise de Cluster .

Programa AC versao 4.4 (CLL)

Identificacao: C9H9O2Br #28

Cluster apos subtrair M - H:

m/z	Int. Rel.	Abundancia
228	25.31%	0.45743
229	2.52%	0.04550
230	24.81%	0.44841
231	2.49%	0.04505
232	0.20%	0.00361

CONDICOES:

Intensidade: Erro Relativo = 10.0% e Erro Absoluto = 0.2

Erro em m/z = 0.30

Composicao: min. 1H

max. 19C 34H 14O 5Cl 2Br

Tempo de analise: 24 segundos.

COMPOSICOES +.

#	Diferenca GI	Ion Composicao
1	0.244%	5 +. C ₉ H ₉ O ₂ Br
2	0.699%	6 +. C ₈ H ₅ O ₃ Br
3	0.856%	4 +. C ₁₀ H ₁₃ OBr
4	1.564%	3 +. C ₁₁ H ₁₇ Br
5	1.616%	11 +. C ₁₁ HOBr

Figura 4 - Relatório da análise da região do íon molecular de C₉H₉O₂Br após a remoção do fragmento M-1, que foi determinado como tendo 3,6% da intensidade de M.

- ANÁLISE DE PADRÕES ISOTÓPICOS -

#	Substância	Fórmula	Ordem AC4	Ordem ELANAL	d	tempo s
1	Ác. p-hidroxibenzóico	$C_7H_6O_3$	7	5	0,707	41
2	3,4,7-Trimetilindanona	$C_{12}H_{14}O$	2	2	0,190	104
3	Salicilaldeído	$C_7H_6O_2$	1	1	0,057	187
4	Oxiclorodifluoreto de fósforo	$POClF_2$	1	2	0,049	488
5	Isocianato de isobutila	C_5H_8NS	1	1	0,102	84
6	Éter bis-tioetóxi-dimetílico	$C_6H_{14}OS_2$	6		1,542	1187
7	(idem 6)		3		1,542	8
8	1-Iodobutano	C_4H_9I	>10		>0,25	142
9	(idem 8)		1		0,291	<1
10	p-Bromofenilsulfonil-hidrazina	$C_6H_7O_2N_2SBr$	4		1,484	42
11	(idem 10)		1		0,481	42

Tabela 4 - Resultados de análises de padrões isotópicos na região do íon molecular. Os espectros de 1 a 9 foram obtidos da referência 13. Os espectros 10 e 11 foram registrados em papel fotossensível no espectrômetro MAT311A. O tempo de processamento se refere ao microcomputador utilizado no sistema de dados do IQ-UNICAMP.

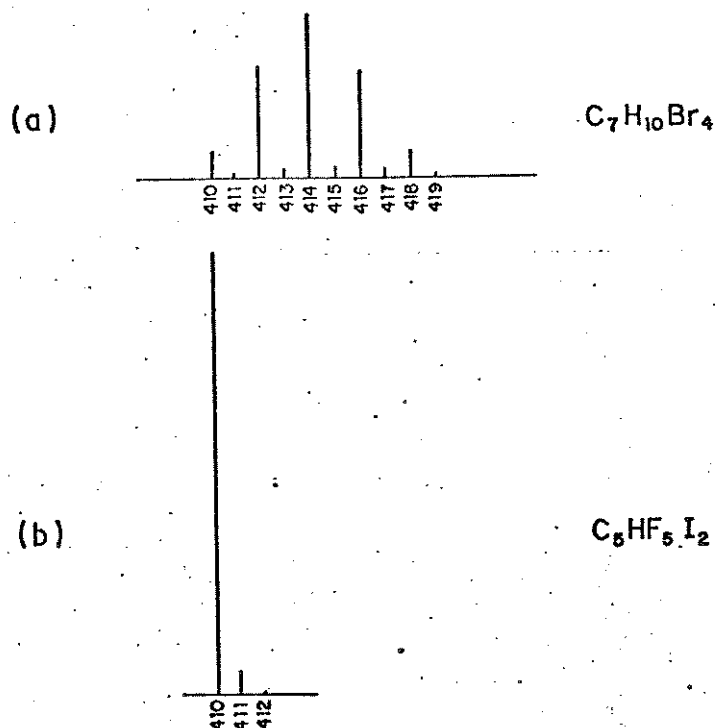


Figura 5 - Espectros simulados para os íons moleculares de $C_7H_{10}Br_4$ e $C_5HF_5I_2$. A soma das intensidades de ambos são iguais.

- ANÁLISE DE PADRÕES ISOTÓPICOS -

Dos trabalhos citados, somente os de Kavanagh [8] e Tenhosaari [9] usam um parâmetro de comparação entre os padrões experimental e calculado. Pode-se comparar estes parâmetros na tabela 5 tomando como referência o parâmetro Q (equação 1). Uma análise destes parâmetros mostra que a lista de composições elementares similares deve ser a mesma independente do parâmetro usado. Entretanto, um exemplo mostra que quantitativamente existem diferenças. Se for comparado o padrão isotópico de $C_7H_{10}Br_4$ (figura 5a) com $C_5HF_5I_2$ (figura 5b), são obtidos os valores apresentados na terceira coluna da tabela 5. Na quarta coluna são apresentados os intervalos dos parâmetros de comparação. Este exemplo representa uma situação extrema encontrada na prática, pois fluor e iodo são monoisotópicos enquanto o bromo apresenta um segundo isótopo muito intenso. O parâmetro Q sugere que a comparação está a apenas 50% da escala. O parâmetro MV é mais realista alcançando 71% e por fim o parâmetro d que atinge 91% da escala. Isto mostra que o parâmetro d é o mais razoável, pois não só ordena as composições como fornece informação quantitativa mais precisa sobre a comparação.

Programa	Parâmetro de Comparação	Resultado $C_7H_{10}Br_4 / C_5HF_5I_2$	Intervalo possível melhor caso	pior caso
Ref. 8	Q	1,02	0	2
Ref. 9	$MV = 100(1 - \sqrt{Q})$	-0,923	100	-41
AC4	$d = 100 \left[\frac{Q}{Q_{max}} \right]^{\frac{1}{2}}$	90,8	0	100

Tabela 5 - Relação entre os parâmetros de comparação. Os resultados das análises simuladas permitem uma comparação entre estes.

A figura 4 mostra o relatório de análise do íon molecular de $C_9H_9O_2Br$ que estava contaminado com o fragmento M-1. O programa determinou a proporção de contaminante em 3,6% de M. Após a eliminação do íon M-1, a composição correta aparece em primeiro

lugar com d igual a 0,244. A análise do íon original apresentou a composição correta apenas em quarto lugar com d igual a 1,433.

Em conclusão, este programa é uma ferramenta útil na interpretação de espectros de massas e serve de base para desenvolvimento de programas mais sofisticados de interpretação automática. Ele pode ser usado para eliminar superposição de padrões isotópicos - um problema comum na análise de fragmentos.

4 Bibliografia

- 1 - R.W. Rozett, *Anal. Chem.* 46, 2085 (1974).
- 2 - B.D. Dombek, J. Lowther and E. Carberry, *J. Chem. Educ.* 48, 729 (1971).
- 3 - C.S. Hsu, *Anal. Chem.* 56, 1356 (1984).
- 4 - J.A. Yergey, *Int. J. Mass Spectr. Ion Phys.* 52, 337 (1983).
- 5 - J.I. Brauman, *Anal. Chem.* 38, 607 (1966).
- 6 - E. McLaughlin and R.W. Rozett, *J. Organomet. Chem.* 52, 261 (1973).
- 7 - H.M. Bell, *J. Chem. Educ.* 51, 548 (1974).
- 8 - P.E. Kavanagh, *Org. Mass Spectr.* 15, 334 (1980).
- 9 - A. Tenhosaari, *Org. Mass Spectr.* 23, 236 (1988).
- 10 - P.T. Palmer and C.G. Enke, *Int. J. Mass Spectr. Ion Proc.* 88, 81 (1989).

- ANÁLISE DE PADRÕES ISOTÓPICOS -

- 11 - G.L. do Lago and C. Kascheres, "New Method of Isotope Pattern Analysis", *Computers & Chemistry*, no prelo.
- 12 - V.R.B. Santos, "Curso de Cálculo Numérico", Livros Técnicos e Científicos, Rio de Janeiro, 3ª Ed., 1977, p. 73.
- 13 - E. Stenhagen, S. Abrahamson and F.W. McLafferty, "Atlas of Mass Spectral Data", Interscience Publishers, New York, 1969.

CAPÍTULO VI

ANÁLISE FATORIAL

Um problema enfrentado por quem tenta utilizar a espectrometria de massas na identificação ou elucidação de estruturas é a presença de substâncias contaminantes da amostra. Isto ocorre também com diversas outras técnicas como espectroscopia de infravermelho, RMN, etc. Este é um problema tão comum que geralmente se utilizam métodos de separação física - como é o caso da cromatografia - antes da análise propriamente dita. Ocorre que nem sempre é possível separar a amostra em seus componentes para então analisá-los. Para contornar este problema foram desenvolvidas técnicas matemáticas que permitam isolar - a partir de um conjunto de dados da amostra - os dados de cada um dos componentes. No caso de análises espectrométricas são necessários espectros de misturas contendo proporções diferentes dos componentes. Isto é possível utilizando-se picos cromatográficos pouco resolvidos, produtos pirolisados ou mesmo adicionando-se um padrão.

A Análise Fatorial é uma técnica matemática para resolução de problemas multidimensionais que, utilizada inicialmente em Ciências Sociais e Psicologia, passou a ser aplicada à Química e à Física há poucas décadas [1,2]. A utilização desta técnica em espectrometria de massas visava inicialmente a determinação do número de componentes da mistura [3,4]. Porém, com o

desenvolvimento de outros métodos [5,6], foi possível a obtenção de informações sobre os espectros dos componentes puros e das concentrações destes nas misturas. O método utilizado aqui é baseado na descrição dada por Knorr e Futrell [5] e Malinowski [6,7].

1 Fundamentos Utilizados no Programa FA3

Os espectros das misturas podem ser tabelados na forma matricial, onde cada coluna representa um espectro, cada linha, um valor de m/z e cada elemento desta matriz, a intensidade do pico. Esta matriz de dados pode ser expressa por

$$D = RC \quad (1)$$

onde D é a matriz de dados com r valores de m/z por c misturas, R é a matriz dos espectros dos componentes puros com r valores de m/z por n componentes e C é a matriz das concentrações com n componentes por c misturas. O problema consiste em encontrar as matrizes R e C , tal que a equação acima seja satisfeita.

Como existe uma equação e duas incógnitas, o sistema permite infinitas soluções, inclusive C igual à matriz identidade (I), o que resultaria em $D = R$. Isto corresponderia a dizer que a matriz de dados D é formada por c espectros de compostos puros e não por misturas. Assim, torna-se necessário estipular que a solução procurada deva ser aquela que consiga reproduzir a matriz de dados utilizando o menor número de componentes.

Para isto é formada a matriz covariância Z :

$$Z = D^t D \quad (2)$$

Assim preparada, a matriz Z possui em cada um de seus elementos a correlação existente entre todos os espectros experimentais tomados dois a dois. A próxima etapa é calcular os autovalores e autovetores, que têm a seguinte propriedade

$$ZQ_j = \lambda_j Q_j \quad (3)$$

onde Q_j é um autovetor e λ_j é um autovalor.

Com este procedimento é possível determinar o número de componentes. Ritter e outros [3] tentam recompor a matriz Z com o menor número possível de autovalores e autovetores. Inicialmente é utilizado somente o primeiro autovetor e seu correspondente autovalor na recomposição. Caso os elementos da matriz obtida não estejam dentro da margem de erro do valor experimental, o segundo autovetor e autovalor são acrescentados à reposição e assim por diante. No final, o número de autovalores é igual ao número de componentes. Este método torna-se muito lento com o aumento das dimensões da matriz e do número de componentes.

Malinowski [4] desenvolveu, baseado em cálculos de erros, outros métodos, dois dos quais são utilizados neste trabalho:

$$RE = \left[\frac{\sum_{j=n+1}^c \lambda_j}{r(c-n)} \right]^{1/2} \quad (4)$$

$$IND = \frac{RE}{(c-n)^2} \quad (5)$$

A função RE indica a diferença entre a matriz de dados e a matriz recomposta com n autovalores. A função Erro Real (RE) apresenta uma queda brusca quando o número de componentes é atingido enquanto a função indicadora (IND) apresenta um ponto de mínimo.

Neste ponto, parte do problema da análise fatorial está resolvida: a determinação do número de componentes. Resta agora encontrar os espectros destes componentes.

Em outro trabalho [7], o mesmo autor demonstra que a matriz

de autovetores Q é uma representação válida da matriz C . Assim, a partir da equação 1, a matriz R é dada por $R = DQ^t$. Deve-se observar que a matriz Q não é utilizada em sua totalidade, mas somente com os n mais importantes autovetores. Estes valores encontrados para R e C representam uma solução matemática abstrata para o problema, pois, em geral, não pode ser diretamente identificada com seu significado físico. Estas matrizes, que podem até possuir concentrações e picos negativos, serão designadas pelo índice a .

Para solucionar este novo problema, é introduzida na equação $D = R_a C_a$ a matriz de transformação T [6], tal que

$$D = R_a T T^{-1} C_a \quad (6)$$

Assim, são encontradas finalmente as matrizes R e C pelas equações

$$R = R_a T \quad (7)$$

$$C = T^{-1} C_a \quad (8)$$

Esta matriz T é formada por um conjunto de n linhas da matriz R_a normalizada, porém seu significado físico está associado à necessidade de que cada componente possua um pico que nos outros seja de intensidade muito menor (ponto de massa pura [5]).

Primeiramente, é formada uma matriz V que é R_a com as linhas normalizadas para que o somatório dos quadrados dos elementos seja um. Knorr e Futrell [5] sugerem que para formar T seja inicialmente escolhida a linha com a menor contribuição na primeira coluna, ou seja, no primeiro autovetor. A segunda linha escolhida será aquela com a maior diferença entre a segunda coluna de ambas e assim por diante. Este método é eficiente quando se tem até 3 componentes, porém não apresenta bons resultados para misturas mais complexas.

Malinowski [6] aprimorou este método, escolhendo a primeira linha da mesma forma e as outras são escolhidas de modo a formarem

a matriz mais ortogonal possível. Isto é conseguido com o cálculo de determinante. Por exemplo, para se escolher a quarta linha, são tomadas as três primeiras já encontradas e as linhas restantes são acrescentadas uma por vez como quarta linha. É feito o cálculo do determinante das matrizes 4x4 desprezando-se as colunas restantes. Quanto mais próximo de 1 for o determinante, maior a ortogonalidade.

Os espectros contidos na matriz R podem ser normalizados do modo tradicional - com pico base igual a 100 - desde que o vetor concentração correspondente ao componente seja multiplicado pelo inverso do fator de normalização.

Outro aspecto que deve ser observado é que a matriz C não pode ser utilizada diretamente para comparações precisas de concentrações entre diferentes componentes. O seu valor real depende de fatores como a secção de ionização específica de cada componente e, portanto, é necessária uma curva de calibração para determinações quantitativas. Este é um problema inerente à técnica de espectrometria de massas.

2 Avaliação do Programa FA3

O programa FA3 parte da matriz de dados experimentais, determina o número de componentes automaticamente e apresenta como resultado as matrizes de concentração e componentes já normalizadas. Para isto foram desenvolvidas subrotinas de multiplicação, inversão, momento, transposição, determinante e cálculos de autovalores e autovetores de matrizes.

O algoritmo para calcular os autovalores é o sugerido por Ortega [8]. O cálculo de determinante é feito através do desenvolvimento de Laplace em um processo recursivo, que apesar de simples torna-se lento com o aumento das dimensões das matrizes.

O programa foi utilizado na resolução de várias misturas simuladas e reais, inclusive para a mistura de ciclohexano e ciclohexeno obtida experimentalmente por Ritter e outros [3] (tabela 1), com resultados bastante próximos aos obtidos por Knorr

- ANÁLISE FATORIAL -

e Futrell [5] (tabela 2). O mesmo sucesso foi alcançado na análise do conjunto de dados simulados propostos por Malinowski [6].

m/z	Ciclohexano			
	80%	60%	40%	20%
27	2,3	3,2	3,4	2,1
28	1,2	1,3	1,3	0,7
29	1,1	1,1	1,1	0,5
39	3,9	5,8	6,8	4,9
40	0,7	1,0	1,0	0,6
41	8,6	10,5	10,5	6,0
42	3,5	3,7	3,1	1,1
43	1,6	1,7	1,4	0,4
51	0,5	1,1	1,5	1,0
53	0,9	1,7	1,7	1,7
54	3,6	8,0	11,3	10,1
55	5,1	5,4	4,5	1,9
56	14,2	14,4	11,4	3,6
67	4,4	11,1	16,0	15,1
68	0,5	0,8	1,1	0,8
69	4,1	4,3	3,3	1,0
79	0,4	0,8	1,2	1,0
81	0,5	1,0	1,6	1,4
82	1,5	4,1	6,1	5,4
84	10,5	11,8	8,5	2,4

Tabela 1 - Espectros das misturas de ciclohexano e ciclohexeno preparadas por Ritter e outros. [9].

O programa foi também utilizado na resolução de picos cromatográficos. Durante os testes do sistema para GC/MS, foi obtido o cromatograma do cloreto de metileno, mostrado na figura 1.

- ANÁLISE FATORIAL -

m/z d	Componente 1 (Ciclohexeno)		Componente 2 (Ciclohexano)	
	FA3	[5]	FA3	[5]
27	11,6	11	13,3	13
28	3,0	3	7,3	7
29	1,8	2	6,9	7
39	28,9	28	18,8	18
40	3,2	3	4,3	4
41	28,9	29	52,0	52
42	2,3	2	24,3	24
43	0,5	0	11,5	11
51	7,1	7	2,1	2
53	9,6	9	3,4	3
54	66,4	66	5,0	5
55	4,7	5	34,4	34
56	2,1	2	100,0	100
67	100,0	100	0,1	0
68	5,1	5	1,9	2
69	0,4	0	29,4	29
79	6,7	6	0,7	0
81	9,3	9	0,4	0
82	36,9	37	0,0	0
84	0,0	0	78,4	78

Tabela 2 - Comparação entre os resultados obtidos pelo programa FA3 e por Knorr e Futrell [5]. Em ambos os casos os espectros foram normalizados para pico base igual a 100.

Como as condições cromatográficas não são as ideais, dois ou mais componentes estão presentes no único pico. O mapa do cromatograma (segunda janela) mostra um conjunto de picos na

- ANÁLISE FATORIAL -

região próxima a 100d que poderia corresponder a picos de baixa intensidade, presentes somente na região do máximo cromatográfico, não se tratando necessariamente de um contaminante. Porém, a observação dos espectros nas regiões anteriores e posteriores ao máximo, mostra uma flutuação das intensidades dos picos na região de m/z 84d. Esta flutuação pode ser observada na figura 2, onde se nota nesta região a presença de dois padrões isotópicos contendo cloro. O padrão com pico mais intenso em 84d deve corresponder a CH_2Cl_2^+ . O padrão em 83d deve corresponder a CHCl_2^+ , proveniente da fragmentação de outra espécie clorada além do CH_2Cl_2 .

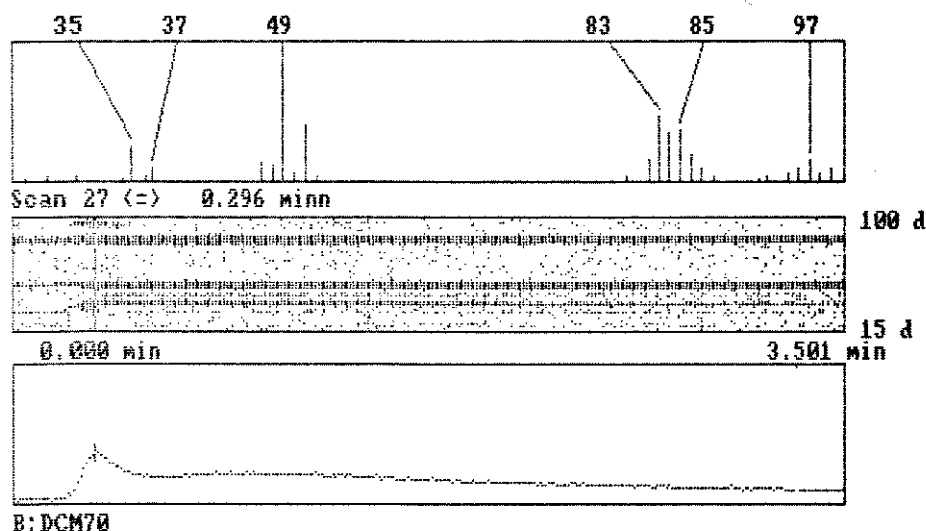
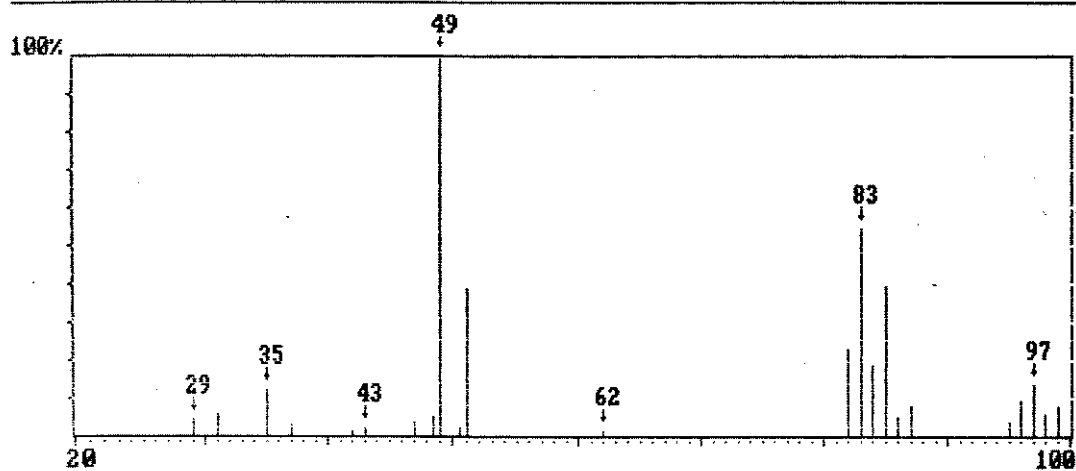


Figura 1 - Cromatograma do cloreto de metileno.

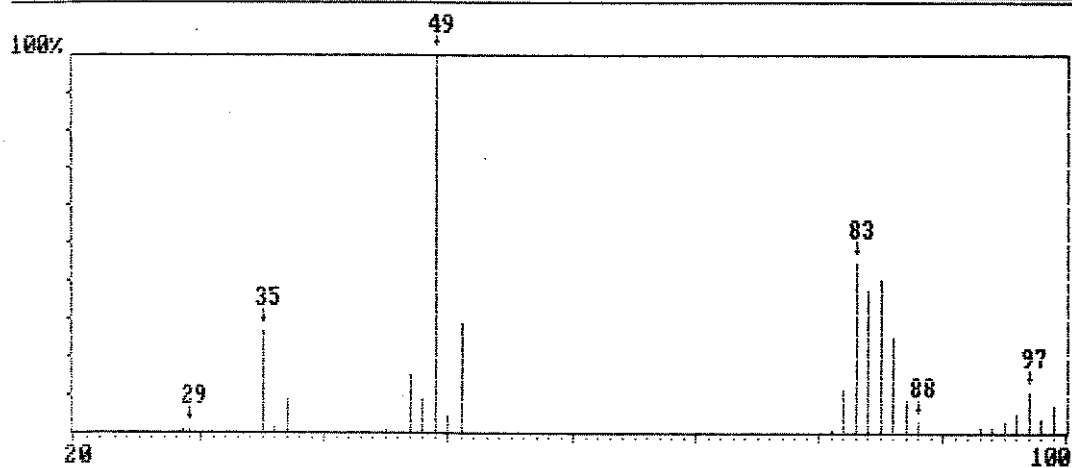
Foram selecionados 8 espectros ao longo do cromatograma, iniciando na subida do pico (scan 23) e indo até a cauda (scan 70), e submetidos ao programa FA3. Este identificou a presença de 3 componentes, sendo que seus perfis de concentração são mostrados na figura 3 e os espectros, na tabela 3.

- ANÁLISE FATORIAL -

Cromatograma: DCM70 Tempo de retencao: 0.240 min



Cromatograma: DCM70 Tempo de retencao: 0.389 min



Cromatograma: DCM70 Tempo de retencao: 0.404 min

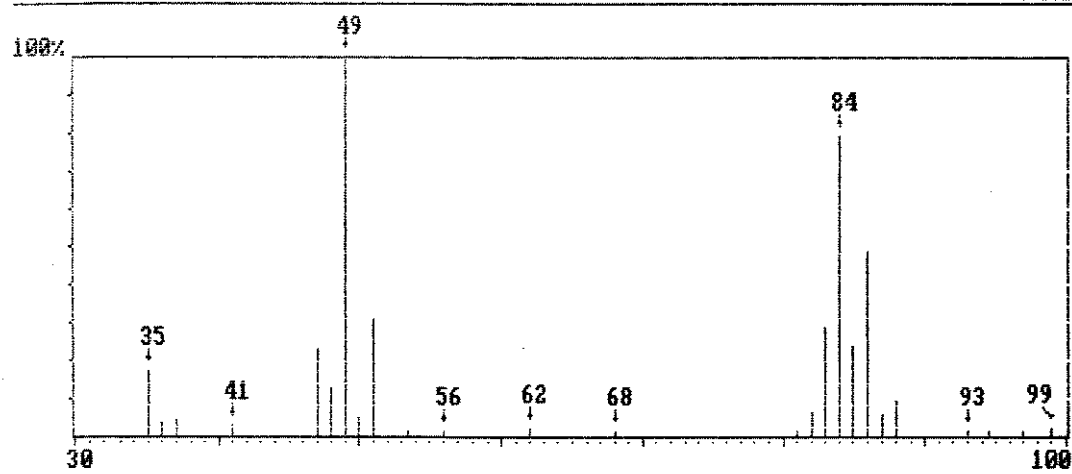


Figura 2 - Espectros 29 (0,240min), 28 (0,389min) e 95 (0,404min) do cromatograma do cloreto de metileno.

- ANÁLISE FATORIAL -

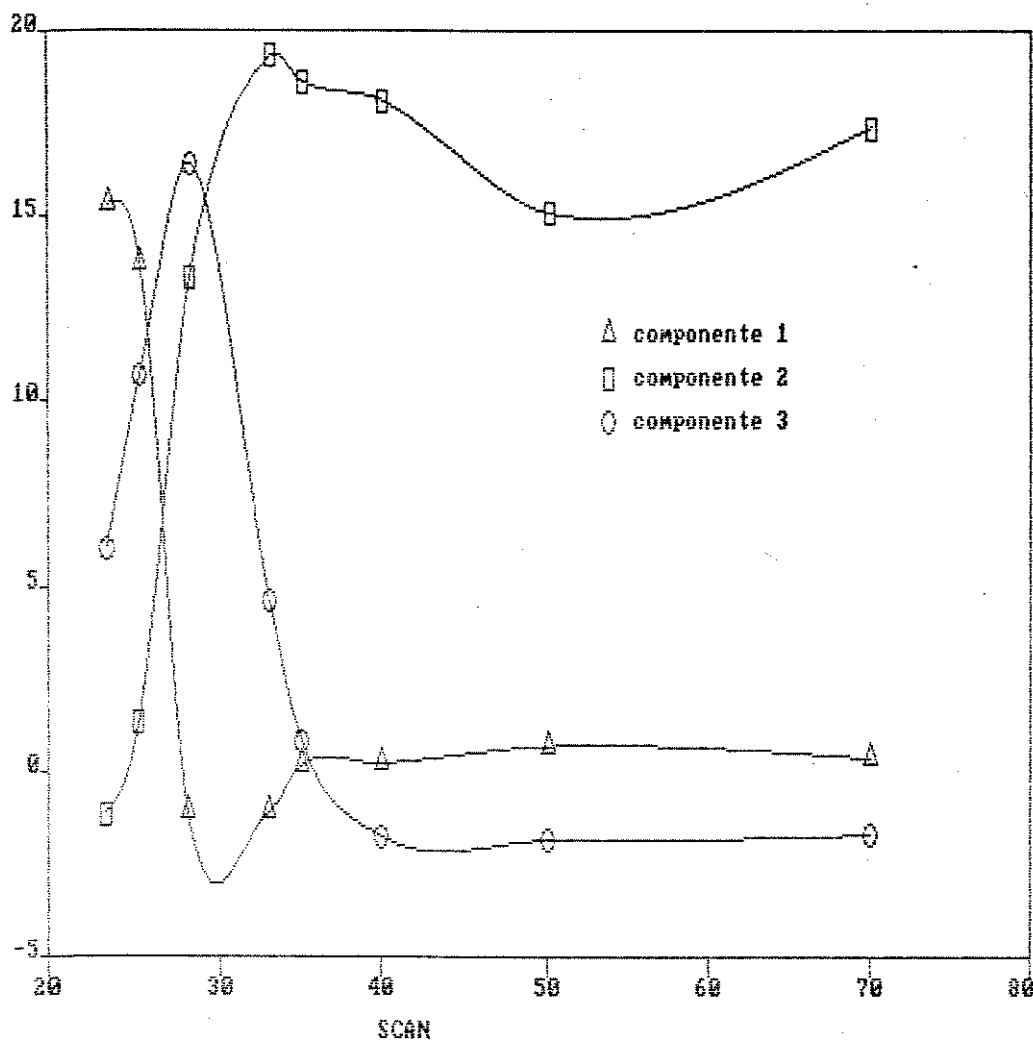


Figura 3 - Perfil de concentração dos 3 componentes identificados por análise fatorial.

Observa-se que para os três componentes o pico base é 49d, que em conjunto com 51d caracteriza o íon CH_2Cl^+ . Pelo padrão isotópico na região de 82 a 87d e pelo perfil de concentração, o componente dois é sem dúvida CH_2Cl_2 . Tanto o componente 1 quanto o 3 apresentam picos na região de 95 a 99d. O conjunto dos picos 97 e 99d indica $\text{C}_2\text{H}_3\text{Cl}_2^+$. Seria esperado um outro pico de menor intensidade em 101, sem registro pois a varredura foi feita até 100d. O conjunto dos picos 96 e 98d deve corresponder a $\text{C}_2\text{H}_2\text{Cl}_2^+$. O pico em 31, presente apenas no componente 1, deve ser referente

- ANÁLISE FATORIAL -

ao fragmento CH_2OH^+ do etanol utilizado como estabilizante.

m/z d	Componente		
	1	2	3
29	3,9	0,0	1,5
31	6,1	0,0	0,0
35	6,8	15,8	35,2
36	0,0	3,0	0,0
37	1,3	3,9	12,0
41	1,2	2,7	-2,1
47	2,9	20,1	11,7
48	5,8	11,9	6,2
49	100,0	100,0	100,0
50	1,6	4,2	4,4
51	43,3	29,5	29,9
82	23,4	4,0	17,3
83	55,8	25,7	60,8
84	32,9	79,5	3,4
85	30,6	21,2	51,7
86	10,3	52,5	-0,3
87	5,7	5,1	10,1
88	2,7	9,4	-1,4
93	-0,8	0,3	1,5
94	0,0	0,0	2,7
95	3,2	0,3	5,1
96	10,0	0,6	9,7
97	14,0	0,9	19,3
98	6,2	0,3	6,4
99	8,9	0,8	12,5

Tabela 3 - Espectros de massas dos 3 componentes puros obtidos por análise fatorial.

Este conjunto de informações sugere que a amostra de cloreto de metileno está contaminada com outros compostos clorados, que não podem ser identificados pela falta de informações sobre os picos acima de 100d. As tabelas contendo as informações detalhadas dos espectros utilizados nesta análise se encontram no Apêndice 3.

3 Estudo sobre o Potencial de Ionização

A identificação de substâncias por espectrometria de massas é feita, com raras exceções de experimentados espectrometristas, por simples comparação com padrões conhecidos. Uma grande quantidade de algoritmos de comparação foram desenvolvidos e estão sendo sempre aperfeiçoados. Os sistemas de dados comerciais incluem programas que utilizam um ou mais destes algoritmos e que apresentam, de modo geral, três problemas:

- ▶Dificuldades com a presença de contaminantes,

- ▶Necessidade de o padrão ter sido registrado sob as mesmas condições instrumentais e

- ▶Necessidade de a substância desconhecida ter seu espectro registrado no banco de dados.

Estas dificuldades estimulam o desenvolvimento de técnicas que não necessitem de padrões e enormes bancos de dados. A análise de padrões isotópicos apresentada no capítulo anterior é uma ferramenta viável na interpretação de espectros de massas. Ocorre, porém, que é muitíssimo comum a sobreposição de padrões isotópicos de fragmentos e nem sempre as técnicas de eliminação de íons contaminantes resolvem adequadamente o problema. Com a finalidade de eliminar este problema de misturas de padrões isotópicos, foi imaginada uma nova técnica utilizando o potencial de ionização.

Um espectro de massas pode ser visto na verdade como uma mistura de uma razoável quantidade de espécies iônicas, sendo que às vezes ocorre sobreposição total ou parcial dos seus padrões isotópicos. O objetivo da técnica é resolver estas misturas, fornecendo os padrões isotópicos de cada espécie separadamente. De posse destes dados, seria possível utilizar programas como o AC4

- ANÁLISE FATORIAL -

para determinar a composição elementar de um íon.

A análise fatorial permite a separação de um espectro de massas de uma mistura em seus componentes, desde que sejam fornecidos espectros de misturas com diferentes proporções destes componentes. A variação na proporção entre as quantidades dos diversos íons pode ser conseguida pela variação da energia dos elétrons na câmara de ionização.

A um determinado valor de energia dos elétrons, uma espécie, inicialmente neutra, pode se ionizar pela remoção de um elétron:



sendo o potencial necessário para que isto ocorra designado por $I(XY)$. Com o aumento da energia dos elétrons, pode-se fornecer energia suficiente para que ocorra a dissociação de uma ligação química, originando uma nova espécie, que por possuir massa diferente, será observada em um menor valor de m/z :



O potencial necessário para que isto ocorra é designado por $V(X^+)$ e está relacionado com a energia de dissociação da ligação pela equação [9]:

$$V(X^+) = D(X-Y) + I(X) + E_c + E_x \quad (11),$$

onde $D(X-Y)$ é a energia de dissociação da ligação $X-Y$, E_c é a energia cinética e E_x a energia de excitação eletrônica, vibracional e rotacional dos fragmentos. Para muitos processos é possível ter energia cinética menor que 0,5eV e as partículas no estado fundamental. Nestes casos, o valor de $V(X^+)$, também conhecido por potencial de aparecimento, é dado pela energia de dissociação da ligação $X-Y$ e pelo potencial de ionização da espécie radicalar X .

- ANÁLISE FATORIAL -

Assim, o método de separação dos íons consiste em adquirir espectros a diferentes valores de energia dos elétrons e, através da análise fatorial, determinar as intensidades dos padrões isotópicos.

Amostra: Cloreto de metileno Merck PA	
Volume injetado: 7 μ l	
Pressão: $\approx 2 \cdot 10^{-5}$ Torr	
Corrente total no ionizador: 100 μ A	
Sensibilidade do detector: 10 $^{-7}$ A/V	
Voltagem no multiplicador de elétrons: 2600V	
Energia dos elétrons: Experimento	1 - 10eV
	2 - 15eV
	3 - 20eV
	4 - 25eV
	5 - 30eV
	6 - 35eV
	7 - 40eV
	8 - 45eV
	9 - 50eV
	10 - 55eV
	11 - 60eV
Região de varredura: 12 a 86d	
Espectros acumulados por experimento: 40	
Número de conversões A/D por pico: 42	

Tabela 4 - Condições de análise do cloreto de metileno.

3.1 Aplicação do Método

Para testar o método proposto foi escolhido inicialmente o cloreto de metileno por duas razões: tem peso molecular abaixo de 100, permitindo utilizar o Finnigan 1015S/L e, por possuir dois átomos de cloro que tem o isótopo 37 com razoável intensidade,

- ANÁLISE FATORIAL -

ocorrem sobreposições dos padrões isotópicos dos fragmentos. A tabela 4 mostra as condições em que foram realizadas as aquisições dos espectros. No Apêndice 3 se encontram as tabelas completas dos espectros utilizados para este estudo. Aqui serão apresentados somente os resultados mais significativos.

►Análise 1: Para reduzir a matriz de dados, foram selecionados os valores de m/z que possuam pico com intensidade superior a 5% do mais intenso em pelo menos um dos espectros. O programa FA3 identificou 6 componentes (tabela 5), sendo que eram esperadas aproximadamente 8 espécies iônicas. Mais importante que o fato de ter identificado um número menor de componentes é a apresentação de picos negativos com intensidade suficiente para não serem considerados como ruídos.

m/z	Componente					
d	1	2	3	4	5	6
13	0,0	0,0	0,0	0,0	0,0	100,0*
14	0,0	0,0	100,0*	0,0	0,0	0,0
35	100,0*	0,0	0,0	0,0	0,0	0,0
36	0,0	0,0	0,0	0,0	32,6*	0,0
37	27,2	10,3	-11,6	-0,5	-4,4	24,1
47	59,9	14,6	44,8	-13,4	100,0	-59,9
48	25,3	-3,1	42,8	0,3	36,6	-79,8
49	29,5	-14,3	40,0	100,0	36,8	-68,1
51	0,0	0,0	0,0	32,8*	0,0	0,0
84	-9,8	100,0	-49,7	16,7	65,4	-8,4
86	0,0	38,4*	0,0	0,0	0,0	0,0

Tabela 5 - Matriz dos componentes da análise 1. Os sinais * indicam os picos de massa pura.

- ANÁLISE FATORIAL -

A solução adotada foi dividir o espectro em conjuntos de picos de m/z próximos e aplicar o programa a estes novos conjuntos de dados. Deste modo foram feitas as análises de 4 regiões: 12 a 15d, 35 a 38d, 47 a 52d e 82 a 86d.

Análise 2: Na região de 12 a 15d seria esperada a identificação dos íons C^+ , CH^+ e CH_2^+ e, embora o número de componentes esteja correto, os padrões isotópicos não são razoáveis (tabela 6). Esta análise ressalta dois importantes aspectos do método:

▶É sempre necessário que o número de picos do conjunto seja superior ao número de componentes. Este é um requisito básico do método de análise fatorial. Se forem utilizados os picos de 12 a 14d, somente dois componentes são determinados.

▶A necessidade de picos de massa pura anula os isótopos menos significativos dos outros componentes.

m/z d	Componentes		
	1	2	3
12	100,0*	0,0	0,0
13	0,0	0,0	100,0*
14	-89,8	100,0	80,1
15	0,0	4,1*	0,0

Tabela 6 - Matriz de componentes da análise 2.

Análise 3: Na região de 35 a 38d estão dois fragmentos comuns aos compostos clorados: Cl^+ e HCl^+ . Porém, a análise indica a presença de 3 componentes (tabela 7). O componente 1 é o HCl^+ e os componentes 2 e 3 correspondem aos isótopos 35 e 37 do cloro.

Análise 4: Restringindo o número de componentes da análise

- ANÁLISE FATORIAL -

anterior para dois, o resultado é bastante satisfatório (tabela 8).

m/z d	Componentes		
	1	2	3
35	0,0	100,0*	0,0
36	100,0*	0,0	0,0
37	0,0	0,0	100,0*
38	27,8	-7,8	28,4

Tabela 7 - Matriz de componentes da análise 3.

m/z d	Componentes	
	1	2
35	0,0	100,0*
36	100,0*	0,0
37	2,9	32,0
38	28,6	1,2

Tabela 8 - Matriz de componentes da análise 4.

Análise 5: Nesta região seriam esperados os íons CH_2Cl^+ , CHCl^+ e CCl^+ correspondentes a 49, 48 e 47d -tomando por base o isótopo 35 do cloro. Para a região de 47 a 51d, o programa acusou a presença de 4 componentes. Como isto não era esperado, foram incluídos os picos em 52d, não alterando o número de componentes identificados (tabela 9). O componente 3 corresponde a CH_2Cl^+ . Os componentes 1 e 4 não possuem similaridade com os outros dois íons esperados. Ignorando-se o pico negativo em 49d, o componente 2

- ANÁLISE FATORIAL -

possui uma boa correspondência com CH_3Cl^+ . Este íon não era esperado, indicando a presença de contaminantes na amostra.

m/z	Componentes			
d	1	2	3	4
47	100,0*	0,0	0,0	0,0
48	0,0	0,0	0,0	90,6*
49	4,3	-343,9	100,0	100,0
50	13,7	100,0	2,2	-14,2
51	0,0	0,0	32,1*	0,0
52	0,0	30,6*	0,0	0,0

Tabela 9 - Matriz de componentes da análise 5.

m/z	Componentes		
d	1	2	3
82	100,0*	0,0	0,0
83	0,0	0,0	8,2*
84	-14,4	100,0	100,0
85	1,3	3,0	1,7
86	0,0	50,1*	0,0

Tabela 10 - Matriz de componentes da análise 6.

Análise 6: Devido à presença de dois átomos de cloro, o íon molecular apresenta três importantes picos: 84, 86 e 88d. Porém, como foram feitas varreduras até 86d, o último pico foi excluído dos espectros. Foram determinados 3 componentes (tabela 10), sendo que o segundo corresponde ao íon molecular CH_2Cl_2^+ . O mais importante a se observar neste caso é a presença do componente 3

- ANÁLISE FATORIAL -

que é completamente estranho ao cloreto de metileno. O perfil de concentração (figura 4) mostra que este componente está presente em alta concentração e com potencial de aparecimento menor. Isto indica definitivamente a contaminação da amostra. Devido à ausência de picos em 86d, este contaminante não é clorado. Como a análise posterior por GC/MS não apresentou este tipo de contaminante, acreditamos que a contaminação tenha ocorrido na injeção da amostra.

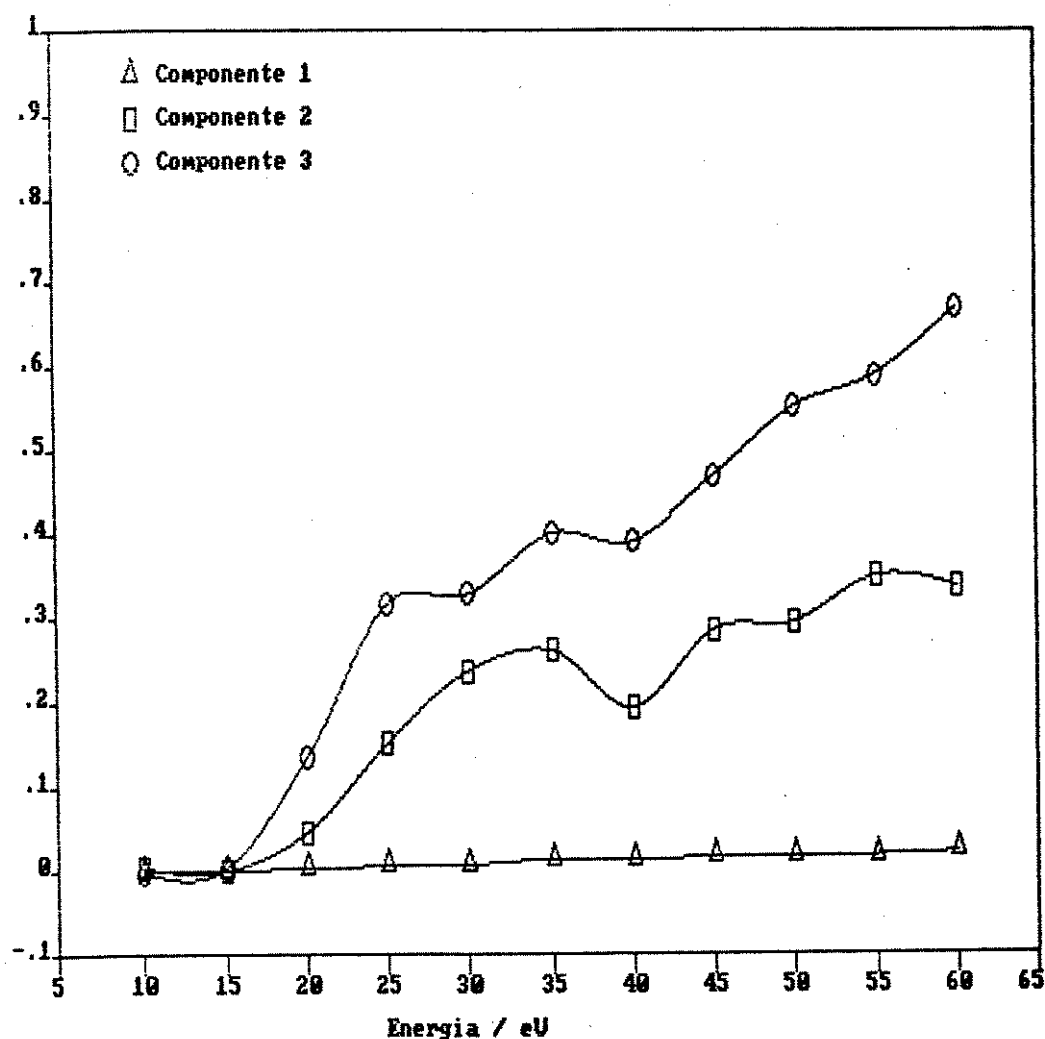


Figura 4 - Perfil de concentração dos 3 componentes identificados pela análise α .

observar o comportamento do método sob este outro tipo de situação. A tabela 11 mostra as condições de análise de benzeno. A região mais relevante para ser utilizada vai de 49 a 53d. A tabela 12 mostra o resultado da análise, que indicou a presença de 4 componentes. Contrariando o esperado, somente o componente 3 apresentou um segundo isótopo bastante abundante (52d) e novamente um pico negativo bastante significativo é apresentado no componente 1.

3.2 Discussão e Conclusões sobre o Método

Como foram obtidos resultados bastante satisfatórios com o programa FA3 quando aplicado aos dados da literatura e do cromatograma do cloreto de metileno, as dúvidas sobre o aparecimento de picos negativos recaem sobre outros pontos do processo. Estas "anomalias" poderiam ser causadas pela instabilidade do espectrômetro de massas ou mesmo sobre o *hardware* da interface. Porém, devido a insistência do fenômeno após mudanças no tipo de amostra, no *hardware* e no *software*, procuramos outra explicação para o que ocorre.

A idéia inicial baseia-se no princípio de que os diversos picos de um padrão isotópico devem sofrer as mesmas variações em função da energia dos elétrons. O que não foi considerado é que há um compromisso entre as abundâncias dos fragmentos e dos íons dos quais são formados. Por exemplo, a fragmentação do benzeno ocorre segundo o diagrama da figura 5. Quando a energia fornecida é baixa, a perda de acetileno a partir do íon molecular é favorecida em relação a perda de hidrogênio radicalar, pois envolve apenas um rearranjo. Assim, a fragmentação via perda de C_2H_2 ocorre preferencialmente. A medida que é fornecida energia suficiente para a quebra da ligação C-H, a fragmentação via $C_6H_5^+$ torna-se um caminho alternativo. Com a formação deste íon, haverá uma menor quantidade de íon molecular disponível para a formação de $C_6H_6^{+}$. Isto cria uma dependência entre os picos em 49 e 52d, de modo que o aumento de um deles causa a diminuição do outro. Este tipo de

- ANÁLISE FATORIAL -

comportamento explicaria os picos negativos.

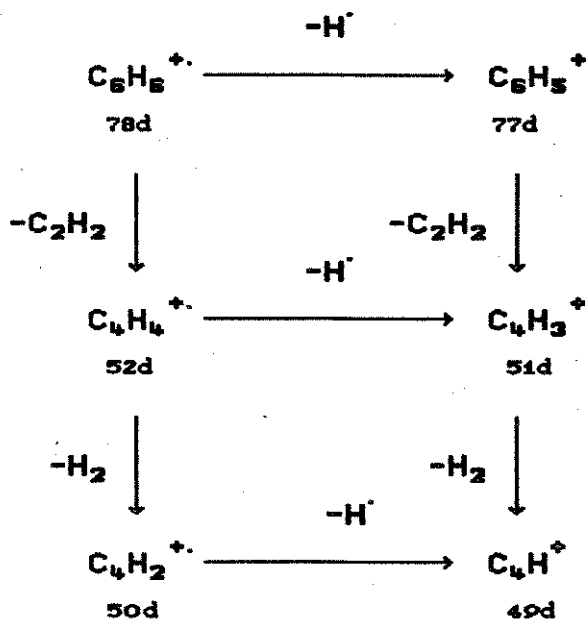


Figura 5 - Fragmentação do benzeno.

De qualquer forma, estes são apenas testes preliminares, sendo que para obter resultados mais conclusivos seria necessário um estudo mais aprofundado com um maior número de análises e com a utilização de um espectrômetro adaptado para este fim. O equipamento deveria apresentar as seguintes características:

► Possuir uma fonte monoenergética. Isto pode ser conseguido com a colocação de um setor eletrostático entre o filamento e a câmara de ionização ou utilizando um outro método (fotoionização por UV, laser, etc.). Desta forma seria possível fazer a determinação precisa dos potenciais de aparecimento, o que forneceria dados adicionais sobre o tipo de ligação rompida para a formação do fragmento.

► Possuir um sistema de controle sobre a pressão da amostra na câmara de ionização, para que não ocorram flutuações na intensidade do espectro.

► Possuir um sistema de detecção e conversão A/D com maior

- ANÁLISE FATORIAL -

faixa dinâmica, pois quando a energia dos elétrons é baixa, o espectro é pouco intenso.

►Fazer o controle sobre esta fonte via *software*, permitindo a varredura do potencial de ionização.

4 Bibliografia

- 1 - R.W. Wallace, *J. Phys. Chem.* 64, 899 (1960).
- 2 - D. Katakis, *Anal. Chem.* 37, 876 (1965).
- 3 - G.L. Ritter, S.R. Lowry and T.L. Isenhour, *Anal. Chem.* 48, 591 (1976).
- 4 - E.R. Malinowski, *Anal. Chem.* 49, 612 (1977).
- 5 - F.J. Knorr and J.H. Futrell, *Anal. Chem.* 51, 1236 (1979).
- 6 - E.R. Malinowski, *Anal. Chim. Acta* 134, 129 (1982).
- 7 - E.R. Malinowski, *Anal. Chem.* 49, 606 (1977).
- 8 - J. Ortega, "The Givens-Householder Method for Symmetric Matrices" in A. Ralston and H.S. Wilf, "Mathematical Methods for Digital Computers", John Wiley & Sons, New York, 1965, Vol. II, p. 94.
- 9 - C.A. McDowell, "Mass Spectrometry", McGraw-Hill, New York, 1963, p. 59.

CAPÍTULO VII

AVALIAÇÃO E CONCLUSÕES

A avaliação global que fazemos do trabalho é positiva, na medida que julgamos termos concluído a proposta inicial, dando nossa contribuição na área de desenvolvimento de *software* e *hardware* para espectrometria de massas e cromatografia gasosa/espectrometria de massas. Como o trabalho aborda vários temas, a seguir damos nossa avaliação final sobre o desempenho e principais contribuições além das perspectivas de estudos futuros e possíveis alterações de cada tópico do trabalho.

1 Sistema de Dados do IQ-UNICAMP

O sensor para campo magnético, apesar das suas limitações atuais, representa uma alternativa interessante, uma vez que utiliza componentes de baixo custo e razoável sensibilidade ao campo magnético.

A principal limitação deste sistema é a faixa de m/z coberta pelos dois espectrômetros (até 100d). Para satisfazer as condições de rotina de análise do IQ-UNICAMP, um equipamento deve atingir valores superiores a 500 daltons. Embora o limite seja praticamente o mesmo para os dois espectrômetros, as causas são distintas.

No caso do Finnigan 1015S/L, o conversor D/A de 16bits é suficiente para se obter bons resultados até 1000d, como é o caso do 4021 que utiliza o mesmo conversor. Porém, duas causas podem

- AVALIAÇÃO E CONCLUSÕES -

estar contribuindo para o baixo desempenho: o circuito de DC/RF e o alinhamento do quadrupolo.

Para o MAT311A o processo é completamente diferente, onde o sensor de campo magnético envia o sinal para um conversor A/D de 12bits. Neste caso, a origem do problema pode ser o sensor magnético que não possui estabilidade adequada ou o sistema de conversão A/D com elevado nível de ruído. As possibilidades de futuras alterações são:

► Utilizar um *Sample & Hold* na entrada do conversor A/D, a fim de minimizar o nível de ruído.

► Alterar o modo de medir ΔV_p , dispensando o conversor A/D. Um modo seria injetar uma rampa de potencial nos emissores dos transistores, medindo a diferença de tempo entre o disparo de ambos. O valor da contagem de tempo seria tomado como função do campo magnético.

A dificuldade de calibração deste sensor está relacionado a fixação do mesmo junto ao eletroímã, pois qualquer movimento altera consideravelmente sua resposta.

2 Sistema de Dados do CCEN-UFPa

Embora exista uma grande variedade de tipos de conversores A/D comerciais mais rápidos que o modelo proposto, este conta com dois aspectos favoráveis: o baixo custo e a facilidade de reposição dos componentes utilizados. O modo de trabalho deste conversor, que é original, pode ser melhorado, eliminando a principal falha encontrada. Quando o valor da tensão de entrada é muito próximo do valor da tensão de referência, ocorre oscilação na saída do comparador. Uma solução possível é fazer com que a saída do comparador atue somente em um intervalo de tempo pequeno após a tensão em sua entrada estabilizar. Após este intervalo, a saída do comparador deve permanecer estática, independente das oscilações em sua entrada.

O conjunto dos programas para calibração, aquisição e tratamento de dados deste sistema tem desempenho bastante

- AVALIAÇÃO E CONCLUSÕES -

satisfatório. O principal recurso a ser adicionado é uma subrotina eficiente para localização e integração dos picos cromatográficos.

3 Programas AC4 e FA3

A principal contribuição do programa AC4 à técnica de análise de padrões isotópicos é o parâmetro de comparação, que fornece valores mais confiáveis e representantes da realidade.

Estes programas diferenciam os sistemas desenvolvidos dos equivalentes comerciais. Eles se caracterizam como ferramentas a serem utilizadas na interpretação dos espectros e não simplesmente na comparação com outros de amostras conhecidas.

Quanto a técnica sugerida para o desmembramento do espectro nos diversos padrões isotópicos que o compõe, concluímos que está ainda no princípio, necessitando de um número maior de testes. Para dar continuidade a este trabalho seria necessário um equipamento mais específico, como sugerido no final do capítulo VI.

APÊNDICE 1

RELAÇÃO DOS COMPONENTES DOS DIAGRAMAS ELETRÔNICOS

Nas páginas seguintes são apresentadas as relações de componentes dos diagramas dos circuitos eletrônicos desenvolvidos neste trabalho. A relação é identificada pelo Número do Documento constante no diagrama.

Salvo menção em contrário, são feitas as seguintes considerações:

► Os resistores são de 1% de precisão e 1/8 ou 1/4W.

► A tensão de isolamento dos capacitores é superior a 50V.

► O prefixo antes da numeração do componente deve ser entendido como:

A ou U para circuito integrado,

C para capacitor,

D para diodo,

J ou P para conector,

Q para transistor,

R para resistor,

SW para chave e

TR para transformador.

► O valor de capacitância anotado após a identificação de um circuito integrado, se refere a um capacitor que foi colocado em paralelo com a alimentação do mesmo.

► O símbolo * no local da descrição de um componente indica que este foi removido do circuito.

- APÊNDICE 1 -

F1015F - CLL

#	Descrição
C1	2200 μ F 25V eletrolítico
C2	100nF cerâmico
C3	10 μ F 25V tântalo
C4	100nF cerâmico
C5	2200 μ F 40V eletrolítico
C6	100nF cerâmico
C7	2200 μ F 40V eletrolítico
C8	100nF cerâmico
D1 e D2	1N4005
D3	LED 10mA
D4 a D7	1N4005
D8 e D9	LED 10mA
R1	330 Ω 5%
R2 e R3	1,2k Ω 5%
TR1	110+110V / 6+6V 1A
TR2	110+110V / 15+15V 1A
U1	7805
U2	7815
U3	7915

F1015P1 - CLL

#	Descrição
C1	1 μ F 16V tântalo
P1	DB25
U1	74LS244 (47nF)
U2	*
U3	74LS30 (47nF)
U4	74LS00 (47nF)

- APÊNDICE 1 -

U5	74LS04 (47nF)
U6	74LS245 (47nF)

F1015P2A - CLL

#	Descrição
C1	100nF cerâmico
C2	2,2nF cerâmico
R1	22k Ω
U7	*
U8	74LS374 (47nF)
U9	74LS00 (47nF)
U10 e U11	74LS138 (47nF)
U12	LM555

F1015P2B - CLL

#	Descrição
U1	74LS374 (47nF)
U2 e U3	74LS244 (47nF)
U4 e U5	74LS374 (47nF)

F1015P3 - CLL

#	Descrição
A1 e A2	LM308
C1 e C2	100pF cerâmico
C3 e C4	1 μ F 16V tântalo

- APÊNDICE 1 -

C5 e C6	100nF cerâmico
R1 e R1a	200k Ω
R2 e R2a	200k Ω
R3b	200k Ω
R3c	100k Ω
R3d	40,2k Ω
R3e	20,0k Ω
R3f	10,0k Ω
R3g	5,10k Ω
R3h	2,49k Ω
R4 a R9	1k Ω 5%
U1 a U3	CD4051 (47nF)

SENSMAT - CLL

#	Descrição
C1 e C2	22nF poliéster
C3 e C4	10 μ F tântalo
C5	220nF poliéster
C6	1nF poliéster
C7 e C8	100nF cerâmico
Q1 e Q2	2N2646
R1	10,0k Ω
R2 e R3	500 Ω
R4	10,0k Ω
R5 e R6	100 Ω
R7	5,00k Ω
R8	100k Ω
R9	200k Ω
R10	5,00k Ω
R11	20k Ω
R12	5k Ω trimpot multivoltas

- APÊNDICE 1 -

R13	71,5k Ω
R14	20,0k Ω
R15 e R16	200k Ω trimpot multivoltas
R17	40,0k Ω
U1	LM324

F4021F - GLL

#	Descrição
C1	100nF 400V poliéster
C2 e C3	2200 μ F 40V eletrolítico
C4 e C5	100nF cerâmico
C6	4,7 μ F 16V tântalo
C7 e C8	100nF cerâmico
D1 a D4	1N4005
D5 a D8	LED 10mA
R1 e R2	1k Ω 5% 1/2W
R3 e R4	1k Ω 5%
R5	330 Ω 5%
R6	1k Ω 5%
R7	330 Ω 5%
SW1	2 pólos X 2 posições
TR1	110+110V / 18+18V 2A
U1	7815
U2	7805
U3	7915
U4	7905

- APÊNDICE 1 -

F4021P1 - CLL

#	Descrição
C1	1 μ F 16V tântalo
P1	DB25
U1	74LS244 (100nF)
U2	74LS04 (100nF)
U3	74LS00 (100nF)
U4	74LS30 (100nF)
U5	74LS138 (100nF)
U6	74LS245 (100nF)

F4021P2A - CLL

#	Descrição
P2	DB25
U1	74LS138 (100nF)
U2	74LS374 (100nF)
(U5c)	ver F4021P2B - CLL

F4021P2B - CLL

#	Descrição
C1	1nF poliéster
C2 e C3	1 μ F 16V tântalo
R1	1k Ω 5%
U3 e U4	74LS244 (100nF)
U5	74LS04 (100nF)
U6	74LS00 (100nF)
U7	74LS132 (100nF)

- APÊNDICE 1 -

U8 e U9 74LS193 (100nF)
 U10 e 11 74LS164 (100nF)

F4021P2C - CLL

#	Descrição
A1	CA741 (100nF)
A2 a A6	LF356N (100nF)
A7	LM319 (100nF)
A8	μ A723 (100nF)
A9 e A10	CD4053 (100nF)
C1 e C2	4,7nF styroflex
C3	*
C4	4,7 μ F 25V tântalo
C5	1nF cerâmico
C6	100pF cerâmico
D1	Zener 4,7V
J1 e J2	Conector blindado para circuito impresso
R1	30,0k Ω
R2 e R3	20,0k Ω
R4	102k Ω
R5	95,3k Ω
R6	10k Ω trimpot multivoltas
R7	*
R8 e R9	20k Ω trimpot multivoltas
R10	8,45k Ω
R11	7,15k Ω
R12	2k Ω trimpot multivoltas
R13	20k Ω trimpot multivoltas
R14	8,45k Ω
R15	7,15k Ω
R16	2k Ω trimpot multivoltas

- APÊNDICE 1 -

R17	560 Ω 5%
R18	10,0 Ω
R19	4,99k Ω
R20	3,00k Ω
R21	10k Ω trimpot multivoltas
R22	7,15k Ω
R23 e R24	*
R25	1,00k Ω
R26	1,02k Ω
R27	10,0k Ω

F4021GC - CLL

#	Descrição
C1	10 μ F 16V tântalo
P3	DB25
R1 a R7	2,7k Ω 5%
U1	74LS245 (100nF)
U2	74LS138 (100nF)
U3	74LS244 (100nF)

APÊNDICE 2

LISTAGEM DOS PROGRAMAS

Arquivo	Tipo	Número de páginas
GRAPH.CLL	Arquivo de Inclusão	2
AUXILIO.CLL	Arquivo de Inclusão	8
MONITOR.PAS	Programa	4
INTERFAC.CLL	Arquivo de Inclusão	3
CAL100.PAS	Programa	14
AJUSTE.PAS	Programa	7
COL100.PAS	Programa	6
MD2.PAS	Programa	12
MATCOL.PAS	Programa	6
MATTRATN.PAS	Programa	17
CAL4021.PAS	Programa	7
GCMS.PAS	Programa	19
TRATA.PAS	Programa	19
GRAF.PAS	Programa	12
AC4.PAS	Programa	26
FA3.PAS	Programa	11

```

Begin
  y := (linha - 1) * 8;
  For x := 0 to 639 do buf[x] := AgulhaComposta(x,y);
  Write(1st, ' '); { 10 espaços }
  Write(1st, ESC, DENSIDADE, #128, #2);
  For x := 0 to 639 do Write(1st, buf[x]);
  WriteIn(1st, ' ');
End;

Begin {Print}
  GraphWindow(0,639,199);
  Write(1st, ESC, 'A', #8); { Acerta o entrelinhamento. }
  linha := 0;
  Repeat
    linha := linha + 1;
  Until (linha = 25) or keypressed;
  WriteIn(1st, ESC, 'Z'); { Retorna o entrelinhamento. }
End;

type TipoJanela = Record
  xi, xf, yi, yf : Integer;
end;

TipoConjJanela = Array[1..10] of TipoJanela;

var janelaagr : TipoConjJanela;

Procedure DefineJanela(xi,y1,x2,y2,j : Integer);
Begin
  With janelaagr[j] do
    begin
      xi := x1;
      yi := y1;
      xf := x2;
      yf := y2;
      Quadro(xi - 1, yi - 1, xf + 1, yf + 1, 1);
    end;
  End;

Procedure SetJanela(j : Integer);
Begin
  With janelaagr[j] do GraphWindow(xi,y1,xf,yf);
End;

```

```

procedure Graphics;
external 'GRAPH.BIN';
external Graphics(0);
external Graphics(3);
external Graphics(6);
external Graphics(9);
external Graphics(12);
external Graphics(13);
external Graphics(18);
external Graphics(21);
external Graphics(24);
external Graphics(27);
external Graphics(30);
external Graphics(33);
external Graphics(36);
external Graphics(39);
external Graphics(42);
external Graphics(45);
external Graphics(48);
external Graphics(51);
external Graphics(54);
external Graphics(58);

Procedure Quadroi(xi,y1,x2,y2,cor : Integer);
Begin
  Draw (x1,y1,x2,y1,cor);
  Draw (x1,y1,x1,y2,cor);
  Draw (x2,y1,x2,y2,cor);
  Draw (x1,y2,x2,y2,cor);
End;

Procedure Print;
type TipoAgulha = Array[0..7,0..1] of byte;
const Peso : TipoAgulha = ((0,128), (0,64), (0,32), (0,16), (0,8),
(0,4), (0,2), (0,1));
DENSIDADE = 'L';
ESC = #27;
var linha : Integer;
buf : Array[0..639] of Byte;
bufs : Array[0..639] of Char absolute buf;

Function AgulhaComposta(x,y : Integer) : Integer;
var ag, i, cor : Integer;
Begin
  ag := 0;
  For i := 0 to 7 do
    begin
      cor := GetDotColor(x,y + i);
      If cor < 0 then cor := 0;
      ag := ag or Peso[i,cor];
    end;
  AgulhaComposta := ag;
End;

Procedure MandarLinha;
var y, x : Integer;

```

```

{
  {
    {
      Auxilio.CLL
      Versao 1.1
    }
    {
      CLL 90
      (Participacao VFJ)
    }
  }
}

Procedure Tab(x : Integer);
Begin
  GoToXY(x, WhereY);
End;

{$V-}
Type CadeiaDe25 = String(25);
CadeiaDe255 = String(255);
CadeiaDe14 = String(14);

Procedure WriteRes(s : CadeiaDe255);
{ ( Enfatiza as letras maiusculas. ) }
var i : Integer;
Begin
  For i := 1 to length(s) do
    begin
      If s[i] in ['A'..'Z'] then TextColor(15) else TextColor(7);
      Write(s[i]);
    end;
  TextColor(7);
  Writeln;
End;

Procedure WriteV(s : CadeiaDe25);
{ Escreve na vertical. }
var i, j, n, ultima : Byte;
v : Byte absolute s;
Begin
  i := WhereX;
  j := WhereY;
  n := 1;
  If i = 98 then ultima := 24 else ultima := 25;
  While (j <= ultima) and (n <= v) do
    begin
      GoToXY(i,j); Write(s[n]);
      n := n + 1;
      j := j + 1;
    end;
End;

Function ReadKbd : Char;
var c : Char;
Begin
  ReadKbd := UpCase(c);
End;

Function Confirma(mensagem : CadeiaDe255) : Boolean;
var tecla : Char;
Begin
  Write(mensagem); WriteRes(' (S/N)');
  Repeat
    ReadKbd; tecla;
  Until tecla in ['S', 'N'];
  If tecla = 'S' then Confirma := TRUE else Confirma := FALSE;
End;

Procedure Aviso(tipo : CadeiaDe14);
{-----}
{
  {
    {
      Aviso sonoro para programacao
    }
    {
      { Carcter inicial
        Aviso
      }
      {
        { c ou C
          Cuidado!
        }
        { i ou I
          Invalido
        }
        { p ou P
          Pronto
        }
        { e ou E
          Erro
        }
      }
    }
  }
}
{-----}

var i : Integer;
t : Char;

Procedure som(f,d : Integer);
Begin
  Sound(f);
  delay(d);
End;

Begin (Aviso)
  t := UpCase(tipo[1]);
  Case t of
    'C' : For i := 1 to 3 do
      begin
        Som(1000,10);
        Som(1500,100);
      end;
    'E' : For i := 1 to 4 do
      begin
        Som(150,100);
        Som(1000,50);
      end;
    'P' : For i := 1 to 3 do
      begin
        Som(200*(1 shl i),100);
      end;
    'I' : begin
      Som(3000,10); Som(5000,20); Som(1000,50);
    end;
  end;
End;
```

```

    Intr($IA,reg);
End;

Procedure RestauraClock;
var reg : RegList;
Begin
    reg.AX := $0000; { Ler o contador. }
    Intr($IA,reg);
    reg.DX := contador1 + reg.DX; { Palavra aenos significativa. }
    reg.CX := contador2 + reg.CX; { Palavra aais significativa. }
    reg.AX := $0100; { Alterar o contador. }
    Intr($IA,reg);
    contador1 := 0;
    contador2 := 0;
End;

Function Tempo : Real; { 0 tempo e' dado em minutos. }
const FATOR = 9.154249E-04; { minutos / contagea }
var reg : RegList;
    td, tc : Real;
Begin
    reg.AX := $0000; { Ler o contador. }
    Intr($IA,reg);
    If reg.DX >= 0 then td := reg.DX else td := reg.DX + 65536.0;
    If reg.CX >= 0 then tc := reg.CX else tc := reg.CX + 65536.0;
    Tempo := FATOR * (td + 65536.0 * tc);
End;

Procedure Relogio(var hora, minuto, segundo, centesiao : Byte);
var reg : RegList;
Begin
    reg.AX := $2000;
    MSDOS(reg);
    hora := hi(reg.CX);
    minuto := lo(reg.CX);
    segundo := hi(reg.DX);
    centesiao := lo(reg.DX);
End;

type par = Record
    car : Char;
    atributo : Byte;
end;

TipoLinha = Array[1..80] of par;
TipoTela = Array[1..25] of TipoLinha;
String20 = String[20];
String80 = String[80];

Procedure Leia(var campo : String80; n : Integer);
const CR = #13;
    ESC = #27;
    BS = #8;
    HOME = #71;
    FIM = #79;
    DIREITA = #77;

```

```

    end;
    'A' : For i := 1 to 5 do
        begin
            Som(4000, 10);
            Som(8000, 10);
            Som(2000, 10);
        end;
    else Som(2500, 200);
    end;
    NoSound;
End;

Procedure Aguarda(tempo : Integer);
{ tempo em milissegundos. }
var i : Integer;
Begin
    i := 0;
    Repeat
        delay(10); { Cada etapa e' de 10 ms. }
        i := i + 10;
    Until keypressed or (i = tempo);
End;

type RegList = Record
    AX, BS, CX, DX, BP, SI, DI, DS, ES, Flags : Integer;
end;

Function Date:CadeiaDe14;
Var Reg : RegList;
    ano, mes, dia : String(2);
Begin
    Reg.AX := $2000;
    MSDOS(Reg);
    Reg.CX := Reg.CX - 1900;
    If Reg.CX > 99 then Reg.CX := Reg.CX - 100;
    Str(Reg.CX, ano); Str(Hi(Reg.DX), mes); Str(Lo(Reg.DX), dia);
    If Length(ano) = 1 then ano := '0' + ano;
    If Length(mes) = 1 then mes := '0' + mes;
    If Length(dia) = 1 then dia := '0' + dia;
    Date := dia + '/' + mes + '/' + ano;
End;

const contador1 : Integer = 0;
    contador2 : Integer = 0;

Procedure ResetClock;
var reg : RegList;
Begin
    reg.AX := $0000; { Ler o contador. }
    Intr($IA,reg);
    contador1 := contador1 + reg.DX; { Palavra aenos significativa. }
    contador2 := contador2 + reg.CX; { Palavra aais significativa. }
    reg.AX := $0100; { Alterar o contador. }
    reg.DX := 0;
    reg.CX := 0;

```

```

ESQUERDA = #75;
DEL = #83;
INS = #82;
var tecla : Char;
x, y, i, cursor : Integer;
Tela : Tipotela absolute $800:$800;
Begin
  n := n - 1;
  y := WhereY;
  x := WhereX;
  cursor := x;
  While Tela[y,cursor].car = '' do cursor := cursor + 1;
  Repeat
    If cursor < x then cursor := x;
    If cursor > x + n then cursor := x + n;
    Tab(cursor);
    Read(kbd,tecla);
    If (tecla = ESC) and keypressed then Read(kbd,tecla);
  Case tecla of
    '0'..'9', '.', '-', ' ', 'E', 'C' : begin
      Tela[y,cursor].car := tecla;
      cursor := cursor + 1;
    end;
    HOME : cursor := x;
    FIM : cursor := x + n;
    DIREITA : cursor := cursor + 1;
    ESQUERDA : cursor := cursor - 1;
    BS : begin
      If cursor > x then cursor := cursor - 1;
      For i := cursor to x + n - 1 do
        Tela[y,i].car := Tela[y,i + 1].car;
        Tela[y,x + n].car := '';
      end;
    end;
    DEL : begin
      For i := cursor to x + n - 1 do
        Tela[y,i].car := Tela[y,i + 1].car;
        Tela[y,x + n].car := '';
      end;
    end;
    INS : begin
      For i := x + n downto cursor + 1 do
        Tela[y,i].car := Tela[y,i - 1].car;
        Tela[y,cursor].car := '';
      end;
    end;
    CR :;
    else Aviso('invalido');
  end;
  Until tecla = CR;
End;

Function Convis(i : Integer) : String20;
{ Converte Integer para String. }
var s : String20;
Begin
  Str(i,s);

```

```

Convis := s;
End;

Function Convis(r : Real) : String20;
{ Converte Real para String. }
var s : String20;
Begin
  Str(r,s);
  Convis := s;
End;

Procedure Telivar atual : Integer; min,max : Integer;
{ ----- }
{ Le um valor inteiro. }
{ ----- }
{ atual - valor default }
{ min - limite inferior }
{ max - limite superior }
{ ----- }
var i, x, y,
  novo, cod : Integer;
  corc, corb : Byte;
  bom : Boolean;
  campo : String(6);
  Tela : Tipotela absolute $800:$800;
  linha : Tipolinha;

Procedure Advverte(s : String80);
Begin
  GoToXY(1,24);
  linha := Tela[24];
  Aviso('erro');
  Write(s);
  Aguarda(2000);
  Tela[24] := linha;
End;

Begin {Tel}
  x := WhereX; y := WhereY; { Guarda a posicao do cursor. }
  i := Tela[y,x].atributo;
  corc := 1 and 15; { Salva as cores atuais. }
  corb := i shr 4;
  TextBackground(corc shr 1);
  Write(actual :6);
  Repeat
    GoToXY(x,y);
    bom := TRUE;
    Leia(campo,6);
    i := 0;
    For cod := 0 to 5 do
      If Tela[y,x + cod].car <> '' then
        begin
          i := i + 1;
          campo[i] := Tela[y,x + cod].car;

```

```

end;
campo[0] := Char(i);
Val(campo,novo,cod);
If length(campo) = 0 then novo := atual;
If cod <> 0 then
begin
  Tela[y,x + cod - 1].atributo := Tela[y,x + cod - 1].atributo or 128;
  Adverte('Caracter invalido!');
  bom := FALSE;
end;
end;
If novo < min then
begin
  Adverte('Deve ser maior que ' + ConvRS(min));
  bom := FALSE;
end;
If novo > max then
begin
  Adverte('Deve ser menor que ' + ConvRS(max));
  bom := FALSE;
end;
end;
Until bom;
atual := novo;
TextColor(corc);
TextBackground(corb);
GoToXY(x,y); Write(novo:16);
End;

```

```

Procedure LeR(var atual : Real; min,max : Real; n,d : Integer);
{
  -----
}
{
  Le um valor real.
}
{
}
{
  n - numero de caracteres do campo
}
{
  d - numero de casas decimais
}
{
  atual - valor default
}
{
  min - limite inferior
}
{
  max - limite superior
}
{
  -----
}
var i, x, y, cod : Integer;
    corc, corb : Byte;
    novo : Real;
    bom : Boolean;
    campo : String[20];
    Tela : TipTela absolute $800:$800;
    linha : Tipolinha;

```

```

Procedure Adverte(s : String80);
Begin
  GoToXY(1,24);
  linha := Tela[24];
  Aviso('erro');
  Write(s);
  Aguarda(300);
  Tela[24] := linha;
End;

```

```

Begin (LeR)
  x := WhereY; y := WhereY; { Guarda a posicao do cursor. }
  GoToXY(x,y);
  i := Tela[y,x].atributo;
  corc := i and 15; { Salva as cores atuais. }
  corb := i shr 4;
  TextColor(corc);
  TextBackground(corb shr 1);
  Write(atual:16);
  Repeat
    GoToXY(x,y);
    bom := TRUE;
    Leia(campo,n);
    i := 0;
    For cod := 0 to n - 1 do
      If Tela[y,x + cod].car <> '' then
        begin
          i := i + 1;
          campo[i] := Tela[y,x + cod].car;
        end;
    campo[0] := Char(i);
    Val(campo,novo,cod);
    If length(campo) = 0 then novo := atual;
    If cod <> 0 then
      begin
        Tela[y,x + cod - 1].atributo := Tela[y,x + cod - 1].atributo or 128;
        Adverte('Caracter invalido!');
        bom := FALSE;
      end;
    If novo < min then
      begin
        Adverte('Deve ser maior que ' + ConvRS(min));
        bom := FALSE;
      end;
    If novo > max then
      begin
        Adverte('Deve ser menor que ' + ConvRS(max));
        bom := FALSE;
      end;
    end;
    Until bom;
    atual := novo;
    TextColor(corc);
    TextBackground(corb);
    GoToXY(x,y); Write(novo:16);
  End;
  ($V*)

```

```

var i : Integer;
Begin
  HiRes; ClearScreen;
  Quadro(328,4,631,126,1);
  For i := 0 to 2 do
  begin
    Draw(330 + i * 150, 127, 330 + i * 150, 130, 1);
    Draw(330 + i * 150, 0, 330 + i * 150, 3, 1);
    Draw(328, 5 + i * 60, 328, 5 + i * 60, 1);
    Draw(332, 5 + i * 60, 339, 5 + i * 60, 1);
  end;
  For i := 0 to 9 do
  begin
    Plot(330 + i * 30, 127, 1);
    Plot(330 + i * 30, 3, 1);
    Plot(327, 5 + i * 12, 1);
    Plot(328, 5 + i * 12, 1);
    Plot(332, 5 + i * 12, 1);
    Plot(333, 5 + i * 12, 1);
  end;
  GraphWindow(330,5,630,125);
  GoToXY(1,20); Write('F10 finaliza');
End;

Procedure Inicio;
Begin
  canal := 0; ganho := 0;
  Seleto(canal, ganho);
  GoToXY(1,1); Write('Canal : ', canal);
  GoToXY(1,2); Write('Ganho : ', ganho);
  xi := 1; xf := 300;
  fatorm := 300.0 / (xf - xi);
  yi := 0; yf := 4095;
  fatory := 120.0 / (yf - yi);
  GoToXY(1,3); Write('Dominio : ', xi :4, ' a ', xf :4);
  GoToXY(1,4); Write('Imagem : ', yi :4, ' a ', yf :4);
  a := 0; ( Vetoer mostrado. )
End;

Procedure Comando;
var pilha : Integer;
Begin
  Read(kbd,optcao);
  If (optcao = ESC) and keypressed then
  begin
    Read(kbd, optcao);
  end
  else
  begin
    optcao := UpCase(optcao);
    Case optcao of
      'C' : begin
        canal := (canal + 1) mod 8;
        Seleto(canal, ganho);
        GoToXY(10,1); Write(canal);
      end;
    end;
  end;
End;

```

```

Program Monitor;
const F1 = #59;
      F2 = #60;
      F3 = #61;
      F4 = #62;
      F5 = #63;
      F6 = #64;
      F7 = #65;
      F8 = #66;
      F9 = #67;
      F10 = #68;
      ESC = #27;
      MAXDAT = 2000;

type Tipovetor = Array[1..MAXDAT] of Integer;
      Tipomatriz = Array[0..1] of Tipovetor;

var optcao : Char;
    canal,
    ganho,
    a,
    xi, xf,
    yi, yf : Integer;
    fatorm, fatory : Real;
    a : Tipomatriz;

{$I GRAPH.CLL}

Procedure Cicleo;
var x, y, i : Integer;
Begin
  y := WhereY;
  x := WhereX;
  For i := x to 79 do
  begin
    GoToXY(i,y);
    Write(' ');
  end;
  GoToXY(x,y);
End;

Procedure Seleto(var canal, ganho : Integer);
Begin
  Port[3307] := canal OR (ganho shl 4);
End;

Function ADC : Integer;
Begin
  Port[3304] := 0;
  While (Port[3306] > 31) do; ( Aguarda a conversao )
  ADC := Port[3305];
  End;

Procedure Tela;

```

```

begin
  Seletor(c, g);
  c := ct - c;
  g := gt - g;
  a[m,i] := ADC;
end;
End;

Procedure Mostra(m : Integer);
var i, m0, xa : Integer;

Function x(sinal : Integer) : Integer;
Begin
  x := Round(fatorx * (sinal - xi));
End;

Function y(sinal : Integer) : Integer;
Begin
  y := Round(fatory * (yf - sinal));
End;

Begin
  m0 := 1 - m;
  For i := xi to xf do
    begin
      xa := x(i);
      Plot(xa, a[m0,i], 0);
      a[m,i] := y(a[m,i]);
      Plot(xa, a[0,i], 1);
    end;
  End;

  BEGIN
    Tela;
    Inicio;
    Repeat
      opcao := ESC;
    Until keypressed then
      Goando
    else
      begin
        Coleta(m);
        Mostra(m);
        m := 1 - m; { Muda o Vetur. }
      end;
    Until opcao = F10;
  END.

```

```

end;
'G' : begin
  ganho := (ganho + 1) mod 8;
  Seletor(canal, ganho);
  GoToXY(18,2); Write(ganho);
end;
'g' : begin
  GoToXY(1,25); ClrEol;
  Write('X inicial? '); Read(xi);
  If xi < 1 then xi := 1;
  If xi > MAXDAT then xi := MAXDAT;
  GoToXY(1,25); ClrEol;
  Write('X final? '); Read(xf);
  If xf < 1 then xf := 1;
  If xf > MAXDAT then xf := MAXDAT;
  If xi > xf then
    begin
      pilha := xi;
      xi := xf;
      xf := pilha;
    end;
  If xf = xi then
    if xf = MAXDAT then xi := xf - 1 else xf := xi + 1;
  GoToXY(11,3); Write(xi * 4);
  GoToXY(18,3); Write(xf * 4);
  GoToXY(1,25); ClrEol;
  fatorx := 300.0 / (xf - xi); { Utilizado ea Mostra. }
  ClearScreen;
end;
'I' : begin
  GoToXY(1,25); ClrEol;
  Write('Y inicial? '); Read(yi);
  GoToXY(1,25); ClrEol;
  Write('Y final? '); Read(yf);
  If yf = yi then yf := yi + 1;
  GoToXY(11,4); Write(yi * 4);
  GoToXY(18,4); Write(yf * 4);
  GoToXY(1,25); ClrEol;
  fatory := 120.0 / (yf - yi); { Utilizado ea Mostra. }
  ClearScreen;
end;
end;
{Case}
opcao := ESC;
end; {else}
End;

Procedure Coleta(m : Integer);
var i : Integer;
c, g, ct, gt : Integer;
Begin
  Seletor(canal, ganho);
  c := canal; g := ganho; { Monocanal. }
  ct := canal + c;
  gt := ganho + g;
  For i := xi to xf do

```

{ Subrotinas para controle e aquisicao de dados do Finnigan 4021. }

Procedure SetDA(v : Integer);

Begin

PortW[\$318] := v;

End;

var potenconv,

numconvad : Integer;

Procedure SetNumConv(n : Integer);

Begin

potenconv := n;

numconvad := 1 shl n;

End;

Function ADC : Integer;

var ad, i : Integer;

Begin

ad := 0;

For i := 1 to numconvad do

Begin

Port[\$31A] := 0;

While (PortW[\$31B] < 0) do;

ad := ad + PortW[\$31B];

end;

ADC := ad shr potenconv;

End;

Procedure SetIn(temp : Integer);

Begin

Port[\$310] := temp shr 1;

End;

Procedure SetSep(temp : Integer);

Begin

Port[\$312] := temp shr 1; { Separador ou detector B }

End;

var da6C : Integer;

solenoido : Byte;

Procedure SetCol(temp : Real);

Begin

da6C := 4095 and not(Round(8.0 * (temp + 100.0)));

Port[\$307] := lo(da6C);

Port[\$30A] := hi(da6C) or solenoido;

End;

Procedure AtivaVento(vento : Boolean);

Begin

If vento then

solenoido := solenoido or 16 else { Ativa bit 4 }

solenoido := solenoido and 239; { Desativa bit 4 }

Port[\$30A] := hi(da6C) or solenoido;

end;

Procedure AtivaSubAmb(ativa : Boolean);

Begin

If ativa then

solenoido := solenoido or 32 else { Ativa bit 5 }

solenoido := solenoido and 223; { Desativa bit 5 }

Port[\$30A] := hi(da6C) or solenoido;

end;

var flowA, auxB : Byte;

Procedure SetFluxo(flowA : Integer);

Begin

Port[\$30C] := flowA or auxB;

End;

Procedure AtivaSplit(valvula, ativa : Char);

{ 'I' para injetor e 'S' para separador }

Begin

Case valvula of

'I' : Begin

If ativa = 'A' then

solenoido := solenoido or 128 else { Ativa bit 7 }

solenoido := solenoido and 127; { Desativa bit 7 }

Port[\$30A] := hi(da6C) or solenoido;

end;

'S' : Begin

If ativa = 'A' then auxB := 1 else auxB := 0;

Port[\$30C] := flowA or auxB;

end;

else Aviso('Invalido');

end;

end;

Procedure Status;

var xpos, ypos : Integer;

status1,

status2, b : Byte;

flowA,

columna,

col40,

injetor,

separador : String[6];

Begin

status1 := Port[\$301];

status2 := Port[\$302];

b := status1 and 3; { Examina os bits 0 e 1. }

Case b of

0 : flowA := 'falha' ;

1 : flowA := 'acima' ;

2 : flowA := 'abaixo' ;

3 : flowA := 'pronto' ;

end;

b := (status1 shr 4) and 3; { Examina os bits 4 e 5. }

```

Program Cal;
{-----}
{ Este programa faz a calibracao da rampa do Finnigan 1015 S/L }
{-----}
(
  (
    CLL 90
  )
)
{-----}

const CanalFinnigan = 2;
      NUMAD      = 42;
      ESC        = #27;
      ESQ        = #75;
      DIR        = #77;
      TESQ       = #115;
      TDIR       = #116;
      CIMA       = #72;
      BAIXO      = #88;
      F1         = #59;
      F2         = #68;
      F5         = #63;
      F9         = #67;
      F10        = #68;

type Tipo_Sinal = Array[1..1000] of Integer;
     Tipo_Pico  = Array[1..50] of Integer;
     String14   = String[14];

var raapa      : Tipo_Sinal absolute $2400:0000;
     sinal      : Tipo_Sinal absolute $3400:0000;
     arqpar     : Text;
     opcao      : Char;
     tecla      : Char;
     maxraap,
     minraap,
     nro, np,
     n, id, i, j,
     mfinal,
     picobase,
     linhabase  : Integer;
     ganho      : Byte;
     soma,
     d2vdaZ,
     dvda,
     vB,
     passo      : Real;
     valorda,
     pico       : Tipo_Pico;
     javarrew,
     javarreulinha: Boolean;

($I GRAPH.CLL)
($I AVISO.CLL)
($I INPRIME.FSL)
($R+)

```

```

Procedure SetMass(m : Integer);
Begin
  PortW($300) := rampa[m];
End;

```

```

Procedure SeletorCanal, ganho : Byte);
Begin
  Port($307) := canal + ganho * 16;
End;

```

```

Function ADC : Integer;
var ad : Integer;
Begin

```

```

  Port($304) := 0;
  While (Port($306) > 31) do; { Aguarda a conversao }
    ad := Port($305);
    { Le o conversor }
  If ad = 4095 then Write(#7);
  ADC := ad;
End;

```

```

Procedure TelaInicial;
Begin

```

```

  TextMode(C40);
  ClrScr;
  TextColor(15);
  GoToXY(10,1);Write('Programa: CAL100A');
  GoToXY(30,20);Write('CLL 98');
  TextColor(7);
  GoToXY(1,3);
  Write('Faz a calibracao do Finnigan 1015 S/L');
  TextColor(31);
  GoToXY(10,15);Write('Aguarde ...');
End;

```

```

Procedure Recuper;
Begin

```

```

  Assign(arqpar, 'PAR100.FIN');
  Reset(arqpar);
  Readln(arqpar, d2vda2);
  Readln(arqpar, dvda);
  Readln(arqpar, v0);
  Close(arqpar);
End;

```

```

Procedure CalcRampa;
var m : Integer;

```

```

  minr,
  maxr : Real;
Begin
  maxr := ((-1.0 - v0) / dvda); { Limita o maximo valor enviado ao DAC. }
  If maxr > 10000.0 then maxraap := Round(maxr);
  minr := ((-32767.0 - v0) / dvda); { Limita o minimo valor enviado ao DAC. }
  If minr < 300.0 then minraap := 300 else minraap := Round(minr);
  For m := minraap to maxraap do

```

```

  rampa[m] := Round((d2vda2 * m) * m2 + dvda * m2 + v0);
  Aviso('pronto');
End;

```

```

Procedure Menu;
Begin

```

```

  TextMode(C80); TextColor(15); ClrScr;
  WriteLn('C - Altera as condicoes de varredura');
  WriteLn('I - Restaura as condicoes padroes');
  WriteLn('L - Determina a linha base');
  WriteLn('A - Faz a aquisicao de um espectro');
  WriteLn('B - Grafica o espectro');
  WriteLn('F - Finaliza o programa');
End;

```

```

Procedure CondicoesPadroes;
Begin

```

```

  ClrScr; TextColor(15);
  m20 := minraap div 100;
  m2final := maxraap div 100;
  passo := 0.05;
  linhaBase := 0;
  ganho := 2;
  WriteLn('Condicoes Padroes');
  WriteLn; delay(200);
  WriteLn('massa inicial = ', m20, ' uma'); delay(200);
  WriteLn('massa final = ', m2final, ' uma'); delay(200);
  WriteLn('incremento de m/z = ', passo, '42, uma/dado'); delay(200);
  WriteLn('ganho de entrada = ', ganho); delay(200);
  i := 0;
  Repeat
    delay(100);
    i := i + 1;
  Until (KeyPressed OR (i = 100));
  Javarreulinha := FALSE;
End;

```

```

Procedure Condicoes;
Begin

```

```

  ClrScr;
  WriteLn('CONDICOES');WriteLn;
  WriteLn('massa inicial? (atual = ', m20, ' uma) ');
  ReadLn(m20);
  WriteLn('massa final? (atual = ', m2final, ' uma) ');
  ReadLn(m2final);
  WriteLn('Incremento de m/z? (atual = ', passo, '42, e minimo = 0.01 uma) ');
  ReadLn(passo);
  WriteLn('Ganho de entrada? (de 0 a 7) (atual = ', ganho, ') ');
  ReadLn(ganho);
  Aviso('pronto');
  Javarreulinha := FALSE;
End;

```

```

Procedure DetLinhaBase;
var i, m;

```

```

If UpCase(tecla) < 'S' then Exit;
WriteLn;
WriteLn('Verifique se o equipamento esta pronto e tecla. ');
WriteLn;
Aviso('cuidado');
ai := az0 + 100;
af := azfinal + 100;
p := Round(passo * 100.0);
az := ai;
id := Round(az0 / passo);
i := id;
SetMass(ai);
aquece := ADC;
Seleto(CanalFinnigan, ganho);
Read(kbd, tecla);
TextColor(31);
WriteLn('Fazendo a varredura...');
WriteLn;
TextColor(7);
Aviso('apito');
Repeat
  SetMass(az);
  soma := 0.0;
  For j := 1 to NUMAQ do soma := soma + ADC;
  Sinal[i] := Round(soma / NUMAQ) - Linhabase;
  az := az + p;
  i := i + 1;
Until(az > af);
n := i - 1;
Linhabase := 0;
Aviso('pronto');
WriteLn('Normalizando o espectro ...');
picobase := Sinal[id];
For i := id to n do
  If Sinal[i] > picobase then picobase := Sinal[i];
  If picobase = 0 then picobase := 1;
fator := 1000.0 / picobase;
For i := id to n do Sinal[i] := Round(Sinal[i] * fator);
javarreu := True;
Aviso('pronto');
End;

```

```

Procedure Graficar(az0, afinal : Integer);
var f, g : Real;
    az0, azfinal,
    i, xp,
    cursant,
    xant,
    yant,
    fim,
    inicio,
    cursor,
    inccursor,
    eax : Integer;
    picosenc : Boolean;

```

```

p, ai,
af : Integer;
Begin
  ClrScr;
  GoToXY(21,2); TextColor(15);
  WriteLn('DETERMINACAO DA LINHA BASE DO ESPECTRO');
  TextColor(7); WriteLn;
  WriteLn('Ganho de entrada = ,ganho);
  WriteLn;
  WriteLn('Confirma? (S/N)');
  Read(kbd, tecla);
  If UpCase(tecla) < 'S' then Exit;
  WriteLn; TextColor(15);
  WriteLn('A corrente de ionizacao deve ser ZERO. ');
  WriteLn('Verifique se o equipamento esta pronto e tecla. ');
  WriteLn;
  Aviso('cuidado');
  ai := az0 + 100;
  af := azfinal + 100;
  SetMass(ai);
  Seleto(CanalFinnigan, ganho);
  soma := ADC;
  Read(kbd, tecla);
  TextColor(31);
  WriteLn('Determinando a linha base...');
  WriteLn;
  TextColor(7);
  Aviso('apito');
  soma := 0.0;
  For i := 1 to NUMAQ do soma := soma + ADC;
  Aviso('pronto');
  Linhabase := Round(soma / NUMAQ);
  javarreulinha := True;
End;

Procedure Scan;
var i, az,
    p, ai,
    af,
    aquece : Integer;
    fator : Real;
Begin
  ClrScr;
  GoToXY(28,2); TextColor(15);
  WriteLn('AQUISICAO DE UM ESPECTRO');
  TextColor(7); WriteLn;
  WriteLn('Condições de Varredura:');
  WriteLn; WriteLn('Massa inicial = ,az0, ' uma');
  WriteLn('Massa final = ,azfinal, ' uma');
  WriteLn('Incremento = ,passo*42, ' uma/dado');
  WriteLn('Ganho de entrada = ,ganho);
  WriteLn('Tempo previsto para a aquisicao = ');
  WriteLn(Round((afinal - az0)/passo/1000.0), ' segundos');
  WriteLn;
  WriteLn('Confirma? (S/N)');
  Read(kbd, tecla);

```

```

If m20 > mfinal then
  begin
    pilha := mfinal;
    mfinal := m20;
    m20 := pilha;
  end;
If m20 = mfinal then
  If m20 > m0 then m20 := m0 else mfinal := mfinal;
  inicio := Round(m20 / passo);
  fim := Round(mfinal / passo) - 1;
  f := 500.0 / (fim - inicio);
End;

Function x(mz : Integer) : Integer;
Begin
  x := Round(f * (mz - inicio));
End;

Function y(sinal : Integer) : Integer;
Begin
  y := Round(g * (max - sinal));
End;

Procedure PosicionaScan;
Begin
  GraphWindow (10,12,510,62);
  ClearScreen;
  For i := inicio to fim do
    Plot(x(i),y(sinal(i)),1);
  If picosenc then
    For i := 1 to np do
      begin
        xp := Round( Pico[i] / (100.0 * passo));
        If ((xp > inicio) AND (xp < fim)) then
          Draw(x(xp), y(0), x(xp), y(sinal(xp)), 1);
        end;
      end;
End;

Procedure PosicionaBarras;
Begin
  If not picosenc then Exit;
  GoToXY (2,12);
  Writeln ('Picos Localizados = ',np : 4);
  GraphWindow (10,100,510,150);
  ClearScreen;
  For i := 1 to np do
    begin
      xp := Round( Pico[i] / (100.0 * passo));
      If ((xp > inicio) AND (xp < fim)) then
        Draw (x(xp), y(0), x(xp), y(sinal(xp)), 1);
      end;
    end;
End;

Procedure PosicionaCursor;
Begin

```

```

Procedure MenuGr;
Begin
  GoToXY (66,4); Window(67,1,80,25);
  GraphWindow(523,1,638,198);ClearScreen;
  HiResColor(15);
  GoToXY (4,2);
  Writeln ('MENU GR');
  Writeln;
  Writeln('L linha base');
  Writeln('E Elimina');
  Writeln('pico');
  Writeln('C Cria pico');
  Writeln('R Calibra');
  Writeln(Chr(24), ' Manual');
  Writeln('A Atualiza');
  Writeln('tela');
  Writeln('M Menu');
  Writeln('inicial');
  Writeln('F2 novo fim');
  Writeln('F1 * inicio');
  Writeln('F5 restaura');
  Writeln('os limites');
  Writeln('F9 e F10 ');
  Writeln('audam passo');
  Writeln('Move cursor');
  Writeln(Chr(27), ' Chr(26));
  Writeln('Ctrl ' Chr(27), ' Chr(26));
  Window(1,1,88,25);
End;

Procedure TelaGrfica;
Begin
  HiRes; HiResColor (15);
  Window(1,1,80,25);
  GraphWindow(0,0,639,199);
  ClearScreen;
  GoToXY (2,1);
  Writeln ('Espectro Original      intensidade maxia = ',picobase);
  GoToXY (1,22); Writeln ('Regiao: ',m20:3, ' a ', mfinal:4, ' uaa');
  GoToXY (23,22); Writeln (' cursor = ', uaa);
  GoToXY (1,23); Writeln ('linha base = ');
  Quadro (9,11,511,63,1);
  Quadro (9,99,511,151,1);
  Quadro (522,0,639,199,1);
  MenuGr;
End;

Procedure CalcFatores;
var pilha : Integer;
Begin
  If max = 0 then max := 1;
  If passo = 0 then passo := 1;
  g := 50.0 / max;

```

```

GraphWindow (10,12,510,199);
If cursor > fim then cursor := fim;
If cursor < inicio then cursor := inicio;
xp := x(cursor);
Draw (xant, 53, xant, 63, 0);
Draw (xant,141, xant,151, 0);
Plot(xant,y(sinal[cursant]),1);
Draw (xp, 53, xp, 63, 1);
Draw (xp,141, xp,151, 1);
Plot(xp,y(sinal[cursant]),0);
cursant := cursor;
xant := xp;
GoToXY(38,22);
WriteLn(cursor * passo:1612);
End;

Procedure Posicionalinha;
var yp : Integer;
Begin
  GraphWindow(0,12,528,62);
  If linhabase < 0 then linhabase := 0;
  If linhabase > max then linhabase := max;
  yp := y(linhabase);
  Draw (3,yant,8,yant,0);
  Draw (512,yant,517,yant,0);
  Draw (3, yp, 8, yp, 1);
  Draw (512, yp, 517, yp, 1);
  yant := yp;
  GoToXY (14, 23);
  Write(linhabase:4);
End;

Procedure Atualiza;
Begin
  CalcFatores;
  Window(1,1,80,25);
  GoToXY (1,22); WriteLn ('Regiao: ',az03,' a ',azfinal:4,' uua');
  GoToXY (23,22); WriteLn (' cursor = ', uua);
  GoToXY (1,23); WriteLn ('linha base = ');
  PosicionaScan;
  PosicionaBarras;
  PosicionaCursor;
  PosicionaLinha;
End;

Procedure EliminaPico;
var me, j;
faixa : Integer;
Begin
  faixa := Round(passot200);
  me := Round(cursor * passo * 100.0 - faixa/2.0);
  i := 1;
  While ((pico[i] < me) AND (i < np)) do i := i + 1;
  If ((pico[i] >= me) AND (pico[i] <= me + faixa)) then
    begin
      np := np + 1;
      pico[np] := Round((soma/soma2)*passot100);
      valordat[np] := rampa(pico[np]);
    end;
  end;
  If (j - i) > 8 then (Pelo menos 8 sinais acima da linha base.)
  begin
    np := np + 1;
    pico[np] := Round((soma/soma2)*passot100);
    valordat[np] := rampa(pico[np]);
  end;
end;

For j:= i + 1 to np do pico[j - 1] := pico[j];
np := np - 1;
end
else Aviso('Invalido');
End;

Procedure CriaPico;
var me, i, j : Integer;
Begin
  If np < 50 then
    begin
      np := np + 1;
      pico[np] := 10000;
      me := Round(cursor * passo * 100.0);
      i := 1;
      While pico[i] < me do i := i + 1;
      For j := np - 1 downto i do pico[j + 1] := pico[j];
      pico[i] := me;
    end
  else Aviso('Invalido');
  End;

Procedure EncPicos;
var i, j : Integer;
soma2,
fator : Real;
Begin
  For i := id to n do sinal[i] := sinal[i];
  picobase := sinal[id];
  For i := id to n do
    begin
      If sinal[i] < 0 then sinal[i] := 0;
      If sinal[i] > picobase then picobase := sinal[i];
    end;
    fator := 1000.0 / picobase;
    For i := id to n do sinal[i] := Round(sinal[i] * fator);
    i := id; np := 0;
    While ((i < n) AND (np < 50))do
      begin
        soma := 0.0; soma2 := 0.0;
        If (sinal[i] > 0) then
          begin
            j := i;
            While ((sinal[j] > linhabase) AND (j < n))do
              begin
                soma := soma + (1.0 * j) * sinal[j];
                soma2 := soma2 + sinal[j];
                j := j + 1;
              end;
            If (j - i) > 8 then
              begin
                np := np + 1;
                pico[np] := Round((soma/soma2)*passot100);
                valordat[np] := rampa(pico[np]);
              end;
            end;
          end;
        i := i + 1;
      end;
    end;
  end;

```

```

i := j;
end;
i := i + 1;
end;
picoenc := True;
Aviso('pronto');
End;

```

```

Procedure Linha;
var tec : Char;
    incr : Integer;

```

```

Procedure MenuLn;
Begin
  GoToXY (66,4); Window(67,1,88,25);
  GraphWindow(523,1,638,198); ClearScreen;
  HiResColor(7);
  GoToXY (4,2);
  Writeln ('MENU');
  Writeln ('DA LINHA BASE');
  Writeln ('');
  Writeln ('');
  Writeln ('');
  Writeln ('P Encontra');
  Writeln ('picos');
  Writeln ('');
  Writeln ('N Volta ao');
  Writeln ('menu SR');
  Writeln ('');
  Writeln ('');
  Writeln ('F9 e F10 ');
  Writeln ('audam passo');
  Writeln ('');
  Writeln ('Move cursor');
  Writeln ('Chr(24),', Chr(25));
  Writeln ('');
  Window(1,1,88,25);
End;

```

```

Begin
  MenuLn;
  incr := 8;
  Repeat
    Read(kbd, tec); tec := UpCase(tec);
  Case tec of
    'p' : begin
      EncPicos;
      tec := 'M';
      tecia := 'A';
      end;
    'I' : PrtScr;
    ESC : begin
      Read(kbd, tec);
      Case tec of
        CIMA : linhabase := linhabase + incr;

```

```

BAIXO : linhabase := linhabase - incr;
F9 : If incr > 1 then incr := incr div 2
      else Aviso('invalido');
F10 : If incr < 20 then incr := incr * 2
      else Aviso('invalido');
else Aviso('invalido');
end;

```

```

Posicionatinha;

```

```

end;
else if tec < 'M' then Aviso('invalido');
end;

```

```

Until ( tec = 'M' );
MenuGr;
End;

```

```

Procedure CalcParametros;
var a, b : Real;
    tecia : Char;

```

```

Procedure RegLin;
var n, l : Integer;

```

```

    y, sy, sx,
    sy, sx2,
    ya, xa,
    x : Real;

```

```

Begin
  n := np;
  sy := 0; sx := 0; sy2 := 0; sx2 := 0; (Linha as somatorias)
  For i := 1 to np do
    begin
      y := valordat[i];
      x := Round(pico[i]/100.0) * 100; ( Arredonda para m/z inteira. )
      sy := sy + y;
      sx := sx + x;
      sy := sy + x;
      sx2 := sx2 + x*x;
    end;

```

```

    xa := sx/n;
    ya := sy/n;
    a := (sy - sx*sy/n)/(sx2 - sx*sx/n); ( inclinacao )
    b := ya - a*xa; ( intercepto )
  End;

```

```

Procedure AtuaPar;
Begin
  Assign(arqpar, 'PAR.FIN');
  Rewrite(arqpar);
  dnda := a;
  v0 := b;
  Writeln(arqpar, dnda);
  Writeln(arqpar, v0);
  Close(arqpar);
  tecia := 'M';
End;

```

```

pico[np] := a + 100;
WriteIn(1st,pico[np] :10, valorda[np] :10);
end;
MenuGr;
End;

Procedure Movimentos;
var tecla : Char;
Begin
  Read(kbd, tecla);
  Case tecla of
    ESC : cursor := cursor - incursor;
    DIR : cursor := cursor + incursor;
    TESO : cursor := inicio;
    TDIR : cursor := fim;
    CIMA : Manual;
    F5 : begin
      az0 := a0;
      a2final := a2final;
    end;
    F10 : If incursor < 10000 then incursor
      else Aviso('invalido');
    F9 : If incursor > 1 then incursor := 1
      else Aviso('invalido');
    F1 : az0 := Round(cursor $ passo);
    F2 : a2final := Round(cursor $ passo);
      else Aviso('invalido');
    end;
    PosicionaCursor;
  End;

Begin
  az0 := a0;
  a2final := a2final;
  np := 0;
  picosenc := false;
  TelaGrfica;
  max := 1000;
  CalcFatores;
  PosicionaScan;
  incursor := 16;
  cursor := inicio;
  cursant := cursor;
  PosicionaCursor;
  PosicionaLinhas;
  Aviso('pronto');
  Repeat
    Read(kbd,tecla); tecla := UpCase(tecla);
  Case tecla of
    ESC : Movimentos;
    'I' : PtsScr;
    'L' : Linhas;
    'E' : EliminaPico;
    'C' : CriaPico;
    'R' : CalcParametros;
  End;
End;

Begin
  HiResColor(14);
  RegIni;
  Aviso('pronto');
  GoToXY (67,4); Window(67,1,80,25);
  GraphWindow(523,1,630,198);ClearScreen;
  WriteIn('RESULTADOS');
  WriteIn;
  WriteIn('inclinacao');
  WriteIn(a : 8 : 3);
  WriteIn('interceptor');
  WriteIn(b : 8 : 1);
  WriteIn;
  WriteIn('Valores');
  WriteIn('anteriores');
  WriteIn(dvdm : 8 : 3);
  WriteIn(v0 : 8 : 1);
  WriteIn;
  WriteIn('A - Atualiza');
  WriteIn('parametros');
  WriteIn('M - Menu GR');
  Repeat
    Read(kbd, tecla); tecla := UpCase(tecla);
    If tecla = 'A' then Atualiza;
    If tecla = 'I' then PtsScr;
  Until (tecla = 'N');
  HiResColor(15);
  MenuGr;
  End;

Procedure Manual;
var m : Integer;
c : Char;
Begin
  GoToXY (66,4); Window(67,1,80,25);
  GraphWindow(523,1,630,198);ClearScreen;
  HiResColor(13);
  GoToXY (4,2);
  WriteIn('CALIBRACAO');
  WriteIn('MANUAL');
  WriteIn('');
  WriteIn('p - entra');
  WriteIn('novo pico');
  WriteIn('A - anula');
  WriteIn('os picos');
  Read(kbd,c);
  c := UpCase(c);
  If c = 'A' then np := 0;
  If c = 'p' then
    begin
      WriteIn;
      WriteIn('m/z = ');
      Readln(m);
      np := np + 1;
    end;
  valorda[np] := round(Round(cursor $ passo + 100.0));

```

```

Program Ajusta_Curva_Polinomial;
{
  {
    { Este programa ajusta um polinomio aos dados }
    {                                     CLL 88 }
  }
}

const MAXI = 20;
      ESC = #27;

type Vektor = Array[1..1000] of Real;
     tipolinha = Array[0..MAXI] of Real;
     tipomatriz = Array[0..MAXI] of tipolinha;

var s : tipomatriz;
    a : tipolinha;
    linha, coluna,
    inreg, fiareg,
    n, i, g, grau : Integer;
    x, y          : Vektor;
    fator         : Real;
    opcao,
    tecla        : Char;
    numero       : String[10];
    codigo       : Integer;
    esordea,
    jaentrou,
    jatale       : Boolean;

(R*)

($! GRAPH.CLL)

Procedure Sonatorio;
var inicio, fim, px : Real;
    k : Integer;

Begin
  For linha := 0 to grau do
  begin
    alinha := 0.0;
    For coluna := 0 to grau do s[linha, coluna] := 0.0;
  end;
  Repeat
    inicio := x(1); fim := x(n);
    WriteLn('Qual o dominio? (na mesma unidade de x)');
    WriteLn(inicio: '); ReadLn(inicio);
    WriteLn(fim: '); ReadLn(fim);
    k := 0;
    inreg := 1;
    While(x[inreg] < inicio) do inreg := inreg + 1;
    fiareg := n;
    While(x[fiareg] > fim) do fiareg := fiareg - 1;
  until opcao = 'F';
  ClrScr;
  WriteLn('Confirma? (S/N) ');
  Read(kbd,tecla);
  Until UpCase(tecla) = 'S';
END.

```

```

Begin
  pilha := a1;
  a1 := a2;
  a2 := pilha;
End;

Begin
  Repeat
    emordem := TRUE;
    For i := 1 to n - 1 do
      begin
        If x[i] > x[i + 1] then
          begin
            Trocar(x[i], x[i + 1]);
            Trocar(y[i], y[i + 1]);
            emordem := FALSE;
          end;
        end;
      until emordem;
    End;

    Procedure Entradas;
    var nome : String(14);
        arq : Text;
    Begin
      fator := 1.0; { para conversao de escalas }
      jalc := FALSE; { true apos calculo dos coeficientes do polinomio }
      jaentrou := FALSE; { true apos a entrada de dados }
      emordem := FALSE; { true apos o ordenamento da abscissa }
      WriteLn('Os dados estao em disquete? (S/N)');
      Repeat
        Read(kbd, tecla); tecla := UpCase(tecla);
      until (tecla in ['S', 'N']);
      If tecla = 'S' then
        begin
          WriteLn('Qual o nome do arquivo? (incluir o drive e complemento)');
          Readln(nome);
          If nome = '' then Exit;
          Assign(arq, nome);
          Reset(arq);
          n := 0;
          While not Eof(arq) do
            begin
              n := n + 1;
              Readln(arq, x[n], y[n]);
            end;
          Close(arq);
          jaentrou := TRUE;
        end
        else
          begin
            WriteLn('Tecla <ESC> para terminar. ');
            WriteLn;
            n := 0;
            Repeat

```

```

k := fiareg - inreg + 1;
WriteLn('Inicio: ', x[inreg], '   Fim: ', x[fiareg]);
If k < 3 then WriteLn('Dados insuficientes. ');
until k >= 3;
WriteLn('Fazendo as Somatorias... ');
s[0,0] := 1.0 * k;
For i := inreg to fiareg do
  begin
    a[0] := a[0] + y[i];
    px := 1.0;
    For g := 1 to grau do
      begin
        px := px * x[i];
        a[g] := a[g] + px * y[i];
        For linha := 0 to g do
          s[linha,g-linha] := s[linha,g-linha] + px;
        end;
      end;
    For g := 1 to grau do
      begin
        px := px * x[i];
        coluna := grau;
        For linha := g to grau do
          begin
            s[linha,coluna] := s[linha,coluna] + px;
            coluna := coluna - 1;
          end;
        end;
      end;
    End;

  Procedure Reset;
  var i, coluna,
      linha : Integer;
      fator : Real;
  Begin
    For linha := 0 to grau do
      begin
        fator := s[linha,linha];
        For coluna := 0 to grau do s[linha,coluna] := s[linha,coluna] / fator;
        a[linha] := a[linha] / fator;
        For i := 0 to grau do
          If i < linha then
            begin
              fator := s[i,linha];
              For coluna := 0 to grau do
                s[i,coluna] := s[i,coluna] - fator * s[linha,coluna];
                a[i] := a[i] - fator * a[linha];
              end;
            end;
          end;
        End;

      Procedure Ordena;

      Procedure Trocar(var a1, a2 : Real);
      var pilha : Real;

```

```

        Writeln('Dado ',n + 1);
        Write('x = ');
        Read(kbd, tecla);
        If tecla <> ESC then
            begin
                n := n + 1;
                Write(tecla);
                Readln(numero);
                Val(tecla + numero, x[n], codigo);
                Write('y = '); Readln(y[n]);
                Writeln;
            end;
        Until tecla = ESC;
        jaentrou := TRUE;
    end;
    If jaentrou then Ordena;
End;

Procedure ListaDados;
Begin
    Writeln('Deseja listar na impressora (1) ou no video (V)?');
    Repeat
        Read(kbd, tecla); tecla := UpCase(tecla);
        Until tecla in ['1', 'V'];
        If tecla = '1' then
            begin
                Writeln(1st, 'Numero de dados: ',n);
                For i := 1 to n do Writeln(1st, 'x = ',x[i], ' y = ',y[i]);
                Writeln(1st);
            end
        else
            begin
                Writeln('Numero de dados: ',n);
                For i := 1 to n do Writeln('x = ',x[i], ' y = ',y[i]);
                Writeln;
            end;
        Repeat Until KeyPressed;
    end;
End;

Procedure SalvaDados;
var nome : String[14];
    arq : Text;
Begin
    Writeln('Qual o nome do arquivo? (Incluir o drive e complemento)');
    Readln(nome);
    If nome = '' then Exit;
    Assign(arq, nome);
    Rewrite(arq);
    For i := 1 to n do Writeln(arq, x[i], y[i]);
    Close(arq);
End;

Procedure CalcPar;
Begin
    grau := 1;

```

```

        Writeln('Qual o grau do polinomio?');
        Readln(grau);
        If grau > 10 then grau := 10;
        If grau < 1 then grau := 1;
        For i := 1 to n do
            begin
                x[i] := x[i] / fator;
                y[i] := y[i] / fator;
            end;
        fator := 1.0;
        Writeln('Qual o fator de multiplicacao?');
        Readln(fator);
        For i := 1 to n do
            begin
                x[i] := x[i] * fator;
                y[i] := y[i] * fator;
            end;
        Somatorio;
        Reseq;
        For i := 0 to grau do Writeln('coef. ',i, ' --> ',a[i]);
        Repeat Until KeyPressed;
        Jacalc := TRUE;
    End;

Procedure Interpola;
var px, xi, yi : Real;
Begin
    Repeat
        Writeln;
        Write('x = ');
        Read(kbd, tecla);
        If tecla <> ESC then
            begin
                Write(tecla);
                Readln(numero);
                Val(tecla + numero, xi, codigo);
                px := 1.0;
                yi := 0.0;
                For i := 0 to grau do
                    begin
                        yi := yi + a[i] * px;
                        px := px * xi;
                    end;
                Writeln('y = ',yi);
            end;
        Until tecla = ESC;
    End;

Procedure Grafica;
const jx1 = 10;
        jxf = 612;
        jyi = 5;
        jyf = 197;
var fx, fy,

```

```

WriteLn('S - Salva dados');
WriteLn('P - Calcula os parametros');
WriteLn('I - Interpola');
WriteLn('G - Grafica');
WriteLn('ESC - Finaliza');
WriteLn;
End;

BEGIN
  EntraDados;
  Repeat
    Menu;
    Read(xb0, opcao); opcao := UpCase(opcao);
    Case opcao of
      'E' : EntraDados;
      'L' : If jaEntrou then ListaDados;
      'S' : If jaEntrou then SalvaDados;
      'P' : If emOrdem then CalcPar;
      'I' : If jaCalc then Interpola;
      'G' : If jaCalc then Grafica;
      else Write('#7');
    end;
  Until (opcao = ESC);
END.

```

```

xi, yi,
maxy, miny,
prodx : Real;
p      : Integer;

Function px(xi : Real) : Integer;
Begin
  px := Round((fx * (xi - x[inreg])));
End;

Function py(yi : Real) : Integer;
Begin
  py := Round((fy * (maxy - yi)));
End;

Procedure Plota(x, y : Integer);
Begin
  Draw(x - 4, y, x + 4, y, 1);
  Draw(x, y - 2, x, y + 2, 1);
End;

Begin
  HiRes; ClearScreen;
  Quadro(jxi - 1, jyi - 1, jxf + 1, jyf + 1, 1);
  GraphWindow(jxi, jyi, jxf, jyf);
  fx := 1.0 * (jxf - jxi) / (x[fiareg] - x[inreg]);
  maxy := y[inreg];
  miny := y[finreg];
  For i := inreg + 1 to fiareg do
    If y[i] > maxy then maxy := y[i] else
    If y[i] < miny then miny := y[i];
  fy := 1.0 * (jyf - jyi) / (maxy - miny);
  For i := inreg to fiareg do Plota(px(x[i]), py(y[i]));
  For p := 0 to jxf - jxi do
    begin
      prodx := 1.0;
      yi := 0.0;
      xi := 1.0 * p / fx + x[inreg];
      For i := 0 to grau do
        begin
          yi := yi + a[i] * prodx;
          prodx := prodx * xi;
        end;
      Plot(p, py(yi), 1);
    end;
  Repeat Until KeyPressed;
End;

Procedure Menu;
Begin
  ClrScr;
  WriteLn(' MENU ');
  WriteLn;
  WriteLn('E - Entra dados');
  WriteLn('L - Lista dados');

```

Programa Col100;

```

(-----)
(
(
( Este programa faz aquisicao de espectros do Finnigan 10155/L
(
(
(
(
(
(
(-----)

```

```

const GRAU = 3;
ESC = #27;
DIR = #77;
ESD = #73;
YDIR = #116;
TESQ = #115;
LIMPA = #71;
FIM = #79;
SinalFinnigan = 2;
NScanDef = 10;
NADDef = 42;
GanhoDef = 2;
MzInitial = 3;
MzFinal = 102;

```

```

type VetorRampa = Array[MzInitial..MzFinal] of Integer;
VetorSinal = Array[MzInitial..MzFinal] of Real;
String14 = String[14];
String255 = String[255];

```

```

var c : Array[0..GRAU] of Real;

```

```

nscan,
nad,
mzi,
mzf,
ganho : Integer;
rampa : VetorRampa;
sinal,
back : VetorSinal;
opcao : Char;
ruído,
linhabase : Real;
salvadir : String255;

```

```

($! AUXILIO.CLL)
($! GRAPH.CLL)
($R+)

```

```

Procedure Condicao;

```

```

Begin
nscan := NScanDef;
nad := NADDef;
mzi := MzInitial;
mzf := MzFinal;
ganho := GanhoDef;

```

```

linhabase := 0.0;
End;

```

```

Procedure Condicao;

```

```

Begin
ClrScr;
WriteRes(' CONDIÇÕES');
Write('m/z initial = ',mzi,' d -> '); Readln(mzi);
If not (mzi in[MzInitial..MzFinal]) then mzi := MzInitial;
Write('m/z final = ',mzf,' d -> '); Readln(mzf);
If not(mzf in[MzInitial..MzFinal]) then mzf := MzFinal;
Write('ganho = ',ganho,' -> '); Readln(ganho);
If not(ganho in[0..7]) then ganho := GanhoDef;
Write('numero de conversoes do A/D = ',nad,' -> '); Readln(nad);
Write('numero de espectros acumulados = ',nscan,' -> '); Readln(nscan);
End;

```

```

Procedure CalcRampa;
var arqpar : text;
mz, i : Integer;

```

```

Function Pol(mz : Real) : Integer;
var px, soma : Real;
begin

```

```

px := 1.0;
soma := 0.0;
For i := 0 to GRAU do
begin
soma := soma + c[i] * px;
px := px * mz;
end;
Pol := Round(soma);
end;

```

```

Begin ( CalcRampa )
Assign(arqpar,'PAR100.4');
Reset(arqpar);
For i := 0 to GRAU do Readln(arqpar,c[i]);
Close(arqpar);
For mz := mzi to mzf do rampa[mz] := Pol(mz);
End;

```

```

Procedure Aquis;
var tecla : Char;
mz : Integer;

```

```

Procedure SetMass(m : Integer);
Begin
PortWt[300] := rampa[m];
End;

```

```

Procedure Seletor(canal, ganho : Integer);
Begin
PortWt[307] := canal + ganho * 16;
End;

```



```

WriteLn('Atual', meio);
Write('Entre novo diretório: '); ReadLn(meio);
ClrDir(meio);
GetDir(0, meio);
erro := 10Result;
If erro < 0 then
begin
  Aviso('erro');
  TextColor(31);
end;
WriteLn('Novo diretório: ', meio);
TextColor(7);
{$I+}
Repeat Until Keypressed;
End;

Procedure Menu;
Begin
  Window(1, 80, 25);
  ClrScr;
  WriteLn('MENU');
  WriteLn('Condições de varredura');
  WriteLn('Condições Iniciais');
  WriteLn('Aquisição de um espectro');
  WriteLn('Gráfico do espectro');
  WriteLn('Salvar o espectro');
  WriteLn('Mudança de Diretório');
  WriteLn('ESC para finalizar');
End;

BEGIN
  GetDir(0, salvadir); { Salva o diretório corrente. }
  CondiInitial;
  CalcRaopa;
  Repeat
    Menu;
    Read(kbd, opcao);
    opcao := UpCase(opcao);
    Case opcao of
      'C' : Condição;
      'I' : CondiInitial;
      'A' : Aquis;
      'G' : Gráfico;
      'S' : Salvar;
      'D' : Diretório;
      ESC : Aviso('cuidado');
    else Aviso('inválido');
    end;
  Until opcao = ESC;
  ClrDir(salvadir); { Restaura o diretório. }
END.

```

```

anzant := 50;
mz := anzant;
PosCursor;
Repeat
  GoToXY(1, 15); Write('m/z = ', mz:3, ' d');
  Read(kbd, tecla);
  If (tecla = ESC) and keypressed then
  begin
    Read(kbd, tecla);
    Case tecla of
      DIR : mz := mz + 1;
      ESO : mz := mz - 1;
      TDIR : mz := mz + 10;
      TESO : mz := mz - 10;
      LIMPA : mz := 0;
      FIN : mz := 0;
    else Aviso('inválido')
    end;
    PosCursor;
  end;
  Until tecla = ESC;
  TextMode;
End;

Procedure Salvar;
var arq : Text;
mz : Integer;
meio : String255;
nome : string14;
fator : Real;
Begin
  fator := sinal(mz);
  For m := mz + 1 to mzf do if sinal(m) > fator then fator := sinal(m);
  fator := 10000.0 / fator;
  ClrScr;
  WriteLn('SALVAR');
  WriteLn;
  GetDir(0, meio);
  WriteLn('Diretório ', meio);
  Write('Nome do arquivo: '); ReadLn(nome);
  Assign(arq, nome);
  Rewrite(arq);
  For m := mzi to mzf do
    If sinal(m) > 0.0 then WriteLn(arq, m, ' ', Round(fator * sinal(m)));
  Close(arq);
End;

Procedure Diretório;
var meio : String255;
erro : Integer;
Begin
  ClrScr; TextColor(7);
  {$I-}
  WriteLn('Mudança de Diretório');
  GetDir(0, meio);

```

```

program MD02;
{
-----
Este programa permite graficar o espectro gerado por
COLIBO e MatIratN.
}
const versao = '2.3';
DIR = #77;      COIR = #116;
ESQ = #75;      CESQ = #115;
HOME = #71;     CHOME= #119;
TFIM = #79;     CFIM = #117;
P8UP = #73;     P8DN = #81;
BS = #8;        FF = #12;
SO = #13;       SO = #14;
DC2 = #18;      SPC = #32;
ESC = #27;      F2 = #68;
F1 = #59;       F4 = #62;
F3 = #61;       F6 = #64;
F5 = #63;       F8 = #66;
F7 = #65;       F9 = #67;
F10 = #68;
MAXPIC = 200;

type String14 = String[14];
String255 = String[255];
TipoV200I = Array[1..MAXPIC] of Integer;
TipoV200R = Array[1..MAXPIC] of Real;

var opcao,
    tecla : Char;
    nome : String14;
    az,
    area : TipoV200I;
    massareal : TipoV200R;
    ESresultado,
    np, i : Integer;
    picosenc : Boolean;

($R-)
{$I GRAPH.CLL}
{$I AUXILIO.CLL}
(I IMPRIME.PSL)

Procedure Menu;
Begin
    TextMode(CSB); ClrScr;
    WriteRes('MENU PRINCIPAL');
    WriteLn;
    WriteRes('- Entrar espectro');
    WriteRes('- Graficar espectro');
    WriteRes('- imprimir Tabela');
    WriteRes('- Salvar espectro');
End;

WriteRes('- Finalizar o programa');
End;

Procedure EntraArquivo;
var arqres : Text;
picobase : Integer;
Begin
    WriteLn;
    WriteLn('Qual o nome do arquivo? (deve estar no drive B:)');
    ReadLn(nome);
    Assign(arqres, 'B:' + nome);
    ($I-)
    Reset(arqres);
    ESresultado := 10result;
    ($I+)
    Case ESresultado of
        $01 : begin
            Close(arqres);
            WriteLn('Arquivo inexistente');
            Aviso('erro');
            Delay(1000);
            end;
        0 : begin end;
        else
            begin
                Close(arqres);
                Aviso('erro');
            end;
    end;
    If ESresultado <> 0 then Exit;
    np := 1;
    While not(Eof(arqres)) do
        begin
            ReadLn(arqres,massareal[np],area[np]);
            np := np + 1;
        end;
    np := np - 1;
    Close(arqres);
    picobase := area[1];
    For i := 2 to np do
        If area[i] > picobase then picobase := area[i];
    For i := 1 to np do area[i] := Round((10000.0 * area[i] / picobase));
    WriteLn('Recolhidos ',np,' picos');
    picosenc := TRUE;
End;

Procedure EntraTeclado;
var temp : TipoV200R;
picobase : Real;
Begin
    ClrScr;
    WriteLn('Para finalizar entre com n/z = 0');
    Window(1,WhereY,80,25);
    none := '';
    For np := 1 to MAXPIC do massareal[np] := 0.0;

```

```

np := 0;
Repeat
  np := np + 1;
  Writeln;
  Writeln('      Pico ', np);
  Write(' m/2 = '); Readln(massareal[np]);
  Write('altura = '); Readln(temp[np]);
  Until (np = MAXPIC) OR (massareal[np] = 0.0);
  If massareal[np] = 0.0 then np := np - 1;
  picosenc := TRUE;
  picobase := temp[1];
  For i := 2 to np do
    If temp[i] > picobase then picobase := temp[i];
  For i := 1 to np do areal[i] := Round(10000.0 * temp[i] / picobase);
End;

Procedure EntraEspectro;
Begin
  picosenc := FALSE;
  ClrScr;
  WriteRes('ENTRA ESPECTRO');
  Writeln;
  WriteRes('Disquete');
  WriteRes('Teclado');
  WriteRes('ESC retorna');
  Repeat
    Read(kbd, tecla);
    tecla := UpCase(tecla);
  Until tecla in ['D', 'T', 'ESC'];
  Case tecla of
    'D' : EntraArquivo;
    'T' : EntraTeclado;
  End;
  If picosenc then
    For i := 1 to np do
      m2[i] := Round((Int(massareal[i] * 2.0 + 0.5) + 5.0) * ( * 10.0 / 2.0 ));
    End;
End;

Procedure SalvaArquivo;
var arqres : Text;
Begin
  Writeln;
  Writeln('Qual o nome do arquivo? (deve estar no drive B:)');
  Readln(nome);
  Writeln('Confirma? (S/N)');
  Read(kbd, tecla);
  If UpCase(tecla) < 'S' then Exit;
  Assign(arqres, 'B:' + nome);
  ($I+);
  Rewrite(arqres);
  ESresultado := IOresult;
  ($I+);
  Case ESresultado of
    $IO : begin
      Close(arqres);

```

```

End;

Begin
  marg := SI + '
'; { compactado + 15 espaços em branco }
  Cabecalho(linha);
  WriteIn(lst,marg,
  Relacao dos picos');
  WriteIn(lst);
  WriteIn(lst,marg,topo,topo,topo);
  linha := linha + 3;
  desloc := np div 3;
  If np mod 3 < 0 then desloc := desloc + 1;
  For i := 1 to desloc do
    begin
      Write(lst,marg);
      For coluna := 0 to 2 do
        begin
          d := coluna * desloc + i;
          If d <= np then Write(lst,massareal[d]:6:2, ' ,az(d)/100.0 :5:1,
          ' ');
          area(d)/100.0 :6:2,
          ' ');
        end;
      end;
    end;
  end;
  np := np - 1;
  end;
End;

Begin
  ClrScr;
  WriteRes('TABELA');
  WriteIn;
  WriteRes('Corrigir um dado');
  WriteRes('Listar na Impressora');
  WriteRes('Listar no Video');
  WriteRes('ESC para voltar ao menu anterior');
  Repeat
    Read(kbd, tecla); tecla := UpCase(tecla);
    Until tecla in ['I', 'V', 'C', 'E'];
  Case tecla of
    'C' : Corrigir;
    'I' : Listar;
    'V' : Listar;
  end;
End;

Procedure Grafica;
var cursor, inicio,
    inicioespec,
    fim, fim espec,
    inreg, fimreg,
    picobase, j : Integer;
    fy, fx : Real;
    naddum : Array[1..MAXPIC] of Boolean;
    fundo, branco, massa : Array[1..30] of Byte;

Function y(sinal : Integer) : Integer;
Begin
  y := Round(fy * (picobase - sinal)) + 15;
End;

Function x(sinal : Integer) : Integer;

```

```

End;

Begin
  marg := SI + '
'; { compactado + 15 espaços em branco }
  Cabecalho(linha);
  WriteIn(lst,marg,
  Relacao dos picos');
  WriteIn(lst);
  WriteIn(lst,marg,topo,topo,topo);
  linha := linha + 3;
  desloc := np div 3;
  If np mod 3 < 0 then desloc := desloc + 1;
  For i := 1 to desloc do
    begin
      Write(lst,marg);
      For coluna := 0 to 2 do
        begin
          d := coluna * desloc + i;
          If d <= np then Write(lst,massareal[d]:6:2, ' ,az(d)/100.0 :5:1,
          ' ');
          area(d)/100.0 :6:2,
          ' ');
        end;
      end;
    end;
  end;
  np := np - 1;
  end;
End;

Begin
  ClrScr;
  WriteIn('Qual o numero do dado?');
  ReadIn(i);
  WriteIn('m/z real = ',massareal[i]:6:2, ' d');
  WriteIn('m/z aproximada = ',az(i)/100.0 :5:1, ' d');
  WriteIn('intensidade = ',area(i)/100.0 :6:2, ' %');
  WriteIn;
  WriteRes('Corrigir m/z Real, Intensidade ou Deletar?');
  Repeat
    Read(kbd, tecla); tecla := UpCase(tecla);
    Until tecla in ['I', 'R', 'D'];
  WriteIn;
  Case tecla of
    'R' : begin
      WriteIn('Qual o novo valor de m/z? (em d)');
      ReadIn(massareal[i]);
    end;
  end;
End;

```

```

begin
  i := y(1000 * i);
  Draw(33, j, 34, j, 1);
  Plot(35, j, 0);
  Plot(37, j, 0);
end;

GoToXY(1,2); Write(pibase/100 :3:0, '%');
inreg := 1;
fiareg := np;
While (m[inreg] < inicio) do inreg := inreg + 1;
While (m[fiareg] > fim) do fiareg := fiareg - 1;
For i := inreg to fiareg do
  Draw(x[m[i]], y[0], x[m[i]], y[area[i]], 1); { desenha o espectro }
  Plota; { por causa do cursor }
  PosCursor(0);
End;

Procedure NuẽPico(pico : Integer);
var xp, xc, yp, yc,
    xt, yt,
    linha, coluna,
    m1, m2 : Integer;
    iguais : Boolean;
begin
  xp := x[m[pico]];
  yp := y[area[pico]] - 4; { -4 para ficar um pouco distanciado da ponta }
  xc := xp - 12; { canto superior esquerdo }
  yc := yp - 12;
  GoToXY(1,10); { transforma caracteres em janela grafica }
  GetPic(fundo, 0, 72, 23, 79);
  Write(m[pico] div 10 :3);
  GetPic(massa, 0, 72, 23, 79);
  PutPic(fundo, 0, 79);
  For coluna := 0 to 2 do
    begin
      If coluna = 2 then coluna := -1;
      xt := xc + 10 * coluna;
      linha := 0;
      iguais := FALSE;
      While (linha <= 20) and not(iguais) do
        begin
          yt := yc - 2 * linha;
          GetPic(fundo, xt, yt,
                xt + 23, yt + 7);
          iguais := TRUE; { testa se o fundo esta em branco }
          For j := 1 to 30 do If fundo[j] <> branco[j] then iguais := FALSE;
          If yt < 0 then
            begin
              yt := 0;
              iguais := TRUE;
            end;
          If iguais then
            begin
              Draw(xp, yp, xt + 12, yt + 4, 1); { linha de conexao }
              Plot(xp - 1, yp - 1, 1);
            end;
        end;
      linha := linha + 1;
    end;
  end;
end;

```

```

Write('F1-inicio F2-fim F3-apl. F4-reduz F5-nua. F6-actual. ');
Write(' F7-impr. F10-sai ');
Atualiza(TRUE);
Repeat
  Read(kbd,tecla);
  tecla := UpCase(tecla);
  If (tecla = ESC) and keypressed then
    begin
      Read(kbd,tecla);
    end;
  Case tecla of
    F1 : begin
      inicio := cursor;
      Atualiza(TRUE);
    end;
    F2 : begin
      fim := cursor;
      Atualiza(TRUE);
    end;
    F3 : if picobase > 200 then
      begin
        picobase := picobase div 2;
        Atualiza(FALSE);
        and else Aviso('erro');
      end;
    F4 : if picobase <= 3000 then
      begin
        picobase := picobase * 2;
        Atualiza(FALSE);
        end else Aviso('erro');
      end;
    F5 : if fiareg >= inreg then Nuera else Aviso('erro');
    F6 : Atualiza(TRUE);
    F7 : begin
      Plota; { Faz suair o cursor }
      GoToXY(15,21); Write(' ');
      Window(1,1,80,25);
      GoToXY(1,25);
      Write(' ');
      Write(' ');
      Print;
      GoToXY(1,25);
      Write('F1-inicio F2-fim F3-apl. F4-reduz F5-nua. F6-actual. ');
      Write(' F7-impr. F10-sai ');
      GoToXY(1,3); Window(1,3,80,24);
      GoToXY(15,21); Write('cursor em m/z = ');
      Plota; { reaparece o cursor }
      PosCursor(0);
    end;
  end;
F10: begin end;
DIR : PosCursor(10); { incrementa 1 d }
ESQ : PosCursor(-10); { decrementa 1 d }
CDIR : PosCursor(100); { incrementa 10 d }
CESQ : PosCursor(-100); { decrementa 10 d }
HOME : PosCursor(inicio - cursor); { move para o inicio da janela }
TFIM : PosCursor(fim - cursor); { move para o fim da janela }
CHOME : begin
  inicio := inicioespec;

```

```

Plot(xp + 1, yp - 1, 1);
PutPic(massa, xt, yt + 7); { imprime m/z do pico }
{ nao nuera picos proximos no desenho }
ai := inicio + Round((xt - 36)/fx); { sau da function x }
af := inicio + Round((xt - 12)/fx);
For j := inreg to fiareg do
  If (mz[j] > ai) and (mz[j] < af) then naonua[j] := FALSE;
end;
linha := linha + 1;
end;
If iguais then
  begin
    For i := inreg to fiareg do { nao nuera picos de m/z distantes (<= 4 d )
      If Abs(mz[pico] - mz[i]) <= 40 then naonua[i] := FALSE;
    end
    else naonua[pico] := FALSE;
    If iguais then Exit;
  end;
end;
End;

Procedure Nuera;
var pb : Integer;
tudonua : Boolean;
Begin
  { nao nuera picos com m/z fracionada ou estourados }
  For i := inreg to fiareg do
    If ((mz[i] mod 10) = 0) and (area[i] <= picobase)
      then naonua[i] := TRUE else naonua[i] := FALSE;
    tudonua := FALSE;
  Repeat
    pb := inreg - 1;
  Repeat
    pb := pb + 1;
  Until naonua[pb] or (pb = fiareg);
  If pb = fiareg then
    If naonua[pb] then NuPico(pb) else Exit;
  For i := pb to fiareg do
    If naonua[i] and (area[i] > area[pb]) then pb := i;
  NuPico(pb);
  Until tudonua;
End;

Begin
  inicioespec := (mz[1] div 100) * 100;
  fimdespec := (mz[np] div 100) * 100 + 100;
  inicio := inicioespec;
  fim := fimdespec;
  cursor := (fim - inicio) div 2;
  picobase := 10000;
  HiRes; ClearScreen;
  HiResColor(15);
  GoToXY(1,1); Write(nome);
  Quadrol(-1,9,640,190,1);
  Quadrol(-1,1,640,100,1);
  GoToXY(1,25);

```

Until (UpCase(opcao) = 'S');
END;

pagina: 11

Arquivo: MD2.PAS

```

cursor := (fim - inicio) div 2;
Atualiza(TRUE);
end;
CFIM : begin
    ( fim da janela = fim dos dados )
    fim := fimespec;
    cursor := (fim - inicio) div 2;
    Atualiza(TRUE);
end;

PSUP : begin
    if cursor = fim then cursor := inicio;
    inicio := cursor - 1500; ( janela de 150 d )
    If inicio < inicioespec then inicio := inicioespec;
    fim := inicio + 1500;
    Atualiza(TRUE);
    PosCursor(inicio - cursor);
end;

PSDN : begin
    if cursor = inicio then cursor := fim;
    fim := cursor + 1500;
    If fim > fimespec then fim := fimespec;
    inicio := fim - 1500;
    Atualiza(TRUE);
    PosCursor(fim - cursor);
end;
else Aviso('invalido');
end;
end
else
begin
    tecla := ESC;
    Aviso('invalido');
end;
until tecla = F10;
Window(1,1,80,25);
End;

BEGIN
    EntraEspectro;
    Repeat
        Repeat
            Menu;
            Read(kbd, opcao);
            opcao := UpCase(opcao);
            Case opcao of
                'E' : EntraEspectro;
                'G' : If picosenc then Grafica;
                'S' : If picosenc then SalvaArquivo;
                'T' : If picosenc then ImprimeTabela;
                'F' : Aviso('cuidado');
            else Aviso('invalido');
            end;
        until (opcao = 'F');
    ClrScr;
    Writes('confirma saida? (S/N)');
    Read(kbd, opcao);

```

```

Program MATCol;
{ Este programa faz aquisicao de espectros do MAT 311A }

const SinalMat = 1;
      RampMat = 0;
      GanhoRampa = 2;
      CR = #13;
      ESC = #27;
      MAXPT = 15000;

type Tipo_Ponteiro = ^Tipo_Aquisicao;
     Tipo_Aquisicao = record
       linhar : Integer;
       fim : Integer;
       ganhosinal : Integer;
       sinal : Array[1..MAXPT] of Integer;
       rampa : Array[1..MAXPT] of Integer;
     end;

String14 = String[14];
String255 = String[255];

var linhabase,
    desvio,
    i, n, x : Integer;
    canal,
    ganho : Byte;
    optao,
    tecla : Char;
    aquis : Tipo_Ponteiro;
    marcaheap : ^Char;
    arq : file of Tipo_Aquisicao;
    nome : String14;
    javarreu : Boolean;

{$R+}

{$I GRAPH.CLL}
{$I AUXILIO.CLL}

Procedure Seletor(sel, g : Byte);
Begin
  Port[$307] := sel OR (g shl 4);
end;

Function ADC : Integer;
var ad : Integer;
Begin
  Port[$304] := 0;
  While (Port[$306] > 31) do;
    { Inicia o conversor A/D }
    ad := Port[$303];
  While (Port[$306] > 31) do;
    { Inicia o conversor A/D }
    While (Port[$306] > 31) do;
      { Aguarda a conversao }
      { Le o conversor }
      ad := ad + Port[$305];
    End;
  End;
  Procedure Condicoes;
  Begin
    ClrScr;
    WriteRes('CONDICOES');
    WriteLn;
    Write('Ganho do sinal = ', ganho, ' -> ');
    ReadLn(ganho);
  End;
  Procedure Aquecimento;
  var fatory : Real;
  maior : Integer;
  Function VL : Integer;
  var soma : Real;
  i : Integer;
  Begin
    soma := 0.0;
    For i := 1 to 270 do soma := soma + ADC;
    VL := Round(soma / 270.0);
  End;
  Function y(sinal : Integer) : Integer;
  Begin
    y := Round(fatory * (maior - sinal));
  End;
  Begin { Aquecimento }
    HlRes; ClrScreen;
    GoToXY(1,1);
    WriteLn('Aguardando o aquecimento...');
    WriteLn('Cada linha corresponde a aproximadamente 1 minuto. ');
    WriteLn('Tecle para sair. ');
    Seletor(SinalMat, ganho);
    i := 0;
    maior := VL;
    WriteLn('Linha base inicial = ', maior :5);
    fatory := 98.0 / (maior + 1);
    Quadro(0,99,681,199,1);
    GraphWindow(1,100,600,198);
    Repeat
      Seletor(SinalMat, ganho);
      i := i mod 600 + 1;
      Plot(i, y(VL), 1);
    Until keypressed;
    GraphWindow(0,0,679,199);
  End;
  While (Port[$306] > 31) do;
    { Aguarda a conversao }
    { Le o conversor }
    ad := ad + Port[$305];
  End;
  Port[$304] := 0;

```

```

Begin
  y := (4095 - sinal) div 22;
End;

Procedure PegaPar;
Begin
  With aquis do
  begin
    Seleto(SinalMat, ganho);
    n := n + 1;
    x := n mod 640;
    sinal[n] := (ADC + ADC) shr 2;
    Seleto(RampaMat, GanhoRampa);
    If sinal[n] = 4095 then Sound(300);
    yp := y(sinal[n]);
    yant := y(sinal[n - deslocamento]);
    rampa[n] := (ADC + ADC) shr 2;
  end;
  Seleto(SinalMat, ganho);
  Plot(x,yant,0);
  Plot(x,yp,1);
End;

Begin (Varre)
  HiRes; ClearScreen;
  DetLinhabase;
  If tecla = ESC then Exit;
  Disparo;
  Aviso('Inicio');
  WriteLn('Em aquisicao...');
  With aquis do
  begin
    fim := MAXPT;
    n := 0;
    deslocamento := 0;
    Repeat
      PegaPar;
    Until (n = 640) or KeyPressed;
    deslocamento := 640;
    Repeat
      PegaPar;
    Until (n = fim) or KeyPressed;
    fim := n;
    Aviso('pronto');
    ganhosinal := ganho;
    linear := desvio;
    For i := 1 to fim do
    begin
      sinal[i] := sinal[i] - linhabase;
      If sinal[i] < 0 then sinal[i] := 0;
    end;
    javarre := TRUE;
  end;
End;

```

```

Procedure Varre;
var yant, yp, deslocamento : Integer;

Procedure DetLinhabase;
const N = 500;
var i : Integer;
media,
soma : Real;
sinal : Array[1..N] of Integer;
Begin
  Seleto(SinalMat, ganho);
  WriteLn('Posicione o MAT 311A para m/z inicial.');
```

```

Repeat
  WriteLn('Tecla');
  Read(kbd,tecla);
  soma := 0.0;
  For i := 1 to N do
  begin
    sinal[i] := ADC;
    soma := soma + sinal[i];
  end;
  media := soma / N;
  linhabase := Round(media); ( linha base = media. )
  soma := 0.0;
  For i := 1 to N do soma := soma + sqrt(sinal[i] - media);
  desvio := Round(sqrt(soma) / n); ( 1 desvio padrao/n. )
  If linhabase - desvio <= 0 then
  Begin
    Aviso('Erro');
    WriteLn('Aumente a linha base.');
```

```

  End;
Until (linhabase - desvio > 0) or (tecla = ESC);
ClearScreen; GoToXY(1,1);
WriteLn('Linha base = ',linhabase);
WriteLn('Desvio padrao = ',desvio);
End;

Procedure Disparo;
var azi, soma : Real;
Begin
  Seleto(RampaMat, GanhoRampa);
  soma := 0.0;
  For i := 1 to 20 do soma := soma + ADC;
  azi := soma + 1.5; ( 50% de margem )
  Write('Pode iniciar',CR);
  Repeat
    soma := 0.0;
    For i := 1 to 20 do soma := soma + ADC;
  Until (soma > azi) or KeyPressed;
  Write('
',CR);
End;

Function y(sinal : Integer) : Integer;

```

```

'V' : Varre;
'S' : If javarreu then Salva else Aviso('erro');
ESC : Aviso('cuidado');
      else Aviso('invalido');
      end;
      Until opcao = ESC;
      Release(marcaheap);
END.

```

```

Procedure Salva;
var erro : Integer;
Begin
  Writeln('Qual o nome do arquivo? (drive B)');
  Readln(nome);
  ($I-)
  Assign(arq, 'B:' + nome);
  Rewrite(arq);
  erro := IOResult;
  If erro < 0 then Aviso('erro');
  Case erro of
    0 : begin
      Write(arq, aquis');
      erro := IOResult;
      If erro < 0 then Aviso('erro');
      Case erro of
        0 : Writeln('Espectro salvo. ');
        $F0 : Writeln('Erro de gravacao ou disco cheio');
        else Writeln('Erro');
        end;
        end;
        $F0 : Writeln('Erro de gravacao ou disco cheio. ');
        $F1 : Writeln('Nao existe espaco no diretorio. ');
        else Writeln('Erro');
        end;
        Close(arq);
        ($I+)
      End;
    End;

Procedure Menu;
Begin
  TextMode;
  ClrScr;
  Writeln(' MENU ');
  Writeln;
  Writeln('Muda as Condições');
  Writeln('Aguarda o aquecimento');
  Writeln('Varre um espectro');
  Writeln('Salva o espectro');
  Writeln('ESC finaliza');
  End;

BEGIN
  Mark(marcaheap);
  aquis := nil;
  New(aquis);
  javarreu := FALSE;
  ganho := 3; ( Banho para o sinal. )
  Repeat
    Menu;
    Read(kbd, opcao);
    opcao := UpCase(opcao);
    Case opcao of
      'C' : Condições;
      'A' : Aquecimento;

```

```

aquis      : TipoPonteiro;
marcaheap  : ^Char;
argq       : file of TipoPontoiscao;
greseq,
ESresultado,
incremento,
op,
inicio, final,
inca!, final,
topov,
n, i      : Integer;
temespec,
picosenc  : Boolean;
fatorx,
fatory    : Real;
arqpar    : Text;
rs,
cfr       : TipoLinhav;

{$R+}

{$I GRAPH.CLL}
{$I AUXILIO.CLL}

Procedure CarqPar;
Begin
  Assign(arqpar, 'PARAM.MAT');
  Reset(arqpar);
  For i := 0 to GRAU do Readln(arqpar, cfr,
    Close(arqpar);
    incremento := 11; { Incremento para lin
End;

```

```

type TipoLinhaV = Array[0..GRAU] of Real;
TipoMatriz = Array[0..GRAU] of TipoLinhaV;
String255 = String[255];
String14 = String[14];
TipoBado = Array[1..15000] of Integer;
TipoScan1 = Array[1..MAXPIC] of Integer;
TipoScan = Array[1..MAXPIC] of Real;
TipoPonteiro = ^TipoAquisicao;
TipoAquisicao = Record
    linear : Integer;
    fim : Integer;
    ganhosinal : Integer;
    sinal : TipoBado;
    rampa : TipoBado;
end;

var opcao,
    tecla : Char;
    nome : String14;
    c, m22 : TipoScan1;
    sz,
    h : TipoScan;

Function ValRampa(x : Integer) : Real;
var px, sy : Real;
    i : Integer;
Begin
    px := x;
    sy := rs[0];
    For i := 1 to GRAU do
        begin
            sy := sy + rs[i] * px;
            px := px * x;
        end;
    ValRampa := sy;
End;

Procedure EntraEspectro;
Begin
    timespec := FALSE;
    WriteLn;
    WriteLn('Dual o nome do arquivo? (deve estar no drive B:)');
    ReadLn(nome);
    Assign(arq, 'B:' + nome);
    ($I-)

```



```

CIMA : If passo < 1000 then passo := passo + 2
      else Aviso('invalido');
      8410 : If passo > 1 then passo := passo div 2
      else Aviso('invalido');
      F1 : c := 1;
      F2 : c := 2;
      else Aviso('invalido');
      end;
      Case tecla of
        DIR, ESQ, CODR,
        CESQ : begin
          Plota;
          cursor[c] := cursor[c] + incr * passo;
          If cursor[c] < 1 then cursor[c] := 1;
          If cursor[c] > aquis*.fim then cursor[c] := aquis*.fim;
          Plota;
        end;
      end;
      Until tecla = ESC;
      Draw(x(cursor[1]), 0, x(cursor[1]), 199, 1);
      Draw(x(cursor[2]), 0, x(cursor[2]), 199, 1);
      inicio := cursor[1];
      final := cursor[2];
      GoToXY(1,1);
      Writeln('Inicio do tratamento no dado ', inicio);
      Writeln('Fim do tratamento no dado ', final);
    End;

    Procedure LimpaRaampa;
    var delta : Integer;
    ( 1. Ajusta um polinomio aos dados. )
    ( 2. Utiliza o polinomio para descrever a evolucao do campo magnetico. )

    Procedure Sonatorio;
    var inicio, fim, px, y : Real;
    Begin
      For linha := 0 to GRAU do ( Limpa a matriz do sonatorio. )
        begin
          a[linha] := 0.0;
          For coluna := 0 to GRAU do s[linha, coluna] := 0.0;
        end;
        Writeln('Fazendo as Sonatorias...');
        s[0,0] := 0.0;
        i := inreg + delta;
        Repeat
          s[0,0] := s[0,0] + 1.0;
          y := aquis*.raampa[i];
          a[0] := a[0] + y;
          px := 1.0;
          For q := 1 to GRAU do
            begin
              px := px + i;
              a[q] := a[q] + px * y;
            end;
          end;
        end;
      end;
    End;
  End;
End;

```

```

For linha := 0 to g do
  s[linha,g-linha] := s[linha,g-linha] + px;
end;
For g := 1 to GRAU do
  begin
    px := px * i;
    coluna := GRAU;
    For linha := g to GRAU do
      begin
        s[linha,coluna] := s[linha,coluna] + px;
        coluna := coluna - 1;
      end;
    end;
    i := i + incremento;
    Until i >= (fiareg - delta);
  end;

Procedure TiraMedia;
var i, j : Integer;
    m : Real;
begin
  i := inreg + delta;
  With aquis^ do
    Repeat
      m := 0.0;
      For j := -delta to delta do m := m + rampa[i + j];
      rampa[i] := Round(m / incremento);
      i := i + incremento;
    Until i >= fiareg - delta;
  end;

  Begin
    delta := incremento div 2;
    inreg := inicio;
    fiareg := final;
    TiraMedia;
    Soaatorio;
    gresq := GRAU; Reseq;
    rs := a; ( Atualiza rs. )
    Aviso('pronto');
  end;

Procedure EncPicos;
var inreg, fiareg,
    k : Integer;

Procedure ConferLiniair;
var liniair5, comeco,
    finale, y1, y2, j : Integer;
    car : Char;

Function x(sinal : Integer) : Integer;
begin
  x := Round(fatorx * (sinal - comeco));
end;

```

```

begin (ConferLiniair)
  HiRes;
  comeco := 2;
  finale := aquis^fim;
  Repeat
    WriteLn('Visualizacao do liniair para identificacao de um pico. ');
    Write('Inicio ',comeco, '--> '); ReadLn(comeco);
    If comeco < 2 then comeco := 2;
    Write('fim ',finale, '--> '); ReadLn(finale);
    If finale > aquis^fim then finale := aquis^fim;
    ClearScreen;
    liniair5 := 5 * aquis^liniair;
    topoy := liniair5;
    fatory := 100.0 / topoy;
    fatorx := 639.0 / (finale - comeco);
    With aquis^ do
      begin
        For i := comeco to finale do
          begin
            y1 := y(sinal[i - 1]);
            If y1 < 0 then y1 := 0;
            y2 := y(sinal[i]);
            If y2 < 0 then y2 := 0;
            Draw(x(i-1), y1, x(i), y2, 1);
          end;
          Draw(0, y(aquis^liniair), 639, y(aquis^liniair), 1);
        end;
        GoToXY(1, 1);
        WriteLn('Regiao: ',comeco, ' a ',finale);
        WriteLn('Liniair para identificacao dos picos: ',aquis^liniair);
        Aviso('pronto');
        WriteLn('Deseja corrigir? (S/N)');
        Repeat
          Read(xbd, car);
          car := UpCase(car);
          If car = 'I' then PrtScr;
          Until car in ('S', 'N');
          If car = 'S' then
            begin
              WriteLn('Qual o novo valor?');
              ReadLn(aquis^liniair);
            end;
          end;
          Until car = 'N';
          TextMode;
        end;

        Procedure LocPicos;
        var i, maxnp, incico, fiapico : Integer;
            d : Array[1..100] of Integer;
            j, base : Integer;
            fator : Real;
            tacia : Char;

        Procedure SetaPico(var inreg, fiareg : Integer);

```

```

var i : Integer;
soma, somarea : Real;

Begin
  Conferenciar;
  HiRes;
  With aquis^ do
    begin
      i := inicio + 1; np := 0;
      final := final - 1;
      While ((i < final) and (np < MAXPIC)) do
        If sinal[i] > limiar then
          begin
            inreg := i - 1; { Inicio da regio de um ou mais picos. }
            While ((sinal[i] > limiar) and (i < final)) do i := i + 1;
            fimreg := i; { Fim desta regio. }
            k := fimreg - inreg + 1; { Numero de dados acima do limiar. }
            If k > 3 then LocPicos; { Nao permite a passagem de ruídos. }
          end
          else i := i + 1;
        end;
      { With }
      GoToXY(1,1); Writeln(np, ' picos encontrados');
      If np > 0 then picosenc := TRUE else picosenc := FALSE;
    end;
  End;

  Procedure Normaliza;
  var picobase, fator : Real;
  Begin
    If not(picosenc) then Exit;
    picobase := h[1];
    For i := 1 to np do
      If h[i] > picobase then picobase := h[i];
    end;
    fator := 100.0 / picobase;
    For i := 1 to np do
      h[i] := fator * h[i];
    end;
    Writeln('Espectro normalizado. Intensidade do pico base = ', picobase * 16.0);
  End;

  Procedure Traduz;
  var px, sy : Real;
  i, j : Integer;
  Begin
    For i := 1 to np do
      begin
        px := 1.0;
        sy := 0.0;
        For j := 0 to GRAU do
          begin
            sy := sy + cfr[j] * px;
            px := px * c[j];
          end;
          az[i] := sy;
        end;
        i := 0;
      end;
      Repeat
        i := i + 1;
      until Abs(az[i + 1] - az[i]) < 0.3 then
        begin
          End;
        end;
      GoToXY(60,25); Writeln('Tecla...'); Read(kbd,tecla);
      If Upcase(tecla) = 'I' then Print;
    end;
  End;

```

```

End;

Procedure Somatorio;
var px, x : Real;
    g, i, linha, coluna : Integer;
begin
    For linha := 0 to GRAU do
        begin
            a[linha] := 0.0;
            For coluna := 0 to GRAU do s[linha,coluna] := 0.0;
        end;
    s[0,0] := 1.0 * nc;
    For i := 1 to nc do
        begin
            a[0] := a[0] + mc[i];
            If cc[i] >= 0 then x := cc[i] else x := cc[i] + 65536.0;
            px := 1.0;
            For g := 1 to GRAU do
                begin
                    px := px * x;
                    a[g] := a[g] + px * mc[i];
                    For linha := 0 to g do
                        s[linha,g - linha] := s[linha,g - linha] + px;
                    end;
                end;
            For g := 1 to GRAU do
                begin
                    px := px * x;
                    coluna := GRAU;
                    For linha := g to GRAU do
                        begin
                            s[linha,coluna] := s[linha,coluna] + px;
                            coluna := coluna - 1;
                        end;
                    end;
                end;
            end;
        end;
    End;

Begin {Calibra}
    RecuperarFC43;
    p69 := 1;
    { Localiza o pico base que corresponde a m/z 69. }
    For i := 2 to np do
        If h[i] > h[p69] then p69 := i;
    j := 1;
    { Localiza o pico 502. }
    While (j < np) and (mz[j] < 450.0) do j := j + 1;
    p502 := j;
    For i := j + 1 to np do
        If h[i] > h[p502] then p502 := i;
    Writeln(' A/D Intensidade m/z atual m/z correspondente no FC43 ');
    nc := 0;
    For i := 1 to np do { Identifica os picos do espectro adquirido. }
        begin
            dmin := 1;
            For j := 2 to np do
                If Abs(mfc[i] - mz[j]) < Abs(mfc[i] - mz[dmin]) then dmin := j;
                If Abs(mfc[i] - mz[dmin]) < 1.0 then
                    ConvCM := y;
            end;
        end;
    ConvCM := y;
End;

{ Junta os dois picos em um unico. }
mz[i] := (mz[i] + h[i]) / (h[i] + h[i + 1]);
h[i] := h[i] + h[i + 1];
For j := i + 2 to np do
    begin
        mz[j - 1] := mz[j];
        h[j - 1] := h[j];
    end;
    np := np - 1;
    i := i - 1;
end;
Until i = np - 1;
For i := 1 to np do
    mz2[i] := Round((mz[i] * 2.0 + 0.5) * 5.0); { * 10.0 / 2.0 }
End;

Procedure Calibra;
var nc, dmin,
    j, i,
    npfc,
    p69fc, p502fc,
    p69, p502 : Integer;
    mfc, hfc, mc : TipoScan;
    cc : TipoScan;
begin
    RecuperarFC43;
    var arq : Text;
    Begin
        Assign(arq, 'FC43.CAL');
        Reset(arq);
        npfc := 0;
        While not Eof(arq) do
            begin
                npfc := npfc + 1;
                Readln(arq, mfc[npfc], hfc[npfc]);
            end;
        Close(arq);
        p69fc := 1;
        While Round(mfc[p69fc]) < 69 do p69fc := p69fc + 1;
        p502fc := npfc;
        While Round(mfc[p502fc]) > 502 do p502fc := p502fc - 1;
    End;

    Function ConvCM(c : Integer) : Real;
    var px, x, y : Real;
        i : Integer;
    begin
        px := 1.0;
        y := a[0];
        For i := 1 to GRAU do
            begin
                px := px * x;
                y := y + a[i] * px;
            end;
        ConvCM := y;
    end;
End;

```

```

begin
  nc := nc + 1;
  ac[nc] := afc[i];
  cc[nc] := c[dmn];
  Write(cc[nc]:5, h[nc]:9, l, ' % ', ac[nc]:10, l, ac[nc]:10, l, ' -> ');
  Readln(ac[nc]);
end
else
begin
  Aviso('erro');
  WriteLn('Pico do FC 43 com m/z ', afc[i]:6, l, ' nao encontrado');
end;
end;
if not Confirma('Confirma calibracao?') then exit;
Somatorio;
greesq := GRAU; Reseq;
For i := 0 to GRAU do WriteLn(a[i]);
Repeat Until keypressed;
End;

Begin {Tratados}
  ClrScr;
  { Limites; }
  Inicio := 1;
  final := aquis*.fim;
  Limpaaepa;
  EncPicos;
  Traduz;
  Normaliza;
  Aviso('pronto');
  WriteLn('Tecla para continuar...');
  Read(kbd, tecla);
  If (tipo = 'C') and (np > 0) then Calibra;
End;

Procedure Grafica;

Function y(sinal : Integer) : Integer;
Begin
  y := (100 - sinal) * 2;
End;

Function x(sinal : Integer) : Integer;
Begin
  x := Round(500.0 * sinal / c[np]);
End;

Begin { Grafica }
  HiRes; ClearScreen;
  For i := 1 to np do
    begin
      Draw(x[i], y(0), x[i], y(Round(h[i])), 1);
    end;
  end;
  Read(kbd, tecla);

```

```

End;

Procedure Resultados;
const topo = ' m/z m/z Intensidade(ea %)' ;

Procedure Listav;
var cor : Integer;
Begin
  ClrScr; TextColor(15);
  WriteLn(' Relacao dos picos');
  WriteLn;
  WriteLn('pico , topo);
  Window(1,4,80,25);
  cor := 7; TextColor(cor);
  For i := 1 to np do
    begin
      WriteLn(i:3, ' ', m2[i]:6, ' ', m2[i]/10.0:5, l, ' ', h[i]:6, l);
      If (i mod 15 = 0) and (i < np) then
        begin
          cor := 21 - cor;
          TextColor(cor);
          WriteLn('Tecla para continuar ....');
          Repeat Until KeyPressed;
          GoToXY(1,WhereY - 1);
        end;
    end;
  WriteLn('Tecla para continuar ....');
  Repeat Until KeyPressed;
  TextColor(7);
End;

Procedure Listal;
var marg : String(20);
linha, coluna, destoc, d : Integer;

Procedure Cabecalho(var linha : Integer);
var i : Integer;
Begin
  linha := 1;
  For i := 1 to 4 do WriteLn(lst);
  Write(lst, SI, SO, ' Espectrometro de Massas MAT 311A', CR);
  WriteLn(lst, SI, SO, ' Espectrometro de Massas MAT 311A');
  WriteLn(lst);
  WriteLn(lst, marg, ' Arquivo: ', nome);
  WriteLn(lst);
  linha := linha + 8;
End;

Begin { Listal }
  marg := SI + ' ';
  Cabecalho(linha);
  WriteLn(lst, marg, ' Relacao dos picos');
  WriteLn(lst);
  WriteLn(lst, marg, topo, topo, topo);
  linha := linha + 3;

```

```

desloc := np div 3;
If np mod 3 < 0 then desloc := desloc + 1;
For i := 1 to desloc do
begin
  Write(ist, marg);
  For coluna := 0 to 2 do
  begin
    d := coluna * desloc + i;
    If d <= np then Write(ist, az[d] :6:2, ' ', az2[d]/10.0 :5:1,
      h[d] :6:2, ' ');
  end;
  WriteIn(ist);
  linha := linha + 1;
  If (linha mod 55) = 0 then
  begin
    Write(ist, FF);
    Cabecalho(linha);
    WriteIn(ist, marg, 'Relacao dos picos (continuacao)');
    WriteIn(ist);
    WriteIn(ist, marg, topo, topo, topo);
    linha := linha + 3;
  end;
end;
If (linha mod 55) < 0 then Write(ist, FF, DC2);
End;

Procedure Corrigir;
var altura : Real;
j : Integer;
Begin
  ClrScr;
  WriteIn('Qual o numero do dado?');
  Readln(i);
  WriteIn('m/z real = ', az[i] :6:2, ' d');
  WriteIn('m/z aproximada = ', az2[i]/10.0 :5:1, ' d');
  WriteIn('intensidade = ', h[i] :6:2, ' %');
  WriteIn;
  WriteRes('corrigir m/z Real, Intensidade ou Deletar?');
  Repeat
    Read(kbd, tecla); tecla := UpCase(tecla);
  Until tecla in ['I', 'R', 'D'];
  WriteIn;
  Case tecla of
    'R' : begin
      WriteIn('Qual o novo valor de m/z (em d)');
      Readln(az2[i]);
      az2[i] := Round(1nt(az2[i] * 2.0 + 0.5) * 5.0); { (10.0 / 2.0) }
    end;
    'I' : begin
      altura := h[i];
      WriteIn('Qual o novo valor de intensidade? (em %)');
      Readln(altura);
      If altura > 100.0 then altura := 100.0;
      h[i] := altura;
    end;
  end;
End;

```

```

'0' : begin
  For j := i + 1 to np do
  begin
    h[j - 1] := h[j];
    az[j - 1] := az2[j];
    az2[j - 1] := az2[j];
  end;
  np := np - 1;
end;
End;

Begin ( Resultados )
  ClrScr;
  WriteRes('RESULTADOS');
  WriteIn;
  WriteRes('Corrigir um dado');
  WriteRes('Listar na Impressora');
  WriteRes('Listar no Video');
  WriteRes('ESC para voltar ao menu anterior');
  Repeat
    Read(kbd, tecla); tecla := UpCase(tecla);
  Until tecla in ['I', 'V', 'C', 'ESC'];
  Case tecla of
    'C' : Corrigir;
    'I' : Listar;
    'V' : ListarV;
  end;
End;

End;

Procedure SalvaArquivo;
var arqres : Text;
Begin
  WriteIn;
  WriteIn('Qual o nome do arquivo? (deve estar no drive B:)');
  Readln(nome);
  WriteRes('confirma? (S/N)');
  Read(kbd, tecla);
  If UpCase(tecla) < 'S' then Exit;
  Assign(arqres, 'B:' + nome);
  ($I-);
  Rewrite(arqres);
  ESResultado := IOResult;
  ($I+);
  Case ESResultado of
    $EO : begin
      Close(arqres);
      WriteIn('Disco com problemas ou cheio');
      Aviso('erro');
      Delay(1000);
    end;
    0 : begin end;
  else
    begin
      Close(arqres);
    end;
  end;
End;

```

```

Program Cal4021;
{-----}
{ Este programa faz a aquisicao de um espectro e calibra a rampa }
{ do Finnigan 4021. }
{-----}
{                               CLL 90                               }
{                               Versao 1.1                          }
{-----}

const ESC = #27;
      MAXPIC = 500;
      MAXGRAUCAL = 4;
      ANQCAL = 'F4021.CAL';

type TipoScan = Array[1..MAXPIC] of Real;
   TipoScan1 = Array[1..MAXPIC] of Integer;
   TipoCoef  = Array[0..MAXGRAUCAL] of Real;
   TipoMatriz = Array[0..MAXGRAUCAL] of TipoCoef;

var opcao : Char;
    graucal,
    linear,
    linhabase,
    grau,
    p09fc,
    p502fc,
    npfc,
    np,
    nuns,
    anupico,
    inicio,
    fim,
    incr : Integer;
    hfc, mfc,
    a, h : TipoScan;
    c : TipoScan1;
    a : TipoCoef;

{$R+}
{$I GRAPH.CLL}
{$I AUXILIO.CLL}

Procedure SetDA(v : Integer);
Begin
  PortH[$318] := v;
End;

var potenconv,
    numconvad : Integer;

Procedure SetNumConv(n : Integer);

```

```

Begin
  potenconv := n;
  numconvad := 1 shl n;
End;

Function ADC : Integer;
var p, ad, i : Integer;
Begin
  ad := 0;
  For i := 1 to numconvad do
    begin
      Port[$310] := 0;
      While (Port[$310] < 0) do;
        p := Port[$310];
      If p = 4895 then WriteLn('Conversor saturado');
      ad := ad + p;
    end;
  ADC := ad shr potenconv;
End;

Procedure TelaInicial;
Begin
  TextMode; ClrScr;
  TextColor(15);
  WriteLn('
  CAL4021'); TextColor(7);
  WriteLn('
  WriteLn('Este programa permite fazer a calibracao do Finnigan 4021. ');
  WriteLn('
  WriteLn('O procedimento e'' o seguinte:');
  WriteLn('
  - Mode as condicoes se for desejado. ');
  WriteLn('
  - Introduza a amostra e faca a varredura. ');
  WriteLn('
  - Faca a calibracao manual ou automatica. ');
  WriteLn('
  TextColor(31);
  Write('Tecle para continuar...'); TextColor(7);
  Repeat Until keypressed;
End;

Procedure CondInicial;
var arq : Text;
Begin
  inicio := 4000;
  fim := MAXINT;
  incr := 1;
  nus := 1; { Numero de desvios padroes para determinar o linear. }
  minpico := 10 div incr; { Numero de dados acima de linear para um pico. }
  SetNumConv(1); { Numero de conversoes = 2. }
  graucal := 1; { Grau do polinomio de calibracao. }
  Assign(arq, 'FC43.CAL');
  Reset(arq);
  npfc := 0;
  While not Eof(arq) do
    begin
      npfc := npfc + 1;
      ReadLn(arq, nfc(npfc), hfc(npfc));

```

```

end;
Close(arq);
p69fc := 1;
While Round(nfc(p69fc)) < 69 do p69fc := p69fc + 1;
p502fc := 1;
While Round(nfc(p502fc)) < 502 do p502fc := p502fc + 1;
End;

Procedure MudaCond;
Begin
  ClrScr;
  WriteLn('
  NUDA CONDIÇÕES');
  WriteLn('
  Write('Inicio (' , inicio, ') -> '); ReadLn(inicio);
  Write('Fim (' , fim, ') -> '); ReadLn(fim);
  Write('Incremento (' , incr, ') -> '); ReadLn(incr);
  Write('Numero de desvios padroes para determinar o linear (' , nus, ') -> ');
  ReadLn(nus);
  Write('Numero de dados para caracterizar um pico (' , minpico, ') -> ');
  ReadLn(minpico);
  Write('Grau do polinomio de calibracao (' , graucal, ') -> ');
  ReadLn(graucal);
End;

Procedure VarreEspect;
const NUMDAT = 50;
var i, j, inicpic, ad : Integer;
    k, soma, soesp : Real;
    a : Array[1..NUMDAT] of Integer;
Begin
  ClrScr;
  WriteLn('
  VARRER UM ESPECTRO');
  WriteLn('
  WriteLn('regiao: ' , inicio, ' a ' , fim);
  WriteLn('incremento: ' , incr);
  WriteLn('
  np := 0;
  SetBA(0);
  delay(50); { Aguarda a reacao do quadrupolo. }
  For j := 1 to NUMDAT do a[j] := ADC;
  soma := 0.0;
  For j := 1 to NUMDAT do soma := soma + a[j];
  linbase := Round(soma / NUMDAT);
  soma := 0.0;
  For j := 1 to NUMDAT do soma := soma + sqr(a[j] - linbase);
  linear := Round(nus * sqrt(soma / NUMDAT));
  WriteLn('Numero de conversoes: ' , numconvad);
  WriteLn('Linha base: ' , linbase);
  WriteLn('Linear: ' , linear, ' com ' , nus, ' desvios padroes');
  WriteLn('Numero minimo para identificacao de um pico: ' , minpico);
  WriteLn('
  If not Confirma('Confirma?') then exit;
  WriteLn('
  Write('Introduza a amostra de FC 43 e tecle');
  Repeat Until Keypressed;

```

```

Tab(1); ClrEol; TextColor(31);
Write('Adquirindo...'); TextColor(7);
WriteLn;
WriteLn('Tecla para interromper');
Aviso('Inicio');
i := inicio;
Repeat
  { Varre ate' encontrar um pico. }
  While (i < fim) and (ADC - linhaBase < limiar) do
    begin
      i := i + 1;
      SetDA(i);
    end;
  If i < fim then
    begin
      soma := 0.0;
      somap := 0.0;
      inicpic := i - 1; { Um antes do inicio de um pico. }
      k := 0.0; { Deslocamento }
      ad := ADC - linhaBase;
      While ad > limiar do
        begin
          i := i + 1;
          SetDA(i);
          k := k + 1.0;
          soma := soma + ad;
          somap := somap + k * ad;
          ad := ADC - linhaBase;
        end;
      If Trunc(k) >= minipico then { Se k < minipico entao foi ruido. }
        begin
          np := np + 1;
          c[np] := inicpic + Round(somap / soma);
          h[np] := soma;
          WriteLn(1st, inicpic, k, i, c, '[np] :6, ' h = ', h[np] :6.0);
        end;
    end;
  Until (i = fim) or keypressed;
  SetDA(inicio);
  Aviso('pronto');
End;

Procedure Calibra;
var nc, dain, j, i, p69, p582 : Integer;
cont582 : Real;
ac : TipoScan;
cc : TipoScan;
s : TipoMatriz;
arq : text;

Function ConvCM(c : Integer) : Real;
var px, x, y : Real;
i : Integer;
Begin
  If c >= 0 then x := c else x := c + 65536.0;

```

```

px := 1.0;
y := a[0];
For i := 1 to grau do
  begin
    px := px * x;
    y := y + a[i] * px;
  end;
ConvCM := y;
End;

Procedure Somatorio;
var px, x, y : Real;
g, i, linha, coluna : Integer;
Begin
  For linha := 0 to grau do
    begin
      a[linha] := 0.0;
      For coluna := 0 to grau do s[linha,coluna] := 0.0;
    end;
    s[0,0] := 1.0 * nc;
    For i := 1 to nc do
      begin
        x := ac[i];
        If cc[i] >= 0 then y := cc[i] else y := cc[i] + 65536.0;
        a[0] := a[0] + y;
        px := 1.0;
        For g := 1 to grau do
          begin
            px := px * x;
            a[g] := a[g] + px * y;
            For linha := 0 to g do
              s[linha,g - linha] := s[linha,g - linha] + px;
            end;
          For g := 1 to grau do
            begin
              px := px * x;
              coluna := grau;
              For linha := g to grau do
                begin
                  s[linha,coluna] := s[linha,coluna] + px;
                  coluna := coluna - 1;
                end;
            end;
          end;
        End;

Procedure Reseq;
var i, coluna,
    linha : Integer;
    fator : Real;
Begin
  For linha := 0 to grau do
    begin
      fator := s[linha,linha];
      For coluna := 0 to grau do s[linha,coluna] := s[linha,coluna] / fator;
    end;
  End;

```

```

AssignIarg,ARGCAL);
Rewrite(arq);
For i := 0 to grau do WriteIn(arq,a[i]);
Close(arq);
End;

Procedure Menu;
Begin
  TextMode; ClrScr;
  WriteRes('MENU');
  WriteIn;
  WriteRes('Mudar as condicoes');
  WriteRes('Varrer um espectro');
  WriteRes('Calibrar');
  WriteRes('Finalizar');
End;

BEGIN
  TelaInicial;
  CondInicial;
  Repeat
    Repeat
      Menu;
      opcao := ReadKbd;
    Case opcao of
      'M' : MudaCond;
      'V' : VarreSpec;
      'C' : Calibra;
      'F' : Aviso('cuidado');
    else Aviso('invalido');
    end;
  Until opcao = 'F';
  ClrScr;
  Until Confirma('Confirma finalizacao?');
END.

```

```

a[linha] := a[linha] / fator;
For i := 0 to grau do
  If i <> linha then
    begin
      fator := s[i,linha];
      For coluna := 0 to grau do
        s[i,coluna] := s[i,coluna] - fator * s[linha,coluna];
        a[i] := a[i] - fator * a[linha];
      end;
    end;
End;

Begin (Calibra)
($ p69 := 1;
  ( Localiza o pico base que corresponde a m/z 69. )
  For i := 2 to np do
    If h[i] > h[p69] then p69 := i;
  a[0] := 0.0;
  a[1] := mfc(p69fc) / c(p69);
  For i := 1 to np do
    m[i] := ConvCM(c[i]); ( Faz a primeira identificacao com m/z 69. )
    j := 1;
    ( Localiza o pico 502. )
    While m[j] < 450.0 do j := j + 1;
    p502 := j;
  For i := j + 1 to np do
    If h[i] > h[p502] then p502 := i;
    If c[p502] > 0 then cont502 := c[p502] else cont502 := c[p502] + 65536.0;
    a[1] := (mfc(p502fc) - mfc(p69fc)) / (cont502 - c(p69));
    a[0] := mfc(p502fc) - a[1] * c(p69);
  grau := 1;
  a[0] := 0.58; a[1] := 1.552E-02;
  For i := 1 to np do
    m[i] := ConvCM(c[i]); ( Faz a identificacao com dois pontos. )
  nc := 0;
  For i := 1 to npfc do ( Identifica os picos do espectro adquirido. )
    begin
      dain := 1;
      For j := 2 to np do
        If Abs(mfc[i] - m[j]) < Abs(mfc[i] - m[dain]) then dain := j;
        If Abs(mfc[i] - m[dain]) < 0.5 then
          begin
            nc := nc + 1;
            m[nc] := mfc[i];
            c[nc] := c[dain];
            WriteIn(1st,m[nc]:7:1,c[nc]:7);
          end
        else
          begin
            Aviso('erro');
            WriteIn('Pico do FC 43 com m/z ',mfc[i]:6:1,' nao encontrado. ');
          end;
        end;
      If not Confirma('Confirma calibracao?') then exit;
      grau := grau + 1;
    end;
  Reseq;

```

Programa 6CMS;

```

(-----)
(
( Este programa faz aquisicao de espectros de massas
( enquanto controla o cromatografo.
(
( CIL 90
(
(-----)

```

```

const versao = '1.2';
const ESC = #27;
CR = #13;
MAXNUMPROG = 10;
MAXTABAQ = 21; ( Deve ser 2 * MAXNUMPROG + 1 )
MAXTAB = 42; ( Deve ser 4 * MAXNUMPROG + 2 )
TEMPOMAX = 100.0; ( Limite em minutos para o tempo. )
INJMIN = 0; ( Limites em oC para injetor e separador. )
INJMAX = 399;
SEPMIN = 0;
SEPMAX = 399;
COLMIN = -100;
COLMAX = 400;
VELMAX = 100.0; ( Limite de velocidade de aquecimento em oC/min. )
FLUXOMIN = 0; ( Limites de fluxo de gas em scca He. )
FLUXOMAX = 250;
ARGPADRAO = 'PADRAO';
ARGCAL = 'F4021.CAL';
MAXGRAU = 4;
MZMIN = 10;
MZMAX = 999;
MAXREG = 250000.0; ( Maximo numero de registros de 4 bytes. )

```

```

type TipoNomeArq = String[64];
TipoMZ = Array[MZMIN..MZMAX] of Integer;
Tipos = Array[-4..4] of Integer;
TipoCalibra = Array[0..MAXGRAU] of Real;
ProgScela = Record
    ti : Integer;
    tempoi : Real;
    velaq : Real;
    tf : Integer;
    tempof : Real;
end;
TipoProgSC = Array[1..MAXNUMPROG] of ProgScela;
ProgScela = Record
    mzi,
    mzf : Integer;
    tempoi : Real;
    tempof : Real;
    nuscanv : Integer; ( numero de convercoes 2°nunconv. )
end;
TipoProgMS = Array[1..MAXNUMPROG] of ProgMScela;
ProgValvCela = Record

```

```

    valvula : Char; ( 1 - injetor S - separador )
    ativa : Char; ( A - ativa D - desativa )
    tempo : Real; ( em minutos )
end;
TipoProgValv = Array[0..MAXNUMPROG] of ProgValvCela;
TipoIdentidade = Record
    operador : TipoNomeArq;
    amostra : TipoNomeArq;
end;
TabelaCela = Record
    tempo : Real;
    comando : Char;
    ponteiro : Integer;
end;
TipoTabela = Array[1..MAXTAB] of TabelaCela;
TabelaCela = Record
    vel,
    tempod,
    td : Real;
end;
TipoTabelaAq = Array[1..MAXTABAQ] of TabelaCela;
TipoRegData = Record
    a : Integer;
    h : Integer;
end;
TipoScan = Array[1..1000] of TipoRegData;
TipoMostra = Record
    tipo : Char;
    mzi,
    mzf,
    sensibilidade : Integer;
    haax,
    fator : Real;
    yant : Integer;
end;
TipoExibicao = Array[1..3] of TipoMostra;
var progSC : TipoProgSC;
progMS : TipoProgMS;
progValv : TipoProgValv;
tabela : TipoTabela;
tabelaAq : TipoTabelaAq;
identidade : TipoIdentidade;
c : TipoCalibra;
exibicao : TipoExibicao;
branco, mzf, h : TipoMZ;
reg : TipoScan;
linhabase,
liniar,
janelaneg,
janelapos,
nuscan,
grau,
nustabaq,
nuprogSC,

```

```

nuaprogMS,
nuaprogValv,
fluxo,
temporno,
tempini,
tempsep,
equilibrio : Integer;
ttempo,      ( Tempo total do cromatograma. )
nuareq,
nuadesvio : Real;
opcao : Char;
arqQC : file of TipoRegData;
nosepar,
nosegc : TipoNomeArq;
az : TipoMZ;
desloc : TipoDS;

($R+)
($I AUXILIO.CLL)
($I INTERFAC.CLL)
($I GRAPH.CLL)

```

```

Function ErroIO : Boolean;

```

```

var erro : Integer;

```

```

Begin

```

```

erro := 10result;

```

```

If erro = 0 then ErroIO := FALSE else ErroIO := TRUE;

```

```

If erro < 0 then Aviso('erro');

```

```

Case erro of

```

```

0 : ;

```

```

1 : WriteLn('Erro 1: Arquivo nao existe ou nome ilegal');

```

```

$B : WriteLn('Formato numerico incorreto');

```

```

$99 : WriteLn('Arquivo incompleto');

```

```

$F0 : WriteLn('Problemas na gravacao ou disco cheio');

```

```

$F1 : WriteLn('Diretorio cheio');

```

```

else WriteLn('Erro de E/S');

```

```

end;

```

```

If erro <> 0 then Aguarda(5000);

```

```

End;

```

```

Function ConvMZBin(az : Real) : Integer;

```

```

var px, soma : Real;

```

```

i : Integer;

```

```

Begin

```

```

px := az;

```

```

soma := cf(0);

```

```

For i := 1 to grau do

```

```

begin

```

```

soma := soma + cf[i] * px;

```

```

px := px * az;

```

```

end;

```

```

If soma > MAXINT then soma := soma - 65536.0;

```

```

ConvMZBin := Round(soma);

```

```

End;

```

```

Procedure RecCal;

```

```

var arq : Text;

```

```

m300, i : Integer;

```

```

Begin

```

```

Assign(arq, 'ARQCAL');

```

```

{$I-}

```

```

Reset(arq);

```

```

{$I+}

```

```

If ErroIO then

```

```

begin

```

```

WriteLn('Arquivo de calibracao com problemas');

```

```

Aguarda(5000);

```

```

end;

```

```

If ErroIO then Exit;

```

```

grau := -1;

```

```

While not Eof(arq) do

```

```

begin

```

```

grau := grau + 1;

```

```

ReadLn(arq, c[grau]);

```

```

end;

```

```

Close(arq);

```

```

For i := MZMIN to MZMAX do m3[i] := ConvMZBin(i);

```

```

m300 := ConvMZBin(300.0); ( 0 deslocamento e' gerado para m/z = 300 d. )

```

```

For i := -4 to 4 do desloc[i] := ConvMZBin(300.0 + i * 0.1) - m300;

```

```

End;

```

```

Procedure RecCond;

```

```

var arq : Text;

```

```

nome : TipoNomeArq;

```

```

i : Integer;

```

```

Begin

```

```

ClrScr;

```

```

WriteLn(' CARREGAR ARQUIVO DE PARAMETROS');

```

```

WriteLn;

```

```

Write('Nome do arquivo : ', nosepar, ' -> '); ReadLn(nome);

```

```

If length(nome) > 0 then nosepar := nome;

```

```

Assign(arq, nosepar + '.PAR');

```

```

{$I-}

```

```

Reset(arq);

```

```

{$I+}

```

```

If ErroIO then Exit;

```

```

( Le os parametros de analise. )

```

```

ReadLn(arq, fluxo);

```

```

ReadLn(arq, tempini);

```

```

ReadLn(arq, temporno);

```

```

ReadLn(arq, tempsep);

```

```

ReadLn(arq, equilibrio);

```

```

ReadLn(arq, nuaprogSC);

```

```

For i := 1 to nuaprogSC do with progSC[i] do

```

```

ReadLn(arq, ti, tempoi, velaq, ti, tempof);

```

```

For i := nuaprogSC + 1 to MAXNUMPROG do progSC[i] := progSC(nuaprogSC);

```

```

ReadLn(arq, janelaneg);

```

```

ReadLn(arq, janelapos);

```

```

ReadLn(arq, nuadesvio);

```

```

ReadLn(arq, nuaprogMS);

```

```

For i := 1 to numprogMS do with progMS[i] do
  ReadIn(arq,azi,azf,tempoi,numconv);
For i := numprogMS + 1 to MAXNUMPROG do progMS[i] := progMS[numprogMS];
ReadIn(arq,numprogValv);
For i := 1 to numprogValv do with progValv[i] do
  ReadIn(arq,vaivua,ativa,tempo);
  For i := numprogValv + 1 to MAXNUMPROG do
    progValv[i] := progValv(numprogValv);
  For i := 1 to 3 do with exibicao[i] do
    ReadIn(arq,tipo,azi,azf,sensibilidade);
  Close(arq);
End;

```

```

Procedure Inicializacab;

```

```

  var i : Integer;

```

```

  Begin;

```

```

    ClrScr;

```

```

    GoToXY(32,5); WriteRes('programa GCMS');

```

```

    GoToXY(35,10); Write('versao '); WriteRes(versao);

```

```

    GoToXY(36,20); WriteRes('c) CL');

```

```

    Recal;

```

```

    With identidade do

```

```

      begin

```

```

        operador := '';

```

```

        aostra := '';

```

```

      end;

```

```

    GoToXY(38,23); WriteIn('Tecla para comecar...'); Aguarda(5000);

```

```

    nomepar := ARQPARADQ;

```

```

    RecCondi;

```

```

    nomec := 'TEMP';

```

```

    solenoid := 64;

```

```

    AtivaVento(FALSE);

```

```

    AtivaSubAmb(FALSE);

```

```

    SetIn(tempin);

```

```

    SetCo(tempforno);

```

```

    SetSep(tempsep);

```

```

    SetFluxo(fluxo);

```

```

    AtivaSplit('S','D');

```

```

    SetDA(0);

```

```

    SetNuaConv(0);

```

```

    With progValv[0] do

```

```

      begin

```

```

        vaivua := 'I';

```

```

        ativa := 'D';

```

```

        tempo := 0.0;

```

```

      end;

```

```

    For i := NZMIN to NZMAX do branco[i] := 0;

```

```

  End;

```

```

Procedure PreparaGC(tipo : TipoNomeArq);

```

```

  Begin

```

```

    Case UpCase(tipo[1]) of

```

```

      'I' : begin

```

```

        SetCol(progGC[1],ti);

```

```

        AtivaVento(FALSE);

```

```

    AtivaSubAmb(FALSE);

```

```

    SetFluxo(fluxo);

```

```

    SetIn(tempin);

```

```

    SetSep(tempsep);

```

```

  end;

```

```

  'E' : begin

```

```

    SetCo(tempforno);

```

```

    AtivaVento(TRUE);

```

```

    SetFluxo(fluxo);

```

```

    SetIn(tempin);

```

```

    SetSep(tempsep);

```

```

    AtivaSplit('I','D');

```

```

    AtivaSplit('S','D');

```

```

  end;

```

```

  'D' : begin

```

```

    SetCo(20.0);

```

```

    AtivaVento(FALSE);

```

```

    AtivaSubAmb(FALSE);

```

```

    SetFluxo(0);

```

```

    SetIn(20);

```

```

    SetSep(20);

```

```

    AtivaSplit('I','D');

```

```

    AtivaSplit('S','D');

```

```

  end;

```

```

  else Aviso('invalido');

```

```

  end;

```

```

End;

```

```

Procedure Condicoes;

```

```

  var tecla : Char;

```

```

Procedure CondCrom;

```

```

  var i : Integer;

```

```

  tempo : Real;

```

```

  car : Char;

```

```

  Begin

```

```

    ClrScr;

```

```

    WriteRes(' CONDIC0ES CROMATOGRAFICAS');

```

```

    WriteIn;

```

```

    Write('Fluxo de gas de arraste : scca He');

```

```

    Tab(28); Lei(fluxo,FLUXOMIN,FLUXOMAX); WriteIn;

```

```

    Write('Temperatura do Injetor : 6C');

```

```

    Tab(28); Lei(tempin,INJMIN,INJMAX); WriteIn;

```

```

    Write('Temperatura do forno : 6C');

```

```

    Tab(28); Lei(tempforno,COLMIN,COLMAX); WriteIn;

```

```

    Write('Temperatura do Separador : 6C');

```

```

    Tab(28); Lei(tempsep,SEPMIN,SEPMAX); WriteIn;

```

```

    Write('Tempo de equilibrio : min');

```

```

    Tab(28); Lei(equilibrio,0,MAXINT); WriteIn;

```

```

    WriteIn;

```

```

    Write('Numero de estagios de aquecimento : ');

```

```

    Lei(numprogGC,1,MAXNUMPROG); WriteIn;

```

```

    WriteIn;

```

```

    TextColor(15);

```

```

    WriteIn('Temperatura Tempo Velocidade Temperatura Tempo Tempo');

```

```

WriteLn(' inicial      inicial      final      final total');
WriteLn(' oC          min      oC/min      min      min');
TextColor(7);
ttempo := 0.0;
For i := 1 to numprogSC do with progSC[i] do
begin
  If i = 1 then
  begin
    Tab(3); LeI(tf,COLMIN,COLMAX);
    Tab(14); LeR(tempo,0.0,TENPOMAX,6,2);
  end
  else
  begin
    ti := progSC[i-1].tf;
    tempo := 0.0;
    TextColor(5);
    Tab(3); Write(ti:6);
    Tab(14); Write(tempo:6,2);
    TextColor(7);
  end;
  Tab(25); LeR(velaq,0.1,VELMAX,6,2);
  tempo := (COLMAX - ti) / (velaq + 1E-20);
  If tempo < TENPOMAX then
  begin
    Tab(37); LeI(tf,COLMIN,COLMAX);
  end
  else
  begin
    tf := Round(TENPOMAX * velaq + ti); { Preve a temperatura final. }
    TextColor(5); Tab(37); Write(tf:6); TextColor(7);
  end;
  Tab(47); LeR(tempo,0.0,TENPOMAX,6,2);
  tempo := tempo + (tf - ti) / (velaq + 1E-20) + tempof;
  ttempo := ttempo + tempo;
  TextColor(5); Tab(54); WriteLn(tempo:6,2); TextColor(7);
end;
TextColor(5); Tab(54); WriteLn('-----');
Tab(54); WriteLn(ttempo:6,2); TextColor(7);
WriteLn;
Write('Numero de atuacoes sobre as valvulas: ');
LeI(numprogValv,0,MAXNUMPROG); WriteLn;
If numprogValv > 0 then
begin
  WriteLn;
  WriteLn;
  WriteLn('Injetor/Separador      Ativa/Desativa      inicio (min)');
  For i := 1 to numprogValv do with progValv[i] do
  begin
    Tab(8); Write(valvula); Tab(8);
  end
  Repeat
    car := ReadKbd;
  Until car in ['1','5','CR'];
  Write(car);
  If car < CR then valvula := car;
  Tab(28); Write(ativa); Tab(28);
  Repeat

```

```

car := ReadKbd;
Until car in ['A','D','CR'];
Write(car);
If car < CR then ativa := car;
Tab(43); LeR(tempo,0.0,ttempo,6,2);
WriteLn;
end;
end;
PreparaSC('espera');
Aguarda(5000);
Aviso('pronto');
End;

Procedure CondEsp;
var i : Integer;
janela : Real;
Begin
  ClrScr;
  WriteRes(' CONDICAOES DO ESPECTROMETRO ');
  WriteLn;
  Write('Janela de m/z: ',janelaneg / 10.0:4,1);
  Write(' d a ',janelapos / 10.0:4,1,' d');
  janela := janelaneg / 10.0;
  Tab(16); LeR(janela,-0.4,0.4,4,1);
  janelaneg := Round(10.0 * janela);
  janela := janelapos / 10.0;
  Tab(27); LeR(janela,janelaneg / 10.0,0.4,4,1); WriteLn;
  janelapos := Round(10.0 * janela);
  Write('Numero de desvios padroes da linha base: ');
  LeI(numdesvio,0.0,10.0,7,3); WriteLn;
  WriteLn;
  Write('Numero de programas: '); LeI(numprogMS,1,MAXNUMPROG); WriteLn;
  WriteLn;
  TextColor(15);
  WriteLn(' m/z m/z inicio da Numero de');
  WriteLn(' inicial final aquisicao Conversoes');
  WriteLn(' d d min 2');
  TextColor(7);
  progMS[i].tempo := 0.0;
  For i := 1 to numprogMS do with progMS[i] do
  begin
    LeI(mzi,MZMIN,MZMAX);
    Tab(10); LeI(mzf,mzi,MZMAX);
    Tab(20);
    If i = 1 then
    begin
      TextColor(5);
      Write(tempo:7,2);
      TextColor(7);
    end
    else LeR(tempo,0.0,TENPOMAX,7,2);
    Tab(30); LeI(numconv,0,4);
    WriteLn;
  end;
  end;
  Aguarda(5000);

```

```

WriteRes('
End;

Procedure CondTela;
var i : Integer;
car : Char;
Begin
  ClrScr;
  WriteRes('
  WriteLn;
  WriteLn('Tipo de acompanhamento');
  WriteRes('Cromatograma de uma regioe de m/z');
  WriteRes('Cromatograma de um Ion');
  WriteRes('Espectro de uma regioe');
  WriteRes('Nao ativada');
  WriteLn;
  WriteRes('Janela tipo m/z sensibilidade');
  WriteLn('
  For i := 1 to 3 do with exibicao[i] do
    begin
      Write(i : 4);
      Tab(11); Write(tipo); Tab(11);
      Repeat
        car := ReadKbd;
      Until car in ['C','E','N','I','CR'];
      Write(car);
      If car < CR then tipo := car;
      Tab(19);
      Case tipo of
        'C' : begin
          LeI(mzi,MZMIN,MZMAX);
          Tab(28); LeI(mzf,mzi,MZMAX);
          Tab(41); LeI(sensibilidade,1,10000);
          end;
        'E' : begin
          LeI(mzi,MZMIN,MZMAX - 250);
          Tab(28); LeI(mzf,mzi,mzi + 250);
          Tab(41); LeI(sensibilidade,1,10000);
          end;
        'I' : begin
          LeI(mzi,MZMIN,MZMAX);
          Tab(41); LeI(sensibilidade,1,10000);
          end;
        'N' :
          end;
      WriteLn;
    end;
  End;

Procedure SalvCond;
var arq : Text;
nome : TipoNonehrq;
i : Integer;
Begin
  ClrScr;
  WriteRes('
  WriteLn;
  WriteLn('Nome do arquivo : ',nomepar,' -> '); ReadLn(nome);
  If length(nome) > 0 then nomepar := nome;
  Assign(arq,nomepar + '.PAR');
  ($I-);
  Rewrite(arq);
  ($I+);
  If ErrOrd then Exit;
  { Salva os parametros de analise. }
  WriteLn(arq,fluxo);
  WriteLn(arq,tempini);
  WriteLn(arq,tempfornol);
  WriteLn(arq,tempsep);
  WriteLn(arq,equilibrio);
  WriteLn(arq,nuauprogSC);
  For i := 1 to nuauprogSC do with progSC[i] do
    WriteLn(arq,ti : 4,tempoi : 8:2,velaq : 8:2,tf : 4,tempof : 8:2);
  WriteLn(arq,janelaneg);
  WriteLn(arq,janelapos);
  WriteLn(arq,nuaesvio : 0:3);
  WriteLn(arq,nuauprogMS);
  For i := 1 to nuauprogMS do with progMS[i] do
    WriteLn(arq,mzi : 6,mzf : 6,tempoi : 10:2,nuaconv : 4);
  WriteLn(arq,nuauprogValv);
  For i := 1 to nuauprogValv do with progValv[i] do
    WriteLn(arq,valor,ativa,tempo : 10:2);
  For i := 1 to 3 do with exibicao[i] do
    WriteLn(arq,tipo,mzi : 6,mzf : 6,sensibilidade : 6);
  Close(arq);
End;

Begin (Condicoes)
Repeat
  ClrScr;
  WriteRes('
  WriteLn;
  WriteLn('Arquivo de parametros : ',nomepar);
  WriteLn;
  WriteRes('Mudar arquivo');
  WriteRes('Cromatografo');
  WriteRes('Espectrometro');
  WriteRes('Tela Grafica');
  WriteRes('Salvar arquivo');
  WriteRes('<ESC> volta ao menu inicial');
  tecla := ReadKbd;
  Case tecla of
    'N' : RecCond;
    'C' : CondCrom;
    'E' : CondEsp;
    'G' : CondTela;
    'S' : SalvCond;
    ESC :
      else Aviso('invalido');
  end;
End;

```

```

Aviso('pronto');
End;

Procedure CondTela;
var i : Integer;
car : Char;
Begin
  ClrScr;
  WriteRes('
  WriteLn;
  WriteLn('Tipo de acompanhamento');
  WriteRes('Cromatograma de uma regioe de m/z');
  WriteRes('Cromatograma de um Ion');
  WriteRes('Espectro de uma regioe');
  WriteRes('Nao ativada');
  WriteLn;
  WriteRes('Janela tipo m/z sensibilidade');
  WriteLn('
  For i := 1 to 3 do with exibicao[i] do
    begin
      Write(i : 4);
      Tab(11); Write(tipo); Tab(11);
      Repeat
        car := ReadKbd;
      Until car in ['C','E','N','I','CR'];
      Write(car);
      If car < CR then tipo := car;
      Tab(19);
      Case tipo of
        'C' : begin
          LeI(mzi,MZMIN,MZMAX);
          Tab(28); LeI(mzf,mzi,MZMAX);
          Tab(41); LeI(sensibilidade,1,10000);
          end;
        'E' : begin
          LeI(mzi,MZMIN,MZMAX - 250);
          Tab(28); LeI(mzf,mzi,mzi + 250);
          Tab(41); LeI(sensibilidade,1,10000);
          end;
        'I' : begin
          LeI(mzi,MZMIN,MZMAX);
          Tab(41); LeI(sensibilidade,1,10000);
          end;
        'N' :
          end;
      WriteLn;
    end;
  End;

Procedure SalvCond;
var arq : Text;
nome : TipoNonehrq;
i : Integer;
Begin
  ClrScr;
  WriteRes('
  WriteLn;
  WriteLn('Nome do arquivo : ',nomepar,' -> '); ReadLn(nome);
  If length(nome) > 0 then nomepar := nome;
  Assign(arq,nomepar + '.PAR');
  ($I-);
  Rewrite(arq);
  ($I+);
  If ErrOrd then Exit;
  { Salva os parametros de analise. }
  WriteLn(arq,fluxo);
  WriteLn(arq,tempini);
  WriteLn(arq,tempfornol);
  WriteLn(arq,tempsep);
  WriteLn(arq,equilibrio);
  WriteLn(arq,nuauprogSC);
  For i := 1 to nuauprogSC do with progSC[i] do
    WriteLn(arq,ti : 4,tempoi : 8:2,velaq : 8:2,tf : 4,tempof : 8:2);
  WriteLn(arq,janelaneg);
  WriteLn(arq,janelapos);
  WriteLn(arq,nuaesvio : 0:3);
  WriteLn(arq,nuauprogMS);
  For i := 1 to nuauprogMS do with progMS[i] do
    WriteLn(arq,mzi : 6,mzf : 6,tempoi : 10:2,nuaconv : 4);
  WriteLn(arq,nuauprogValv);
  For i := 1 to nuauprogValv do with progValv[i] do
    WriteLn(arq,valor,ativa,tempo : 10:2);
  For i := 1 to 3 do with exibicao[i] do
    WriteLn(arq,tipo,mzi : 6,mzf : 6,sensibilidade : 6);
  Close(arq);
End;

Begin (Condicoes)
Repeat
  ClrScr;
  WriteRes('
  WriteLn;
  WriteLn('Arquivo de parametros : ',nomepar);
  WriteLn;
  WriteRes('Mudar arquivo');
  WriteRes('Cromatografo');
  WriteRes('Espectrometro');
  WriteRes('Tela Grafica');
  WriteRes('Salvar arquivo');
  WriteRes('<ESC> volta ao menu inicial');
  tecla := ReadKbd;
  Case tecla of
    'N' : RecCond;
    'C' : CondCrom;
    'E' : CondEsp;
    'G' : CondTela;
    'S' : SalvCond;
    ESC :
      else Aviso('invalido');
  end;
End;

```

```

Until tecla = ESC;
End;

Procedure Identificacao;
var s : TipNomeArq;
Begin
  ClrScr;
  Writes(' IDENTIFICACAO DA ANALISE ');
  Writeln;
  With identidade do
    begin
      BufLen := 64;
      Write('Operador: '); HighVideo; Write(operador); LowVideo; Write(' -> ');
      HighVideo; Readln(s); LowVideo;
      If length(s) > 0 then operador := s;
      Writeln;
      Write('Amostra: '); HighVideo; Write(amostra); LowVideo; Write(' -> ');
      HighVideo; Readln(s); LowVideo;
      If length(s) > 0 then amostra := s;
      Writeln;
      BufLen := 255;
      end;
    end;
  End;

Procedure VarreGCMS;
var tecla : Char;
nextc,
inicio,
fim : Integer;
minuto,
dtdt,
tempo,
t0 : Real;
interrucao : Boolean;

Procedure GeraTabela;
var i, j, k, m : Integer;
dtgc, dtas, dtvalv, tempo : Real;
Begin
  i := 1; j := 1; m := 1; k := 0;
  tempo := 0.0;
  Repeat
    If i <= numprogMS then
      dtas := (progMS[i].tempo - tempo) else dtas := 1E+20;
    If j <= numprogValv then
      dtvalv := (progValv[j].tempo - tempo) else dtvalv := 1E+20;
    If m <= numtabaq then
      dtgc := (tabela[m].tempo - tempo) else dtgc := 1E+20;
    If (dtas < dtvalv) and (dtas < dtgc) then
      begin
        k := k + 1;
        With tabela[k] do
          begin
            tempo := progMS[i].tempo;
            comando := 'M'; { MS }
            ponteiro := i;
          end;
          i := i + 1;
        end
      else if dtgc < dtvalv then
        begin
          k := k + 1;
          With tabela[k] do
            begin
              tempo := tabela[m].tempo;
              comando := 'G'; { GC }
              ponteiro := m;
            end;
            m := m + 1;
          end
        else
          begin
            k := k + 1;
            With tabela[k] do
              begin
                tempo := tabela[m].tempo;
                comando := 'G'; { GC }
                ponteiro := m;
              end;
            end
          end;
          m := m + 1;
        end
      else
        begin
          k := k + 1;
          With tabela[k] do
            begin
              tempo := tabela[m].tempo;
              comando := 'G'; { GC }
              ponteiro := m;
            end;
            m := m + 1;
          end
        end;
      end;
    end;
  End;

Until tecla = ESC;
End;

```

```

Procedure SetaTelaBgrafica;
var i : Integer;
Begin
  HiRes;
  HiResColor(7);
  ClearScreen;
  DefineJanela(1,1,501,51,1);
  DefineJanela(1,65,501,115,2);
  DefineJanela(1,129,501,179,3);
  For i := 1 to 3 do
    begin
      GoToXY(1,8 + i);
      With exhibit[i] do
        Case tipo of
          'C' : begin
            Write('Cromatograma ('azf,'a','azf,'d) sens. 'sensibilidade);
            hmax := (janelapos - janelaneg + 1) * 4095.0 / sensibilidade;
            hmax := (azf - azf + 1) * hmax;
            fator := 50.0 / hmax;
            yant := 50;
          end;
          'E' : begin
            Write('Espectro de massas ('azf,'a','azf,'d) sens. 'sensibilidade);
            hmax := (janelapos - janelaneg + 1) * 4095.0 / sensibilidade;
            fator := 50.0 / hmax;
            yant := 50;
          end;
          'I' : begin
            Write('Ion m/z = 'azf,'d sens. 'sensibilidade);
            hmax := (janelapos - janelaneg + 1) * 4095.0 / sensibilidade;
            fator := 50.0 / hmax;
            yant := 50;
          end;
          'N' : Write('Janela desativada');
        end;
      end;
    end;
  End;

```

```

Procedure AguardaInjecao;
{ 1. Abre arquivo para armazenar o cromatograma.
  2. Aguarda o disparo.
  3. Limpa o relógio. }
var c : Char;
Begin
  Assign(arqGC,'GC.SCN');
  Rewrite(arqGC);
  GoToXY(64,1); Write('Base: 'linhabase);
  GoToXY(64,2); Write('Linari: 'linari);
  GoToXY(64,5); Write('Injeção e tecla');
  Repeat
    Status;
    Aguarda(2000);
  Until keypressed;
  Read(kbd,c);
  GoToXY(64,5); Write('ESC interrompe');

```

```

tempo := progValv[i].tempo;
comando := 'V'; { Valvula }
ponteiro := j;
end;
j := j + 1;
end;
tempo := tabela[k].tempo;
Until (i > numprogMS) and (j > numtaban);
tabela[k + 1].tempo := 10*20; { Garante a finalizacao da tabela. }
End;

Procedure ExecutaTab;
Begin
  With tabela[nexec] do
    Case comando of
      'M' : With progMS[ponteiro] do
        begin
          inicio := azf;
          fim := azf;
          SetNumConv(numconv);
          h := branco;
        end;
      'G' : begin
          dTdt := tabela[ponteiro].vel;
          tempo0 := tabela[ponteiro].tempo0;
          t0 := tabela[ponteiro].t0;
        end;
      'V' : With progValv[ponteiro] do AtivaSplit(valvula,ativa);
        end;
    nexec := nexec + 1;
  End;

Procedure AguardaEstabilizar;
Begin
  ClrScr;
  WriteLn(' ESTABILIZACAO ');
  WriteLn;
  Write('Aguardando que o cromatografo se estabilize por ');
  WriteLn('equilíbrio,' minuto(s)...');
  WriteLn;
  WriteLn('Pressione qualquer tecla se desejar ignorar esta estabilizacao');
  WriteLn;
  Repeat
    Status;
    Aguarda(2000);
  Until keypressed or StatusOk;
  ResetClock;
  While not keypressed and (Tempo < equilibrio) do
    begin
      GoToXY(20,20);
      Write(Tempo :4:2,' min');
      Status;
      Aguarda(2000);
    end;
  End;

```

```

Aviso('inicio');
HiResColor(15);
ResetClock;
End;

Procedure CalcLinhaBase;
var i : Integer;
varianca,
media,
soma : Real;
y : Array[1..500] of Integer;
Begin
  SetDA(ConvMIDin(10.0)); ( Posiciona em 10 daltons. )
  delay(100); ( Aguarda reaccao do quadrupolo. )
  For i := 1 to 500 do y[i] := ADC;
  soma := 0.0;
  For i := 1 to 500 do soma := soma + y[i];
  media := soma / 500.0;
  soma := 0.0;
  For i := 1 to 500 do soma := soma + sqr(media - y[i]);
  varianca := soma / 500.0;
  linhaBase := Round(media);
  linha := Round(sqrt(nuadesvio * sqrt((janelapos - janelaneg + 1) * varianca)));
End;

Procedure Relogio(var CX, DX : Integer);
var reg : RegList;
Begin
  reg.AX := $2C00;
  MSBDS(reg);
  CX := reg.CX; ( MSB - hora          LSB - minuto )
  DX := reg.DX; ( MSB - segundo       LSB - centesimo )
End;

Procedure Scan;
var np, m, d : Integer;
y : TipoDS;
soma : Real;
CX, DX : Integer;
regtemp : TipoRegData;
Begin
  nuascan := nuascan + 1;
  np := 0;
  Relogio(CX,DX);
  For m := fim downto inicio do
    begin
      h[m] := 0;
      For d := janelapos downto janelaneg do
        begin
          SetDA(ez[a] + desloca(d)); ( Posiciona o conversor D/A. )
          y[d] := ADC - linhaBase; ( Acumula a intensidade. )
          h[m] := h[m] + y[d];
        end;
      If h[m] > linha then ( Se maior que o limiar entao trata o pico. )
        begin

```

```

      np := np + 1;
      soma := 0.0;
      For d := janelaneg to janelapos do soma := soma + 0.1 * y[d] * d;
      azr[m] := Round(20.0 * (m + soma / (1.0 * h[m]))); ( * 20 para resol. 0.05d )
      reg[np].m := azr[m];
      reg[np].h := h[m];
      WriteLn(1st,azr[m] / 20.0 :6:1, h[m] :0);
    end;
  end;
End;

SetDA(az[fin]);
If np > 0 then
  begin
    regtemp.m := CX; ( Salva o horario. )
    regtemp.h := DX;
    WriteArqGC(regtemp);
    regtemp.m := nuascan; ( Salva o numero do scan e o numero de picos. )
    regtemp.h := np;
    WriteArqGC(regtemp);
    For m := 1 to np do WriteArqGC(reg[m]);
    nuareg := nuareg + 2.0 * np * 4.0;
  end;
End;

Procedure Plota;
var az0, az1,
y, x, m, i : Integer;
soma : Real;
Begin
  For i := 1 to 3 do
    begin
      SetJanela(i);
      With ezibcao[i] do
        Case tipo of
          'C' : begin
              soma := 0.0;
              For m := az1 to azf do soma := soma + h[m];
              x := nuascan mod 500;
              If x = 0 then ClearScreen;
              y := Round(fator * (hmax - soma));
              If y < 0 then y := 0;
              Draw(x - 1, yant, x, y, 1);
              yant := y; ( Atualiza yant )
            end;
          'I' : begin
              x := nuascan mod 500;
              If x = 0 then ClearScreen;
              y := Round(fator * (hmax - h[az1]));
              If y < 0 then y := 0;
              Draw(x - 1, yant, x, y, 1);
              yant := y;
            end;
          'E' : begin
              ClearScreen;
              If inicio > az1 then az0 := inicio else az0 := az1;
              If fim < azf then az1 := fim else az1 := azf;

```

```

For a := m0 to m1 do
begin
  x := (a - m1) * 2;
  y := Round(fator * (hmax - h1e));
  If y < 0 then y := 0;
  Draw(x, 50, x, y, 1);
end;
end;

end;
end;
End;

Procedure Salvar;
var nome : TipoNomeArq;
    arq : Text;
    i : Integer;
Begin
  Repeat
    If Confirma('Deseja salvar o cromatograma?') then
      begin
        Identificacao;
        ClrScr;
        WriteRes('    SALVAR CROMATOGRAMA');
        WriteLn;
        Write('Nome do arquivo : ', nomegc, ' -> '); Readln(nome);
        If length(nome) > 0 then nomegc := nome;
        ($I-)
        Assign(arq, nomegc + '.INF');
        Rewrite(arq);
        If IOResult = 0 then
          begin
            WriteLn(arq, identidade.aastral);
            WriteLn(arq, identidade.operador);
            WriteLn(arq, Date);
            WriteLn(arq, fluxo);
            WriteLn(arq, tempini);
            WriteLn(arq, temporno);
            WriteLn(arq, tempsepi);
            WriteLn(arq, equilibrio);
            WriteLn(arq, nuaprogSC);
            For i := 1 to nuaprogSC do with progSC[i] do
              WriteLn(arq, ti :4, tempoi :8:2, velai :4, tempof :8:2);
            WriteLn(arq, janelaneg);
            WriteLn(arq, janelapos);
            WriteLn(arq, nuedesvio :8:3);
            WriteLn(arq, linbase);
            WriteLn(arq, linlar);
            WriteLn(arq, nuaprogMS);
            For i := 1 to nuaprogMS do with progMS[i] do
              WriteLn(arq, m1 :6, szf :6, tempoi :10:2, nusconv :4);
            WriteLn(arq, nuaprogValv);
            For i := 1 to nuaprogValv do with progValv[i] do
              WriteLn(arq, valvula.ativa, tempo :10:2);
            Close(arq);
          end;
        end;
      end;
    end;
  end;
End;

```

```

Renome(arqGC, nomegc + '.SCN');
($I+)
end;

Until not ErroLD;
End;

Begin (VarreGCMS)
  PrepararGC('inicio');
  GeralTelaInq;
  GeralTela;
  GeralTelaGrafica;
  numscan := 0;
  { Numero do Scan. }
  numreg := 0;
  { Numero de registros salvos. }
  nexec := 1;
  { Passo da tabela a ser executado. }
  CalcLinbase;
  While (tabela[nexec].tempo = 0.0) do ExecutaTab;
  SetDelta([progMS[1].mzf] + desloc[janelapos]);
  AguardaInjecao;
  Repeat
    interrompe := FALSE;
  Repeat
    minuto := Tempo;
    While (minuto >= tabela[nexec].tempo) do ExecutaTab;
    SetCol(dtdt * (minuto - tempo) + t0); { Atualiza a Coluna. }
    Scan;
    Plota;
    If numscan mod 10 = 0 then
      If keypressed then interrompe := TRUE else interrompe := FALSE;
    Until (minuto >= tempo) or (numreg >= MAXREG) or interrompe;
    If interrompe then
      begin
        tecla := ReadKbd;
        If keypressed then tecla := ReadKbd;
        If tecla = 'I' then Print;
      end;
    Until (tempo >= ttempo) or (numreg >= MAXREG) or (tecla = ESC);
    Close(arqGC);
    PrepararGC('espera');
    TextMode; ClrScr;
    If numreg > 0.0 then Salvar else
      begin
        WriteLn('Nao foi adquirido nenhum espectro!');
        Aguarda(5000);
      end;
    end;
  End;

  Procedure Menu;
  Begin
    ClrScr;
    WriteRes('          MENU');
    WriteLn;
    WriteRes('Condicoes');
    WriteRes('Analise');
    WriteRes('Fim');
  End;

```

```

End;
BEGIN
    Inicializacao;
    Repeat
        Repeat
            Menu;
        Repeat
            Status;
            Aguarda(2000);
        Until Keypressed;
        opcao := ReadKbd;
        Case opcao of
            'C' : Condicoes;
            'A' : VarresCMS;
            'F' : Aviso('cuidado');
            else Aviso('invalido');
        end;
        Until opcao = 'F';
    ClrScr;
    Until Confirma('Confirma a finalizacao do programa?');
    PrepararSC('desliga');
END.

```

```

Programa Trata;
{-----}
(
(
Este programa trata o cromatograma gerado por GCMS.
)
(
CLL 98
)
(
)
(
)
{-----}

const versao = '1.2';
ESC = #27;
FF = #12;
DIR = #77;
ESD = #75;
CIMA = #72;
BAIXO = #80;
CEIS = #116;
CESG = #115;
HOME = #71;
FIM = #79;
MAXSCAN = 3640;
MZMIN = 10;
MZMAX = 999;
MAXNUMPROG = 10;

type TipoRegData = Record
    m : Integer;
    h : Integer;
end;

TipoRegDataR = Record
    mz : Real;
    h : Integer;
end;

TipoScan = Array[1..MZMAX] of TipoRegDataR;
TipoNoseArq = String[64];
TipoRegScan = Record
    inicio : Real;
    tempo : Real;
    tic : Real;
end;

TipoPonteiro = ^TipoTic;
TipoPic = Array[1..MAXSCAN] of TipoRegScan;
TipoPontIela = ^TipoTelagr;
TipoTelagr = Array[0..16383] of Byte;
TipoMz = Array[MZMIN..MZMAX] of Integer;
ProgScceia = Record
    ti : Integer;
    tempoi,
    velaq : Real;
    tf : Integer;
    tempof : Real;
end;

TipoProgSC = Array[1..MAXNUMPROG] of ProgScceia;

```

```

tela : TipoPontTela; { Permite salvar a tela grafica tg. }
tg : TipoTelaGr absolute $8000:$8000;

{$N+}
{$I AUXILIO.CLL}
{$I GRAPH.CLL}

Function ErroIO : Boolean;
var erro : Integer;
Begin
    erro := IOresult;
    If erro = 0 then ErroIO := FALSE else ErroIO := TRUE;
    If erro <> 0 then Aviso('erro');
    Case erro of
        0 : ;
        1 : Writeln('Erro 1: Arquivo nao existe ou nome ilegal');
        $10 : Writeln('Formato numerico incorreto');
        $99 : Writeln('Arquivo incompleto');
        $F0 : Writeln('Problemas na gravacao ou disco cheio');
        $F1 : Writeln('Diretorio cheio');
        else Writeln('Erro de E/S');
    end;
    If erro <> 0 then Aguarda(5000);
End;

Procedure Inicializacao;
Begin
    nome := 'GC';
    Assign(arq,nome);
    teainfo := FALSE;
    teacrom := FALSE;
    TextMode;ClrScr;
    GoToXY(33,5); Writeln('PROGRAMA TRATA');
    GoToXY(33,10); Write('versao '); Writeln(versao);
    GoToXY(36,20); Writeln(' (c) CLL ');
    Aguarda(5000);
    Aviso('pronto');
End;

Procedure AbreArq;

Procedure FazIc;
var reg : TipoRegData;
    soma, n : Real;
    i, np : Integer;
    hora, minuto,
    segundo, centesimo : Byte;
    teapant : Real;
Begin
    GoToXY(1,10);
    Reset(arq);
    n := 0;
    pbgc := 0.0;
    teapant := 0.0;
    Repeat

```

```

ProgNScela = Record
    az,
    azf : Integer;
    tempol : Real; { Inicio da aquisicao. }
    numconv : Integer; { numero de convercoes 2*numconv. }
end;

TipoProgMS = Array[1..MAXNUMPROG] of ProgNScela;
ProgValvCela = Record
    valvula : Char; { I - injetor S - separador }
    ativa : Char; { A - ativa D - desativa }
    tempo : Real; { em minutos }
end;

TipoProgValv = Array[0..MAXNUMPROG] of ProgValvCela;
TipoIdentidade = Record
    operador : TipoNomeArq;
    aostra : TipoNomeArq;
end;

TipoCrom = Record
    az,
    janelas,
    posicao,
    pbx,
    pby : Integer;
end;

TipoVarCromIon = Array[1..10] of TipoCromIon;

var progGC : TipoProgGC;
    progMS : TipoProgMS;
    progValv : TipoProgValv;
    identidade : TipoIdentidade;
    arq : File of TipoRegData;
    datanal,
    nome : TipoNomeArq;
    opcao : Char;
    gc : TipoPontTeira;
    marcaheap : ^Char;
    linhabase,
    limiar,
    janelaneg,
    janelapos,
    janelapos,
    nueprogGC,
    nueprogMS,
    nueprogValv,
    fluxo,
    teapini,
    tempsep,
    temporno,
    equilibrio,
    senoraz,
    maioraz,
    nuascan : Integer;
    nuadesvio,
    pbgc : Real;
    teainfo,
    teacrom : Boolean;

```

```

Reset(arq);
($!);
If ErroID then
begin
WriteLn('Erro no arquivo de informacoes!');
limiar := 0;
teminfo := FALSE;
end
else
begin
( Le os parametros de analise. )
Readln(arq, identidade.amostrat);
Readln(arq, identidade.operador);
Readln(arq, datanal);
Readln(arq, fluxo);
Readln(arq, tempini);
Readln(arq, temporno);
Readln(arq, tempoeep);
Readln(arq, equilibrio);
Readln(arq, nuemprogSC);
For i := 1 to nuemprogSC do with progSC[i] do
Readln(arq, ti, tempoi, velaq, tf, tempof);
For i := nuemprogSC + 1 to MAXUMPROS do progSC[i] := progSC[nuemprogSC];
Readln(arq, janelaneg);
Readln(arq, janelapos);
Readln(arq, numdesvio);
Readln(arq, linhabase);
Readln(arq, liniaar);
Readln(arq, nuemprogMS);
For i := 1 to nuemprogMS do with progMS[i] do
Readln(arq, azi, xzi, tempoi, nuconv);
Readln(arq, nuemprogValv);
For i := 1 to nuemprogValv do with progValv[i] do
Readln(arq, valvula, ativa, tempo);
Close(arq);
teminfo := TRUE;
end;
End;

Begin
Repeat
teacrom := FALSE;
CIPScr;
Writes(' ABERTURA DO ARQUIVO DO CROMATOGRAMA');
WriteLn;
Write('Nome: '); Readln(nome);
If nome < '' then
begin
Close(arq);
{$!-}
ReCondi;
Assign(arq, nome + '.SCN');
Reset(arq);
{$!+}
If not ErroID then teacrom := TRUE;
end;
End;

```

```

Read(arq, reg); ( Le o tempo. )
hora := Hi(reg.a);
minuto := Lo(reg.a);
segundo := Hi(reg.h);
centesimo := Lo(reg.h);
Read(arq, reg); ( Le nuSCAN e np )
nuSCAN := reg.a;
If nuSCAN mod 10 = 0 then Write('#');
gc[nuSCAN].inicio := n;
gc[nuSCAN].tempo := hora * 60.0 +
minuto + segundo / 60.0 + centesimo / 6000.0; ( Tempo em minutos. )
np := reg.h;
( with gc[nuSCAN] do
WriteLn(1st, nuSCAN + 1, np + 1, tempo - tempant :0:4);
tempant := gc[nuSCAN].tempo;
soma := 0.0;
For i := 1 to np do
begin
Read(arq, reg);
soma := soma + reg.h;
end;
gc[nuSCAN].tic := soma;
If soma > pbgc then pbgc := soma;
n := n + 2.0 + np; ( Inicio do proximo scan. )
Seek(arq, n);
Until Eof(arq) or (nuSCAN = MAXSCAN);
WriteLn;
If not Eof(arq) then
begin
Aviso('cuidado');
WriteLn('Numero de espectros muito grande!');
end;
GoToY(1, 10);
menoraz := MZMAX;
maioraz := MZMIN;
For i := 1 to nuSCAN do
begin
Seek(arq, gc[i].inicio + 1.0); ( Salta o registro de tempo. )
Read(arq, reg);
np := reg.h;
Read(arq, reg); ( Le o pico de maior m/z. )
If reg.a > maioraz then maioraz := reg.a;
Seek(arq, gc[i].inicio + 1.0 + 1.0 * np);
Read(arq, reg); ( Le o ultimo pico (menor m/z). )
If reg.a < menoraz then menoraz := reg.a;
If i mod 10 = 0 then Write(' ');
end;
End;

Procedure RecCondi;
var arq : Text;
i : Integer;
Begin
Assign(arq, nome + '.INF');
{$!-}

```

```

end;
Until teacrom or (nose = '');
If teacrom then FazTic;
End;

Procedure Graficar;
var cursor,
    cursant,
    atualscan,
    primscan,
    ultscan,
    np, i : Integer;
    picobase,
    linhabase,
    xi, fatorm,
    yi, fatory : Real;
    tecla : Char;
    scan : TipoScan;
    ci : TipoVarCromalon;
    ni : Integer;

Function px(x : Real) : Integer;
Begin
    px := Round(fatorm * (x - xi));
End;

Function py(y : Real) : Integer;
Begin
    py := Round(fatory * (y - yi));
End;

Procedure Plota(cur : Integer);
var x, y, i : Integer;
    tic : Real;
Begin
    SetJanela(3);
    x := px(cur);
    tic := gc[cur].tic;
    If tic > picobase then y := 0 else
        If tic < linhabase then y := 50 else y := py(tic);
    For i := -3 to 3 do Plota(x,y + i,1 - GetDotColor(x,y + i));
    SetJanela(2);
    For y := 0 to 40 do Plota(x,y,1 - GetDotColor(x,y));
    End;

Procedure PosiCursor;
var x : Integer;
Begin
    Plota(cursant);
    Plota(cursor);
    cursant := cursor;
    GoToXY(1,24);
    Write('Cursor', cursor, '<>', gc[cursor].tempo :7:3, ' min');
End;

end;
Until teacrom or (nose = '');
If teacrom then FazTic;
End;

Procedure PosiMapa;
var np, i, j : Integer;
    reg : TipoRegData;
Begin
    SetJanela(2); ClearScreen;
    GoToXY(65,10); Write(maiorm / 20.0 :0:0, ' d');
    GoToXY(65,15); Write(menorm / 20.0 :0:0, ' d');
    GoToXY(1,16); Write(gc[primscan].tempo :7:3, ' min');
    GoToXY(56,16); Write(gc[ultscan].tempo :7:3, ' min');
    fatory := 40.0 / (menorm - maiorm);
    yi := maiorm;
    For i := primscan to ultscan do
        Begin
            Seek(ark,gc[i].inicio + 1); { Salta o registro de tempo. }
            Read(ark,reg);
            np := reg.h;
            For j := 1 to np do
                Begin
                    Read(ark,reg);
                    Plot(px(i),py(reg.m),1);
                End;
            End;
            fatory := 50.0 / (linhabase - picobase);
            yi := picobase;
        End;
    End;

Procedure PosiCrom(mapa : Boolean);
var y : Integer;
    tic : Real;
Begin
    fatorm := 500.0 / (ultscan - primscan);
    xi := primscan;
    fatory := 50.0 / (linhabase - picobase);
    yi := picobase;
    If mapa then PosiMapa else
        Begin
            Plota(cursor);
            Plota(atualscan); { Elimina o cursor do espectro selecionado. }
        End;
    End;
    SetJanela(3);
    ClearScreen;
    For i := primscan to ultscan do
        Begin
            tic := gc[i].tic;
            If tic > picobase then y := 0 else
                If tic < linhabase then y := 50 else y := py(tic);
            Plot(px(i),y,1);
        End;
    End;
    Plota(cursor); { Inicializa o cursor. }
    PosiCursor;
    Plota(atualscan); { Fixa o primeiro scan exibido. }
End;

Procedure ZoomX;
var media : Integer;

```

```

var limi, lias : Real;
xi, xf,
yi, yf,
pb, j, i : Integer;
Begin
  GoToXY(1,1);
  Write('
For i := 0 to 7 do
  begin
    { Determina os limites de x/z para cada uma das casas. }
    limi := i * 64.0 / fatorx + scan[mp].az; { Limite inferior. }
    lias := limi + 64.0 / fatorx; { Limite superior. }
    { Localiza o pico base da regiao. }
    pb := i;
    While (scan[pb].mz > lias) and (pb < np) do pb := pb + 1;
    j := pb;
    While j <= np do with scan[j] do
      begin
        If (mz > lias) and (h > scan[pb].h) then pb := j;
        j := j + 1;
      end;
    With scan[pb] do
      If (mz > limi) and (mz <= lias) then
        begin
          { Calcular os limites da seta. }
          yi := py(h) - 3;
          xi := px(mz);
          yf := -4;
          xf := 64 * i + 32;
          Draw(xi,yi,xf,yf,1);
          GoToXY(8 * i + 1,1);
          Write(mz :7:1);
        end;
      end;
    end;
  end;
End;

Begin (PosiScan)
  GoToXY(1,9); Write('Scan ',n,' <=> ',gc[n].tempo :7:3,' min');
  Plota(atualscan); { Apaga o Anterior. }
  atualscan := n;
  Plota(atualscan);
  Seek(arq,gc[n].inicio + 1); { Salta o registro de tempo. }
  Read(arq,reg);
  np := reg.h;
  yi := 0.0;
  For i := 1 to np do
    begin
      Read(arq,reg);
      scan[i].h := reg.h;
      scan[i].mz := reg.m / 20.0;
      If reg.h > yi then yi := reg.h;
    end;
  SetJanela(1);
  ClearScreen;
  fatory := 50.0 / (0.0 - yi);

```

```

Begin
  media := (ultscan - primscan) div 4;
  If media > 0 then
    begin
      primscan := cursor - media;
      If primscan < 1 then primscan := 1;
      ultscan := cursor + media;
      If ultscan > numscan then ultscan := numscan;
    end
  else
    begin
      primscan := 1;
      ultscan := numscan;
    end;
  PosiCrom(TRUE); { Atualiza o mapa. }
End;

Procedure ZoomV;
var y, media : Real;
Begin
  y := gc[cursor].tic;
  media := (picbase - linhabase) / 4.0;
  If media > 1.0 then
    begin
      linhabase := y - media;
      If linhabase < 0.0 then linhabase := 0.0;
      picbase := y + media;
      If picbase > pbgc then picbase := pbgc;
    end
  else
    begin
      linhabase := 0.0;
      picbase := pbgc;
    end;
  end;
End;

PosiCrom(FALSE);
End;

Procedure PosiScan(n : Integer);
var x, i : Integer;
reg : TipoRegData;
fatory,
fatorx,
xi, yi : Real;
Function px(x : Real) : Integer;
Begin
  px := Round(fatorx * (x - xi));
End;
Function py(y : Real) : Integer;
Begin
  py := Round(fatory * (y - yi));
End;
Procedure NumerarPicos;

```

```

xi := scan[np].az - 1.0; { Começa um dalton antes. }
fatorx := 500.0 / (scan[i].az - xi + 1.0); { Trazinha um dalton depois. }
For i := 1 to np do with scan[i] do Draw(px[mz], 50, px[mz], py(h), i);
Numerapicos;
End;

```

```

Procedure SalvaScan(n : Integer);
var np, i : Integer;
  reg : TipoRegData;
  arqscan : Text;
Begin

```

```

  Seek(arq, gc^[n].inicio + 1); { Salta o registro de tempo. }
  Read(arq, reg);
  np := reg.h;
  For i := 1 to np do

```

```

    begin
      Read(arq, reg);
      scan[i].h := reg.h;
      scan[i].az := reg.m / 20.0;
    end;
  Assign(arqscan, 'SCAN' + Conv(S(n));
  Rewrite(arqscan);
  Writeln(arqscan, nome);
  Writeln(arqscan, n);
  Writeln(arqscan, gc^[n].tempo);
  Writeln(arqscan, linia);
  For i := np downto 1 do with scan[i] do
    Writeln(arqscan, az : 6:2, h : 6);
  Close(arqscan);
End;

```

```

Procedure Cromats;
var j, i,
  ion, cursa,
  cury, curyant,
  cursor, cursant,
  scani, scanf : Integer;
  yi, fatory,
  xi, fatorx : Real;
  yant, y, branco : TipoMZ;
  pb : Array[4..6] of Integer;
  tecla : Char;
Begin

```

```

  Function px(x : Real) : Integer;
  Begin
    px := Round(fatorx * (x - xi));
  End;

  Function py(y : Real) : Integer;
  Begin
    py := Round(fatory * (y - yi));
  End;

```

```

  Procedure LocPibase;
  var reg : TipoRegData;

```

```

  Begin
    janel := c[i].janela;
    az := c[i].az;
    yi := pb[janela];
    fatory := 50.0 / (0.0 - yi + 1.0);
    SetJanela(janela);
  End;

```

```

  np, i, j, s : Integer;
  Begin
    Writeln;
    Writeln( Localizando o pico base para cada janela... );
    Writeln;
    For i := 1 to ni do ci[i].pby := 0;
    For s := scani to scanf do
      begin
        If (s - scani) mod 10 = 0 then Write('#');
        Seek(arq, gc^[s].inicio + 1); { Salta o registro de tempo. }
        Read(arq, reg);
        np := reg.h;
        y := branco; { Limpa o vetor. }
        For i := 1 to np do { Recolhe o espectro. }
          begin
            Read(arq, reg);
            y[Round(reg.m / 20.0)] := reg.h;
          end;
        For j := 1 to ni do with ci[j] do
          If y[mz] > pby then
            begin
              pbx := s;
              pby := y[mz];
            end;
          end;
        pb[4] := 0;
        pb[5] := 0;
        pb[6] := 0;
        For j := 1 to ni do with ci[j] do
          If pby > pb[janela] then pb[janela] := pby;
        End;

```

```

Procedure PlotaCrom;
var i, s, az, janela, np : Integer;
  reg : TipoRegData;
  Begin
    Yant := branco; { Limpa y anterior. }
    For s := scani to scanf do
      begin
        Seek(arq, gc^[s].inicio + 1); { Salta o registro de tempo. }
        Read(arq, reg);
        np := reg.h;
        y := branco; { Limpa o vetor. }
        For i := 1 to np do { Recolhe o espectro. }
          begin
            Read(arq, reg);
            y[Round(reg.m / 20.0)] := reg.h;
          end;
        For i := 1 to ni do
          begin
            janela := ci[i].janela;
            az := ci[i].az;
            yi := pb[janela];
            fatory := 50.0 / (0.0 - yi + 1.0);
            SetJanela(janela);
          end;
        End;

```

```

Draw(px(s - 1),py(yant[az]),px(s),py(y[az]),1);
end;
yant := y; { Atualiza yant. }
end;
End;

Procedure Plota(curs, cury : Integer);
var x, d, i : Integer;
Begin
  x := px(curs);
  For i := 4 to 6 do
    begin
      SetJanela(i);
      For d := -3 to 3 do Plota(x,cury + d,1 - GetDotColor(x,cury + d));
      For d := -3 to 3 do Plota(x + d,cury,1 - GetDotColor(x + d,cury));
    end;
  end;
End;

```

Procedure PosiCursor;

```

var x : Integer;
Begin
  Plota(cursant,curyant);
  Plota(cursor,cury);
  cursant := cursor;
  curyant := cury;
End;

```

Begin (Cronats)

```

For i := MIN to MAX do branco[i] := 0;
tela := tq; { Salva a tela grafica. }
TextMode;
WriteRes;
Write;
Write('Numero de ions: '); Lei(ni,0,10); Writein;
If ni > 0 then
  begin
    scani := priescan;
    scant := ultescan;
    Write('Primeiro scan: '); Lei(scani,1,nuascan); Writein;
    Write('Ultimo scan: '); Lei(scant,1,nuascan); Writein;
    Writein;
    Writein(' a/z Janela');
    For i := 1 to ni do
      begin
        Lei(cil[i],az,Round(menoraz / 20.0),Round(maioraz / 20.0));
        Tab(7); Lei(cil[i],Janela,4,6);
        Writein;
        cil[i].posicao := 0;
      end;
    LocPicoBase;
    HiRes;
    DefineJanela(1,10,50,60,4);
    DefineJanela(1,74,50,124,5);
    DefineJanela(1,130,50,180,6);
    GoToXY(65,2); Write(pb[4]);
  end;

```

```

GoToXY(65,10); Write(pb[5]);
GoToXY(65,18); Write(pb[6]);
fatorx := 500.0 / (scant - scani);
xi := scani;
PlotaCrom;
GoToXY(1,25);
Write(' <Seleciona <+>->move <T>raca <E>fetiva <I>apriee <A>bandona');
ion := 1;
CURSM := 1;
cursor := cil[ion].pbx;
cursant := cursor;
yi := pb[cil[ion].Janela];
fatory := 50.0 / (0.0 - yi + 1.0);
cury := py(cil[ion].pby);
curyant := cury;
Plota(cursor,cury); { Inicializa o cursor. }
Repeat
  GoToXY(curs0,8 + (cil[ion].Janela - 4) + 1);
  Write(cil[ion].az :3);
  tecla := ReadKbd;
  If (tecla = ESC) and keypressed then
    begin
      Read(kbd,tecla);
      Case tecla of
        DIR : If cursor < scant then cursor := cursor + 1 else
          Aviso('erro');
        ESD : If cursor > scani then cursor := cursor - 1 else
          Aviso('erro');
        CDIR : If cursor < scant then cursor := cursor + 10 else
          Aviso('erro');
        CESD : If cursor > scani then cursor := cursor - 10 else
          Aviso('erro');
        CIMA : If cury > 0 then cury := cury - 1 else Aviso('erro');
        BAIXO : If cury < 50 then cury := cury + 1 else Aviso('erro');
        HOME : cursor := scani;
        FIM : cursor := scant;
        else Aviso('invalido');
      end;
    end
  end
else
  Case tecla of
    'I' : begin
      Plota(cursor,cury);
      GoToXY(1,25);
      Write('
      Print;
      GoToXY(1,25);
      Write(' <Seleciona <+>->move <T>raca <E>fetiva <I>apriee <A>bandona');
      Plota(cursor,cury);
    end;
    'S' : begin { Seleciona o ion. }
      GoToXY(curs0,8 + (cil[ion].Janela - 4) + 1);
      Write(' ');
      ion := ion mod ni + 1;
      cursor := cil[ion].pbx;
    end;
  end;
end;

```

```

yi := pb(cilom).janela);
fatory := 50.0 / (0.0 - yi + 1.0);
cury := py(cilom).pby);
end;
'+' : begin
  GoToXY(cursa, 8 * (cilom).janela - 4) + 1);
  Write(' ');
  If cursa < 61 then cursa := cursa + 1 else Aviso('erro');
end;
'-' : begin
  GoToXY(cursa, 8 * (cilom).janela - 4) + 1);
  Write(' ');
  If cursa > 1 then cursa := cursa - 1 else Aviso('erro');
end;
'T' : With cilom do
  begin
    posicao := cursa;
    Plota(cursa, cury);
    SetJanela(janela);
    Draw(px(cursor), cury, cursa * 8 + 8, -4, 1);
    Plota(cursa, cury);
  end;
'E' : begin
  For i := 0 to 2 do
    begin
      GoToXY(1, 8 * i + 1);
      For j := 1 to 64 do Write(' ');
    end;
  end;
  For i := 1 to ni do
    If cilom.posicao > 0 then
      begin
        GoToXY(cilom.posicao, 8 * (cilom).janela - 4) + 1);
        Write(cilom.az : 3);
      end;
    end;
  end;
  'A' : Aviso('pronto');
  else Aviso('invalido');
end;
PosiCursor;
Until tecla = 'A';
end
else Wires;
tecla := tecla; ( Reconpo a tela grafica. )
End;

Procedure LocPicos;
var temporet, integral, somat, soma, liniair, tic : Real;
k, fim, y : Integer;
Begin
  tela := tq; ( Salva a tela grafica. )
  GraphWindow(0, 0, 639, 199);
  ClearScreen;
  SetJanela(1);
  fatory := 500.0 / gc(numscan).tempo; ( x em minutos. )
  xl := 0.0;

```

```

fatory := 50.0 / (linhabase - picobase);
yi := picobase;
For i := 1 to numscan do
  begin
    tic := gc[i].tic;
    If tic > picobase then y := 0 else
      If tic < linhabase then y := 50 else y := py(tic);
    Plot(px(gc[i].tempo), y, 1);
  end;
liniair := 0.0; { Determina a linha base pela media dos 30 primeiros scans. }
If numscan > 30 then fim := 30 else fim := numscan;
For i := 1 to fim do liniair := liniair + gc[i].tic;
liniair := liniair / 30.0;
GoToXY(1, 0);
i := 31;
Repeat
  While (i <= numscan) and (gc[i].tic <= liniair) do i := i + 1;
  soma := 0.0;
  somat := 0.0;
  k := 0;
  While (i <= numscan) and (gc[i].tic > liniair) do
    begin
      soma := soma + gc[i].tic;
      somat := somat + gc[i].tic * gc[i].tempo;
      k := k + 1;
      i := i + 1;
    end;
  If k > 3 then
    begin
      temporet := somat / soma;
      integral := soma;
      Draw(px(temporet), 0, px(temporet), 50, 1);
      WriteLn('Tempo de retencao: ', temporet : 8.3, ' min   integracao: ',
        integral : 0.0);
    end;
  Until i >= numscan;
Repeat Until keypressed;
tq := tela; ( Reconpo a tela grafica. )
End;

Procedure Imprime;
Begin
  Plota(cursor);
  GoToXY(1, 24);
  Write('
GoToXY(1, 24);
Write(nome);
GoToXY(1, 23);
Write('
Print;
GoToXY(1, 24);
Write('
GoToXY(1, 25);
Write('E') spectro <S> salvar zoom <X> Y) <R> restaura <I> imprime <D> oc <F> im');
Plota(cursor);

```

```

'S' : SalvaScan(cursor);
'I' : Imprime;
'C' : Croma;
'L' : LeCroma;
'F' : Aviso('pronto');
else Aviso('invalido');
end;
PosiCursor;
Until tecla = 'F';
TextMode;
End;

Procedure Informacoes;
var margem : tipoNomeArq;
i : Integer;
tempo, tempo : Real;
Begin
  ClrScr;
  Writeln('RELATOKID DE INFORMACOES');
  Writeln;
  Writeln('Ligue a impressora...');
  Writeln;
  If not Confirma('Confirma a impressao?') then Exit;
  margem := ' ';
  Writeln(1st,margem,'UFPA Finnigan 8C/MS CL1');
  Writeln(1st);
  Writeln(1st,margem,'Mostra: ',identidade.mastra);
  Writeln(1st,margem,'Operador: ',identidade.operador);
  Writeln(1st,margem,'Data: ',data);
  Writeln(1st);
  Writeln(1st,margem,'CROMATOGRAFIA');
  Writeln(1st,margem,'Vazao do gas de arraste: ',fluxo,' ml/min');
  Writeln(1st,margem,'Temperatura do injetor: ',tempini,' oC');
  Writeln(1st,margem,'Temperatura do forno: ',tempforno,' oC');
  Writeln(1st,margem,'Temperatura do separador: ',tempsep,' oC');
  Writeln(1st);
  Writeln(1st,margem,'Temperatura Tempo Velocidade Temperatura Tempo Tempo');
  Writeln(1st,margem,' inicial initial final final total');
  Writeln(1st,margem,' oC min oC/min oC min min');
  ttempo := 0.0;
  For i := 1 to numprogSC do with progSC[i] do
    begin
      tempo := tempoi + (tf - ti) / (velaq + IE-20) + tempof;
      ttempo := ttempo + tempo;
      Writeln(1st,margem,ti :9,tempoi :11:2,
        velaq :11:2,tf :9,tempof :13:2,tempo :7:2);
    end;
  Writeln(1st,margem); For i := 1 to 54 do Writeln(1st,' ');
  Writeln(1st,'-----');
  Writeln(1st,margem); For i := 1 to 54 do Writeln(1st,' ');
  Writeln(1st,ttempo :6:2);
  Writeln(1st);
  If numprogValv > 0 then

```

```

End;

Begin (Graficar)
  HiRes;
  DefineJanela(1,10,501,60,1);
  DefineJanela(1,75,501,115,2);
  DefineJanela(1,129,501,179,3);
  ni := 0; { Usado em Croma. }
  For i := 1 to 10 do
    begin
      c[i].az := 28;
      c[i].janela := 4;
    end;
  GoToY(1,25);
  Write('Espectro <Salvar zoom><Y> <Restaura <Y>apriar <Y>oc <Y>in');
  picobase := pbqc;
  linhabase := 0.0;
  priScan := 1;
  ultScan := numScan;
  cursor := numScan div 2;
  cursant := cursor;
  atualScan := cursor;
  PosiCrom(TRUE);
  PosiScan(atualScan);
  Repeat
    tecla := ReadKbd;
    If (tecla = ESC) and keypressed then
      begin
        Read(Kbd,tecla);
        Case tecla of
          DIR : If cursor < ultScan then cursor := cursor + 1 else
            Aviso('erro');
          ESQ : If cursor > priScan then cursor := cursor - 1 else
            Aviso('erro');
          CUR : If cursor <= ultScan - 10 then cursor := cursor + 10 else
            Aviso('erro');
          CESQ : If cursor >= priScan + 10 then cursor := cursor - 10 else
            Aviso('erro');
          HOME : cursor := priScan;
          FIM : cursor := ultScan;
          else Aviso('invalido');
        end;
      end;
  else
    Case tecla of
      'X' : ZoomX;
      'Y' : ZoomY;
      'R' : begin
        priScan := 1; { Restaura os limites. }
        ultScan := numScan;
        picobase := pbqc;
        linhabase := 0.0;
        PosiCrom(TRUE);
      end;
      'E' : PosiScan(cursor);
    end;
  end;

```

```

begin
  WriteLn(1st,margem,'Injetor/Seperador      Ativa/Desativa      Inicio (min)');
  For i := 1 to numprogValv do with progValv[i] do
    WriteLn(1st,margem,'      ',valvula,
      '      ',ativa,tempo :21:2);
  WriteLn(1st);
end;

```

```

WriteLn(1st,margem,'ESPECTROMETRO');
WriteLn(1st,margem,'Janela de a/z: ',janelaneg / 10.0 :0:1,
  ' d a ',janelapos / 10.0 :0:1, ' d');
WriteLn(1st,margem,'Linha base: ',linbase);
WriteLn(1st,margem,'Desvios padroes da linha base: ',nuadesvio :0:3);
WriteLn(1st);
WriteLn(1st,margem,'a/z      a/z      inicio da Numero de');
WriteLn(1st,margem,'inicial      final      aquisicao      Conversoes');
WriteLn(1st,margem,'d      d      min      2^');
For i := 1 to numprogMS do with progMS[i] do
  WriteLn(1st,margem,'a/z i6,a/z i10,tempo i11:2,nuaconv i9);
  WriteLn(1st,FF);
End;

```

```

Procedure Menu;
Begin
  ClrScr;
  WriteRes('      MENU');
  WriteLn;
  WriteLn('Arquivo de trabalho: ',nome);
  WriteLn;
  WriteRes('Abrir arquivo');
  WriteRes('Gravar');
  WriteRes('Imprimir informacoes');
  WriteRes('Finalizar');
End;

```

```

BESIN
  Mark(marcaheap);
  gc := nil;
  New(gc);
  New(tela);
  Inicializacao;
  AbreArq;
  Repeat
    Repeat
      Menu;
      opcao := ReadKbd;
    Case opcao of
      'A' : AbreArq;
      'G' : If teacron then Gravar else Aviso('erro');
      'I' : If teinfo then Informacoes else Aviso('erro');
      'F' : Aviso('cuidado');
      else Aviso('invalido');
    end;
  Until opcao = 'F';
  ClrScr;

```

```

Arquivo: TRATA.PAS
pagina: 18
begin
  WriteLn(1st,margem,'Injetor/Seperador      Ativa/Desativa      Inicio (min)');
  For i := 1 to numprogValv do with progValv[i] do
    WriteLn(1st,margem,'      ',valvula,
      '      ',ativa,tempo :21:2);
  WriteLn(1st);
end;
WriteLn(1st,margem,'ESPECTROMETRO');
WriteLn(1st,margem,'Janela de a/z: ',janelaneg / 10.0 :0:1,
  ' d a ',janelapos / 10.0 :0:1, ' d');
WriteLn(1st,margem,'Linha base: ',linbase);
WriteLn(1st,margem,'Desvios padroes da linha base: ',nuadesvio :0:3);
WriteLn(1st);
WriteLn(1st,margem,'a/z      a/z      inicio da Numero de');
WriteLn(1st,margem,'inicial      final      aquisicao      Conversoes');
WriteLn(1st,margem,'d      d      min      2^');
For i := 1 to numprogMS do with progMS[i] do
  WriteLn(1st,margem,'a/z i6,a/z i10,tempo i11:2,nuaconv i9);
  WriteLn(1st,FF);
End;

```

```

Procedure Menu;
Begin
  ClrScr;
  WriteRes('      MENU');
  WriteLn;
  WriteLn('Arquivo de trabalho: ',nome);
  WriteLn;
  WriteRes('Abrir arquivo');
  WriteRes('Gravar');
  WriteRes('Imprimir informacoes');
  WriteRes('Finalizar');
End;
BESIN
  Mark(marcaheap);
  gc := nil;
  New(gc);
  New(tela);
  Inicializacao;
  AbreArq;
  Repeat
    Repeat
      Menu;
      opcao := ReadKbd;
    Case opcao of
      'A' : AbreArq;
      'G' : If teacron then Gravar else Aviso('erro');
      'I' : If teinfo then Informacoes else Aviso('erro');
      'F' : Aviso('cuidado');
      else Aviso('invalido');
    end;
  Until opcao = 'F';
  ClrScr;

```



```

var teep : tipoV200R;
picobase : Real;
Begin
  ClrScr;
  WriteLn('Para finalizar entre com m/z = 0');
  Window(1,WhereY,80,25);
  nome := '';
  For np := 1 to MAXPIC do massareal[np] := 0.0;
  np := 0;
  Repeat
    np := np + 1;
    WriteLn;
    WriteLn('      Pico ',np);
    Write('      m/z = '); ReadLn(massareal[np]);
    Write('      altura = '); ReadLn(teep[np]);
  Until (np = MAXPIC) OR (massareal[np] = 0.0);
  If massareal[np] = 0.0 then np := np - 1;
  picosenc := TRUE;
  picobase := teep[1];
  For i := 2 to np do
    If teep[i] > picobase then picobase := teep[i];
  For i := 1 to np do area[i] := Round(10000.0 * teep[i] / picobase);
End;

Procedure EntraEspectro;
Begin
  picosenc := FALSE;
  ClrScr;
  WriteRes('ENTRA ESPECTRO');
  WriteLn;
  WriteRes('Disquete');
  WriteRes('Teclado');
  WriteRes('ESC retorna');
  Repeat
    Read(kbd, tecla);
    tecla := UpCase(tecla);
  Until tecla in ['D', 'T', 'ESC'];
  Case tecla of
    'D' : EntraArquivo;
    'T' : EntraTeclado;
  End;
  If picosenc then
    For i := 1 to np do
      mz[i] := Round(Int(massareal[i] * 2.0 + 0.5) * 5.0); (* 10.0 / 2.0 *)
    End;
  Procedure SalvaArquivo;
  var arqres : Text;
  i : Integer;
  Begin
    WriteLn;
    WriteLn('Qual o nome do arquivo?');
    ReadLn(nome);
    WriteLn('Confirma? (S/N)');
    Read(kbd, tecla);

```

```

If UpCase(tecla) <> 'S' then Exit;
Assign(arqres,nome);
(*!-
Rewrite(arqres);
ESresultado := iDresult;
(*!+
Case ESresultado of
  $F0 : begin
    Close(arqres);
    WriteLn('Disco com problemas ou cheio');
    Aviso('erro');
    Delay(1000);
  end;
  0 : begin end;
  else
    begin
      Close(arqres);
      Aviso('erro');
    end;
  end;
  If ESresultado <> 0 then Exit;
  WriteLn(arqres,nomegc);
  WriteLn(arqres,nscan);
  WriteLn(arqres,teaporet);
  WriteLn(arqres,linaxar);
  For i := 1 to np do WriteLn(arqres,massareal[i] * 16.2, ' ',area[i]);
  Close(arqres);
End;

Procedure ImprimeTabela;
const topo = ' m/z      m/z Intensidade(em z) ';
;

Procedure ListaV;
Begin
  ClrScr;
  WriteLn('Relacao dos picos');
  WriteLn;
  WriteLn('dado ',topo);
  Window(1,4,80,25);
  For i := 1 to np do
    begin
      WriteLn(i:3, ' ',massareal[i] * 16.2, ' ',mz[i] * 10.0 * 16.2);
      If (i mod 15 = 0) and (i < np) then
        begin
          WriteLn('Tecla para continuar ....');
          Repeat Until KeyPressed;
          GoToXY(1,WhereY - i);
        end;
      end;
    end;
  WriteLn('Tecla para continuar ....');
  Repeat Until KeyPressed;
End;

Procedure ListaI;
var marg : String(20);

```

linha, coluna, desloc, d : Integer;

Procedure Cabecalho(var linha : Integer);

var i : Integer;

Begin

 linha := 1;

 For i := 1 to 4 do WriteIn(lst);

 WriteIn(lst,SI,SQ, ' Programa GRAF (CLL - 'versao,')',CR);

 WriteIn(lst,SI,SQ, ' Programa GRAF (CLL - 'versao,')');

 WriteIn(lst);

 WriteIn(lst, 'Arquivo: ', nome);

 WriteIn(lst);

 linha := linha + 8;

End;

Begin
 arg := SJ + ' ' ; { compactado + 15 espacos em branco }

 Cabecalho(linha);

 WriteIn(lst, arg, 'Arquivo do cromatograma: ', nomegc);

 WriteIn(lst, arg, 'Numero do scan: ', nscan);

 WriteIn(lst, arg, 'Tempo de retencao: ', temporet :0:3, ' min');

 WriteIn(lst, arg, 'Limiar de deteccao: ', limiar / 100.0 :0:2, ' %');

 WriteIn(lst);

 WriteIn(lst, arg, ' Relacao dos picos');

 WriteIn(lst);

 WriteIn(lst, arg, topo, topo, topo);

 linha := linha + 3;

 desloc := np div 3;

 If np mod 3 < 0 then desloc := desloc + 1;

 For i := 1 to desloc do

 begin

 Write(lst, arg);

 For coluna := 0 to 2 do

 begin

 d := coluna * desloc + i;

 If d <= np then Write(lst, massareal[d] :0:2, ' ', az[d]/10.0 :5:1, ' ',

 area[d]/100.0 :0:2, ' ');

 end;

 WriteIn(lst);

 linha := linha + 1;

 If (linha mod 51) = 0 then

 begin

 Write(lst, FF);

 Cabecalho(linha);

 WriteIn(lst, arg, ' Relacao dos picos (continuacao)');

 WriteIn(lst);

 WriteIn(lst, arg, topo, topo, topo);

 linha := linha + 3;

 end;

 end;

 If (linha mod 51) < 0 then Write(lst, FF, DC2);

End;

Procedure Corrigir;

var altura : Real;

j : Integer;

Begin

 ClrScr;

 WriteIn('Qual o numero do dado?');

 ReadIn(i);

 WriteIn('m/z real = ', massareal[i] :0:2, ' d');

 WriteIn('m/z aproximada = ', az[i]/10.0 :5:1, ' d');

 WriteIn('intensidade = ', area[i]/100.0 :0:2, ' %');

 WriteIn;

 WriteRes('Corrigir m/z Real, Intensidade ou Deletar?');

 Repeat

 Read(kbd, tecla); tecla := UpCase(tecla);

 Until tecla in ['I', 'R', 'D'];

 WriteIn;

 Case tecla of

 'R': begin

 WriteIn('Qual o novo valor de m/z? (em d)');

 ReadIn(massareal[i]);

 az[i] := Round(Int(massareal[i] * 2.0 + 0.5) * 5.0); { * 10.0 / 2.0 }

 end;

 'I': begin

 altura := area[i] / 100.0;

 WriteIn('Qual o novo valor de intensidade? (em %)');

 ReadIn(altura);

 If altura > 100.0 then altura := 100.0;

 area[i] := Round(100.0 * altura);

 end;

 'D': begin

 For j := i + 1 to np do

 begin

 area[j - 1] := area[j];

 massareal[j - 1] := massareal[j];

 az[j - 1] := az[j];

 end;

 np := np - 1;

 end;

 end;

End;

Begin

 ClrScr;

 WriteRes('TABELA');

 WriteIn;

 WriteRes('Corrigir um dado');

 WriteRes('Listar na Impressora');

 WriteRes('Listar no Video');

 WriteRes('ESC para voltar ao menu anterior');

 Repeat

 Read(kbd, tecla); tecla := UpCase(tecla);

 Until tecla in ['I', 'V', 'C', 'ESC'];

 Case tecla of

 'C': Corrigir;

 'I': Listar;

 'V': ListarV;

 end;

End;

```
Procedure Grafica;
```

```
var cursor, inicio,
```

```
    inicioespec,
```

```
    fim, fim espec,
```

```
    inreg, fiareg,
```

```
    picobase, j : Integer;
```

```
fy, fx : Real;
```

```
naoacha : Array[1..MAXPIC] of Boolean;
```

```
fundo, branco, massa : Array[1..30] of Byte;
```

```
Function Y(sinal : Integer) : Integer;
```

```
Begin
```

```
    y := Round(fy + (picobase - sinal)) + 15;
```

```
End;
```

```
Function X(sinal : Integer) : Integer;
```

```
Begin
```

```
    x := Round(fx + (sinal - inicio)) + 36;
```

```
End;
```

```
Procedure Plota;
```

```
Begin
```

```
    Plot(x(cursor), 155, 1 - GetDotColor(x(cursor), 155));
```

```
    Plot(x(cursor), 156, 1 - GetDotColor(x(cursor), 156));
```

```
    Plot(x(cursor), 157, 1 - GetDotColor(x(cursor), 157));
```

```
End;
```

```
Procedure PosCursor(desloc : Integer);
```

```
Begin
```

```
    Plota;
```

```
    cursor := cursor + desloc;
```

```
    If cursor < 0 then cursor := 0; (entre 0 e 600 d)
```

```
    If cursor > 600 then cursor := 600;
```

```
    Plota;
```

```
    GoToXY(31,21); Write(cursor div 10 #3);
```

```
End;
```

```
Procedure Atualiza(eixox : Boolean);
```

```
var incua : Boolean;
```

```
    pilha : Integer;
```

```
Begin
```

```
    GoToXY(1,3); Window(1,3,80,24);
```

```
    If eixox then GraphWindow(0,16,639,183) else GraphWindow(0,16,639,171);
```

```
    ClearScreen;
```

```
    GetPic(Branco, 0, 0, 23, 7); (branco para numeracao de picos)
```

```
    GraphWindow(0,16,639,183);
```

```
    If inicio > fim then
```

```
        begin
```

```
            pilha := inicio;
```

```
            inicio := fim;
```

```
            fim := pilha;
```

```
        end;
```

```
    If inicio >= fim espec then inicio := fim espec - 1500;
```

```
    If fim := inicioespec then fim := inicioespec + 1500;
```

```
    If inicio = fim then inicio := inicioespec;
```

```
    fx := 600.0 / (fim - inicio);
```

```
    fy := 140.0 / picobase;
```

```
    Quadra(35,15,637,155,1);
```

```
    If eixox then (faz o eixo horizontal)
```

```
    begin
```

```
        If fx < 0.2 then incua := FALSE else incua := TRUE;
```

```
        For i := inicio to fim do
```

```
            If (i mod 100) = 0 then Draw(x(i), 156, x(i), 158, 1)
```

```
                else If incua then
```

```
                    If (i mod 10) = 0 then Plot(x(i), 157, 1);
```

```
                    GoToXY(4,21); Write(Round(inicio/10) #3);
```

```
                    GoToXY(78,21); Write(Round(fim/10) #3);
```

```
                    GoToXY(15,21); Write('cursor em #/2 = ');
```

```
                end;
```

```
        For i := 1 to 10 do (grada o eixo vertical)
```

```
            If 1000 # i < picobase then
```

```
                begin
```

```
                    j := y(1000 # i);
```

```
                    Draw(33, j, 34, j, 1);
```

```
                    Plot(35, j, 0);
```

```
                    Plot(637, j, 0);
```

```
                end;
```

```
            GoToXY(1,2); Write(picobase/100 #3:0,'x');
```

```
            inreg := 1;
```

```
            fiareg := 0;
```

```
            While (m[inreg] < inicio) do inreg := inreg + 1;
```

```
            While (m[fiareg] > fim) do fiareg := fiareg - 1;
```

```
            For i := inreg to fiareg do
```

```
                Draw(x(m[i]), y(0), x(m[i]), y(m[i]), 1); (desenha o espectro)
```

```
            Plotat; (por causa do cursor)
```

```
            PosCursor(0);
```

```
        End;
```

```
Procedure NumPico(pico : Integer);
```

```
var xp, xc, yp, yc,
```

```
    xt, yt,
```

```
    linha, coluna,
```

```
    ni, nf : Integer;
```

```
    iguais : Boolean;
```

```
Begin
```

```
    xp := x(m[pico]);
```

```
    yp := y(area[pico]) - 4; (-4 para ficar um pouco distanciado da ponta)
```

```
    xc := xp - 12; (canto superior esquerdo)
```

```
    yc := yp - 12;
```

```
    GoToXY(1,10); (transforma caracteres em janela grafica)
```

```
    GetPic(fundo, 0, 72, 23, 79);
```

```
    Write(m[pico] div 10 #3);
```

```
    GetPic(massa, 0, 72, 23, 79);
```

```
    PutPic(fundo, 0, 79);
```

```
    For coluna := 0 to 2 do
```

```
        begin
```

```
            If coluna = 2 then coluna := -1;
```

```
            xt := xc + 10 # coluna;
```

```

linha := 0;
iguais := FALSE;
While (linha <= 20) and not(iguais) do
begin
  yt := yc - 2 * linha;
  GetPic(fundo, xt, yt,
        xt + 23, yt + 7);
  iguais := TRUE; { testa se o fundo esta em branco }
  For j := 1 to 30 do if fundo[j] <> branco[j] then iguais := FALSE;
  If yt < 0 then
  begin
    yt := 0;
    iguais := TRUE;
  end;
  If iguais then
  begin
    Drawxp, yp, xt + 12, yt + 4, 1; { linha de conexao }
    Plot(xp - 1, yp - 1, 1);
    Plot(xp + 1, yp - 1, 1);
    PutPic(massa, xt, yt + 7); { imprime m/z do pico }
    { nao nuera picos proximos no desenho }
    m1 := inicio + Round((xt - 36)/fx); { saiu da function x }
    m2 := inicio + Round((xt - 12)/fx);
    For j := inreg to fiareg do
      If (m2[j] > m1) and (m2[j] < m1) then naonua[j] := FALSE;
    end;
    linha := linha + 1;
  end;
  If iguais then
  begin
    For i := inreg to fiareg do { nao nuera picos de m/z distantes <= 4 d }
      If Abs(m2[pico] - m2[i]) <= 40 then naonua[i] := FALSE;
    end
    else naonua[pico] := FALSE;
    If iguais then Exit;
  end;
end;

Procedure Nuera;
var pb : Integer;
tudonua : Boolean;
Begin
  For i := inreg to fiareg do {nao nuera picos coa dupla carga ou estourados}
    If ((m2[i] mod 10) = 0) and (area[i] <= picobase)
    then naonua[i] := TRUE else naonua[i] := FALSE;
  tudonua := FALSE;
  Repeat
    pb := inreg - 1;
  Until naonua[pb] or (pb = fiareg);
  If pb = fiareg then
    If naonua[pb] then Nuapico(pb) else Exit;
  For i := pb to fiareg do
    If naonua[i] and (area[i] > area[pb]) then pb := i;

```

```

Nuapico(pb);
Until tudonua;
End;

Begin
  inicioespec := (m2[1] div 100) * 100;
  fim espec := (m2[mn] div 100) * 100 + 100;
  inicio := inicioespec;
  fim := fim espec;
  cursor := (fim - inicio) div 2;
  picobase := 10000;
  H.Res; ClearScreen;
  HikesColor(15);
  GoToXY(1,1);
  Write('Cromatograma: ', nomegc, ' Tempo de retencao: ', teopret:0:3, ' min');
  Quadrol(-1,9,640,190,1);
  Quadrol(-1,11,640,188,1);
  GoToXY(1,25);
  Write('1-inicio F2-fim F3-ampl. F4-reduz F5-nua. F6-atual.');
```

```

Print;
GoToXY(1,25);
Write(F1-inicio F2-fim F3-apl. F4-reduz F5-num. F6-atual.);
Write(' F7-impr. F8-sai');
GoToXY(1,3); Window(1,3,80,24);
GoToXY(15,21); Write('cursor em x/z = ');
Plot; { reaparece o cursor }
PostCursor(0);
end;

F10: begin end;
DIR : PosCursor(10); { incrementa 1 d }
ESQ : PosCursor(-10); { decrementa 1 d }
CDIR : PosCursor(100); { incrementa 10 d }
CESQ : PosCursor(-100); { decrementa 10 d }
HOME : PosCursor(inicio - cursor); { move para o inicio da janela }
YFIM : PosCursor(fim - cursor); { move para o fim da janela }
CHOME : begin
    inicio := inicioespec;
    cursor := (fim - inicio) div 2;
    Atualiza(TRUE);
end;

CFIM : begin
    fim := fimdespec;
    cursor := (fim - inicio) div 2;
    Atualiza(TRUE);
end;

PGUP : begin
    If cursor = fim then cursor := inicio;
    inicio := cursor - 1500; { janela de 150 d }
    If inicio < inicioespec then inicio := inicioespec;
    fim := inicio + 1500;
    Atualiza(TRUE);
    PosCursor(inicio - cursor);
end;

PGDN : begin
    If cursor = inicio then cursor := fim;
    fim := cursor + 1500;
    If fim > fimdespec then fim := fimdespec;
    inicio := fim - 1500;
    Atualiza(TRUE);
    PosCursor(fim - cursor);
end;

else Aviso('invalido');
end;
end
else
begin
    tecla := ESC;
    Aviso('invalido');
end;
until tecla = F10;
Window(1,1,80,25);
End;

```

BEGIN

```

EntraEspectro;
Repeat
    Repeat
        Menu;
        Read(kbd, opcao);
        opcao := UpCase(opcao);
        Case opcao of
            'E' : EntraEspectro;
            'G' : If picosenc then Grafica;
            'S' : If picosenc then SalvaArquivo;
            'I' : If picosenc then ImprimeLabela;
            'F' : Aviso('cuidado');
            else Aviso('invalido');
        end;
    until (opcao = 'F');
    ClrScr;
    WriteRes('confirma saida? (S/N)');
    Read(kbd, opcao);
    until (UpCase(opcao) = 'S');
END.

```

Programa Analise_de_Cluster;

```

const MAXPIC = 20;
MAXPRE = 5;
MAXLISTA = 10;
cation = '+';
cationrad = '-';
ESC = #27;
versao = '4.4';

```

```

type TipoElemento = (Carbono, Hidro, Nitro, Oxi, Silicio,
Fosforo, Enxofre, Fluor, Cloro, Bromo, Iodo);

```

```
Todos = Carbono..Iodo;
```

```

TipoAbd = Array[1..3] of Real;
TipoAbdElementos = Array[Todos] of TipoAbd;
TipoNuqPicos = Array[Todos] of Integer;
TipoMCont = Array[Todos] of Real;
String2 = String[2];
String255 = String[255];
TipoV20 = Array[0..MAXPIC] of Real;
TipoV201 = Array[0..MAXPIC] of Integer;
TipoContribucao = Array[Todos] of TipoV20;
TipoMat5x20 = Array[0..MAXPRE] of TipoV20;
TipoMat5x1 = Array[0..MAXPRE] of Integer;
TipoPreCalc = Array[Todos] of TipoMat5x20;
TipoPreCalcN = Array[Todos] of TipoMat5x1;
TipoSimbolo = Array[Todos] of String2;
TipoReg = Record

```

```

    delta : Real;
    nuq : TipoNuqPicos;
    ion : String2;
    gi : Integer;
end;

```

```
TipoLista = Array[1..MAXLISTA] of TipoReg;
```

```

{ Abd contem a abundancia dos isotopos de cada elemento e
Iso contem a massa de cada um dos isotopos.

```

```

Dados compilados de
J.H.Beynon, R.A.Saunders and A.E.Williams, "The Mass Spectra
of Organic Molecules", Elsevier, Amsterdam, 1968, Apendice 2.
Foram ignorados os isotopos deuterio e oxigenio 17 para simplificar
o programa. }

```

```

const Abd : TipoAbdElementos = ((0.9889, 0.0111, 0.0 ), ( C )
(1.0 , 0.0 , 0.0 ), ( H )
(0.9964, 0.0036, 0.0 ), ( N )
(0.9976, 0.0020, 0.0 ), ( O )
(0.9217, 0.0471, 0.0312), ( Si )
(1.0 , 0.0 , 0.0 ), ( P )
(0.9503, 0.0076, 0.0420), ( S )
(1.0 , 0.0 , 0.0 ), ( F )
(0.7577, 0.2423, 0.0 ), ( Cl )
(0.5052, 0.4948, 0.0 ), ( Br )
(1.0 , 0.0 , 0.0 ));

```

```

Iso : TipoAbdElementos = (( 12.0000, 13.0034, 0.0 ), ( C )
( 1.0078, 0.0 , 0.0 ), ( H )
( 14.0031, 15.0001, 0.0 ), ( N )
( 15.9949, 17.9992, 0.0 ), ( O )
( 27.9769, 28.9765, 29.9738), ( Si )
( 30.9738, 0.0 , 0.0 ), ( P )
( 31.9721, 32.9715, 33.9679), ( S )
( 18.9984, 0.0 , 0.0 ), ( F )
( 34.9689, 36.9659, 0.0 ), ( Cl )
( 78.9184, 80.9164, 0.0 ), ( Br )
(126.9047, 0.0 , 0.0 ));

```

```

Simbolo : TipoSimbolo = ('C', 'H', 'N', 'O', 'Si',
'P', 'S', 'F', 'Cl', 'Br', 'I');

```

```
var opcao, tecla : Char;
```

```
ultiad,
```

```
mi, i,
```

```
: Integer;
```

```
: Real;
```

```
: Real;
```

```
: Real;
```

```
: Real;
```

```
: Real;
```

```
: Real;
```

```
: Real;
```

```
: Real;
```

```
: Real;
```

```
: Real;
```

```
: Real;
```

```
: Real;
```

```
: Real;
```

```
: Real;
```

```
: Real;
```

```
: Real;
```

```
: Real;
```

```
: Real;
```

```
: Real;
```

```
: Real;
```

```
: Real;
```

```
: Real;
```

```
: Real;
```

```
: Real;
```

```
: Real;
```

```
: Real;
```

```
: Real;
```

```
: Real;
```

```
: Real;
```

```
: Real;
```

```
: Real;
```

```
: Real;
```

```
: Real;
```

```
: Real;
```

```
: Real;
```

```
: Real;
```

```
: Real;
```

```
: Real;
```

```

janormal,
janelal      : Boolean;

```

```

{$R+}

```

```

Procedure Tab(x : Integer);

```

```

Begin

```

```

  GoToXY(x, WhereY);

```

```

End;

```

```

Procedure WriteResis : String255;

```

```

var i : Integer;

```

```

Begin

```

```

  For i := 1 to length(s) do

```

```

    begin

```

```

      If s[i] in ['A'..'Z'] then TextColor(15) else TextColor(7);

```

```

      Write(s[i]);

```

```

    end;

```

```

  TextColor(7);

```

```

  Writeln;

```

```

End;

```

```

Function Eleva(x : Real; a : Integer) : Real; { faz "x" elevado a "a" }

```

```

var i : Integer;

```

```

  r : Real;

```

```

Begin

```

```

  r := 1.0;

```

```

  For i := 1 to a do r := r * x;

```

```

  ElevaR := r;

```

```

End;

```

```

Function Fat(x : Integer) : Real; { faz o fatorial de "x" entre 0 e 33 }

```

```

var i : Integer;

```

```

  r : Real;

```

```

Begin

```

```

  r := 1.0;

```

```

  If x <= 33 then For i := x downto 2 do r := r * i

```

```

  else

```

```

    begin

```

```

      Sound(1000);

```

```

      WriteLn(x, ' muito grande para fatorial. ');

```

```

      NoSound;

```

```

    end;

```

```

  Fat := r;

```

```

End;

```

```

Procedure EntraCluster;

```

```

var nome : String[14];

```

```

  arq : Text;

```

```

Begin

```

```

  For i := 1 to MAXPIC do

```

```

    begin

```

```

      mz[i] := 0;

```

```

      orig[i] := 0.0;

```

```

      altmax[i] := 0.0;

```

```

      altmin[i] := 0.0;

```

```

    end;

```

```

  Crlfcr;

```

```

  Writeln('ENTRA OS PICS DO CLUSTER');

```

```

  Writeln;

```

```

  WriteRes('Disquete');

```

```

  WriteRes('Teclado');

```

```

  WriteRes('ESC abandonado');

```

```

  Writeln;

```

```

  Repeat

```

```

    Read(kbd, tecla);

```

```

    tecla := UpCase(tecla);

```

```

    Until tecla in ['T', 'D', 'ESC'];

```

```

    If tecla = ESC then Exit;

```

```

  Case tecla of

```

```

    'D' : begin

```

```

      Writeln('Qual o nome do arquivo?');

```

```

      Readln(nome);

```

```

      Assign(arq, nome);

```

```

      Reset(arq);

```

```

      Readln(arq, np);

```

```

      Readln(arq, picobase);

```

```

      Readln(arq, mz[1]);

```

```

      For i := 2 to MAXPIC do mz[i] := mz[i - 1] + 1;

```

```

      For i := 1 to np do Readln(arq, orig[i]);

```

```

      Close(arq);

```

```

    end;

```

```

  'T' : begin

```

```

    Writeln('Qual a m/z do primeiro pico?');

```

```

    Readln(mz[1]);

```

```

    Writeln('Qual a m/z do ultimo pico?');

```

```

    Readln(mz[2]);

```

```

    np := mz[2] - mz[1] + 1;

```

```

    For i := 2 to MAXPIC do mz[i] := mz[i - 1] + 1;

```

```

    For i := 1 to np do

```

```

      begin

```

```

        Writeln('pico', i);

```

```

        Writeln('      m/z = ', mz[i], ' d');

```

```

        Write ('intensidade = '); Readln(orig[i]);

```

```

        Writeln;

```

```

      end;

```

```

      picobase := orig[i];

```

```

      For i := 2 to np do

```

```

        If orig[i] > picobase then picobase := orig[i];

```

```

        Write('Intensidade do pico base = ', picobase :11:5, ' --> ');

```

```

        Readln(picobase);

```

```

        If picobase = 0.0 then picobase := 1.0;

```

```

      end;

```

```

    end;

```

```

    If np = 1 then { Acrescenta mais um pico para dar liberdade. }

```

```

      begin

```

```

        np := 2;

```

```

        mz[2] := mz[1] + 1;

```

```

        orig[2] := 0.0;

```

```

      end;

```

```

min := alt[i];
For i := 2 to np do
  If alt[i] < min then min := alt[i];
soma := 0.0;
For i := 1 to np do soma := soma + sqrt(alt[i]);
d2max := 1.0 - 2.0 * min + soma;
End;

Procedure Simples(el : TipoElemento); ( Para C, N e G. )
Begin
  cal[el,1] := Elevar(Abd[el,1], cont[el]);
  cal[el,2] := Abd[el,2] * Elevar(Abd[el,1], cont[el] - 1) * cont[el];
  cal[el,3] := 1 - cal[el,1] - cal[el,2];
  For i := 0 to 2 do
    caz[el,i+1] := Iso[el,i] * (cont[el] - i) + Iso[el,2] * i;
  If cal[el,3] > 0.0 then ni[el] := 3 else
    If cal[el,2] > 0.0 then ni[el] := 2 else ni[el] := 1;
  End;
End;

Procedure Halo(el : TipoElemento); ( Para Cl e Br. )
var f : Real;
Begin
  ni[el] := cont[el] + 1;
  f := Fat(cont[el]);
  For i := 0 to cont[el] do
    Begin
      p := Elevar(Abd[el,i], cont[el] - i) * Elevar(Abd[el,2], i);
      cal[el,i+1] := p * f / (Fat(cont[el] - i) * Fat(i));
      caz[el,i+1] := Iso[el,i] * (cont[el] - i) + Iso[el,2] * i;
    end;
  End;

Procedure Completa(el : TipoElemento);
var f : Real;
  q : Integer;
Begin
  f := Fat(cont[el]);
  q := 0;
  For i := 0 to cont[el] do
    For j := 0 to (cont[el] - i) do
      Begin
        k := cont[el] - i - j;
        p := Elevar(Abd[el,i],k) * Elevar(Abd[el,2],j) * Elevar(Abd[el,3],i);
        q := q + 1;
        If q > MAXPIC then Exit;
        cal[el,q] := p * f / (Fat(i) * Fat(j) * Fat(k));
        caz[el,q] := Iso[el,i] * k + Iso[el,2] * j + Iso[el,3] * i;
      end;
      ni[el] := q;
    end;
  End;

Procedure FazPreCalc;
var i, j : Integer;
Begin
  ClrScr;
  WriteLn('Pico base = ', picobase :11:5);
  WriteLn('  n/z  original normalizado');
  For i := 1 to np do
    Begin
      Write(az[i] :6, ' ', orig[i] :11:5);
      orig[i] := orig[i] / picobase * 100.0;
      WriteLn(orig[i] :10:2, '%');
    end;
  picobase := 100.0;
  Repeat Until keypressed;
  teadados := TRUE;
  janormal := FALSE;
  jaanal := FALSE;
  fatorsub := 0.0; ( Ainda nao foi subtraido nenhuma formula. )
End;

Procedure CondicoesIniciais;
Begin
  errorel := 10.0; ( 10% da intensidade do pico )
  erroab := 0.2; ( ou 0.2 absoluto. )
  erroaz := 0.3; ( 0.3 d para baixa resolucao )
  ni[Hidro] := 1; ( Monoisotopicos )
  ni[Fluor] := 1;
  ni[Fosforo] := 1;
  ni[Clodo] := 1;
  ni[Hidro] := 1;
  ni[Fluor] := 1;
  ni[Fosforo] := 1;
  ni[Clodo] := 1;
  cal[Hidro,1] := 1.0;
  cal[Fluor,1] := 1.0;
  cal[Fosforo,1] := 1.0;
  cal[Clodo,1] := 1.0;
  jaanal := FALSE;
End;

Procedure EntraCondicoes;
Begin
  ClrScr;
  Writeln('CONDICOES');
  WriteLn;
  Write('Intensidade: erro relativo = ,errorel :3:1, % --> ');
  ReadLn(errorel);
  Write('erro absoluto = ,erroab :3:1, --> ');
  ReadLn(erroab);
  Write('n/z: erro = ,erroaz :3:2, d --> ');
  ReadLn(erroaz);
  janormal := FALSE;
End;

Procedure Calcd2max;
( Determina a maior distancia entre dois clusters no hiperespaco. )
var min, soma : Real;
Begin
  ClrScr;

```

```

If cont[el] <= MAXPRE then
begin
  cal[el] := pcal[el,cont[el]];
  cex[el] := pcex[el,cont[el]];
  n[el] := pcn[el,cont[el]];
end
else
begin
  Case el of
    Carbono, Nitro, Oxi : Samples[el];
    Cloro, Bromo : Halo[el];
    Silicio, Enxofre : Completa[el];
  end;
  { Samples(Carbono);
  { Samples(Nitro);
  { Samples(Oxi);
  { Halo(Cloro);
  { Halo(Bromo);
  { Completa(Silicio);
  { Completa(Enxofre);
  For i := 1 to MAXPIC do altc[i] := 0.0; { Limpa o vetor resultado }
  Junta(Carbono);
End;

```

Procedure AnalisaCluster(Analisaemnos1 : Boolean);

```

Procedure Normaliza;
var intmin,
    intmax : TipoV20;
    somamax,
    somamin,
    erro : Real;
Begin
  { determina os erros das intensidades }
  somamax := 0.0;
  somamin := 0.0;
  For i := 1 to np do
    begin
      erro := errorel + orig[i] / 100.0;
      If erro < erroab then erro := erroab;
      intmax[i] := orig[i] + erro;
      intmin[i] := orig[i] - erro;
      If intmin[i] < 0.0 then intmin[i] := 0.0;
      somamax := somamax + intmax[i];
      somamin := somamin + intmin[i];
    end;
  { Calcula a intensidade maxima e minima de cada pico. }
  For i := 1 to np do
    begin
      altmin[i] := intmin[i] / (somamax - intmin[i] + intmin[i]);
      altmax[i] := intmax[i] / (somamax - intmin[i] + intmax[i]);
    end;
  { Normaliza em relacao a soma total. }
  somamax := 0.0;
  For i := 1 to np do somamax := somamax + orig[i];
  For i := 1 to np do alt[i] := orig[i] / somamax;

```

```

WriteLn('PRE-CALCULO');
WriteLn;
Write('Preparando a tabela de abundancia isotopica para ');
WriteLn(MAXPRE, ' atomos de cada elemento...');
For el := Carbono to Iodo do
  For i := 0 to MAXPRE do
    begin
      cont[el] := i;
      Completa[el];
      pcal[el,i] := cal[el];
      pcex[el,i] := cex[el];
      j := n[el];
      While (cal[el,j] = 0.0) do j := j - 1;
      pcn[el,i] := j;
    end;
End;

Procedure MontaCluster(var cont : TipoNuaPicos;
  var altc : TipoV20;
  var nzi : Integer);
{ O vetor cont determina o numero de atomos de cada elemento
e o cluster calculado retorna em altc, com m/z determinado
pelo mesmo vetor nzi. }

```

Procedure CalcPicos;

```

Begin
  p := 1.0;
  m := 0.0;
  For el := Carbono to Iodo do
    begin
      p := p * cal[el,i[el]];
      m := m + cex[el,i[el]];
    end;
  i := Round(m - nzi) + 1; { encontra o indice do vetor resultado }
  If i <= MAXPIC then altc[i] := altc[i] + p;
End;

```

Procedure Junta[el] : TipoElemento;

```

Begin
  If el in [Hidro, Fluor, Fosforo] then elan := Succ[el];
  i[elan] := i;
  Repeat
    If elan < Bromo then Junta(Succ[elan]) else CalcPicos;
    i[elan] := i[elan] + 1;
  Until (i[elan] > n[elan]);
End;

Begin { MontaCluster }
  cex[Hidro,i] := Iso[Hidro,i] * cont[Hidro];
  cex[Fluor,i] := Iso[Fluor,i] * cont[Fluor];
  cex[Fosforo,i] := Iso[Fosforo,i] * cont[Fosforo];
  cex[Iodo,i] := Iso[Iodo,i] * cont[Iodo];
  { As variaveis ii e ni dos elementos acima ja foram definidas
  como 1 nas condicoes iniciais. }
  For el := Carbono to Bromo do

```

```

( Terceira consideracao : O numero de Br depende da m/z do ultimo pico. )
mi := (mz[m] - m[1]) div 2;
If max[Bromo] > mi then max[Bromo] := mi; { prevalece o menor }
( Quarta consideracao : Verifica se ha a possibilidade de pelo menos
um atomo dos elementos Si, S, Cl e Br. )
If altmax[3] < Add(Silicio,3) then max[Silicio] := 0;
If altmax[3] < Add(Enxofre,3) then max[Enxofre] := 0;
If altmax[2] < Add(Cloro,2) then max[Cloro] := 0;
If altmax[3] < Add(Bromo,2) then max[Bromo] := 0;
( Compara os numeros maximos com os minimos. )
For el := Carbono to Iodo do
If max[el] < min[el] then max[el] := min[el];
ClrScr;
WriteRes('NUMERO MAXIMO DE CADA ELEMENTO');
WriteLn;
For el := Carbono to Iodo do
begin
TextColor(14);
Write(simbolo[el]);
TextColor(7);
If length(simbolo[el]) = 1 then Write(' ');
Write(' = ', max[el] : 3, ' --> ');
posicao[el] := WhereX;
Repeat
maxtemp := max[el];
Read(maxtemp);
If maxtemp < min[el] then ( Garante a coerencia entre min e max. )
begin
Write(7, ' minimo estipulado de ', min[el]);
delay(3000);
Tab(posicao[el]); ClrEol;
end
else
begin
max[el] := maxtemp;
WriteLn;
end;
Until max[el] >= min[el];
end;
End;

```

```

Janormal := TRUE;
Calcd2max;
End;

Procedure NuaMax;
var posicao[255], maxtemp : Integer;
soma : Real;
coerente : Boolean;
Begin
ClrScr;
WriteRes('TIPO DE IONS A CONSIDERAR');
WriteRes(' - Cation e cation radical');
WriteRes(' - somente cation Radicalar');
Repeat
Read(kbd, tecla);
tecla := UpCase(tecla);
Until tecla in ['C', 'R'];
If tecla = 'R' then socatrad := TRUE else socatrad := FALSE;
Repeat
ClrScr;
WriteRes('NUMERO MINIMO DE CADA ELEMENTO');
WriteLn;
For el := Carbono to Iodo do min[el] := 0;
If analise then min[Hidro] := 1 else min[Hidro] := 0; ( Analise de M - 1. )
For el := Carbono to Iodo do
begin
TextColor(14);
Write(simbolo[el]);
TextColor(7);
If length(simbolo[el]) = 1 then Write(' ');
Write(' = ', min[el] : 3, ' --> ');
ReadLn(min[el]);
end;
soma := 0.0;
For el := Carbono to Iodo do soma := soma + min[el] * Iso[el,1];
If soma > m[1] then
begin
coerente := FALSE;
WriteLn(7, 'Composicao minima supera m/z do cluster');
delay(3000);
end
else coerente := TRUE;
Until coerente;
( Primeira consideracao: Limita os heteroatomos pela m/z do primeiro pico
e pela presenca de carbono (pelo menos 1/4 para cada heteroatomo). )
max[Carbono] := m[1] div 12;
For el := Nitro to Enxofre do
begin
max[el] := Trunc(m[1] / (Iso[el,1] + 3.0 * min[Carbono]));
end;
( Segunda consideracao : O numero de hidrogenio e halogenio e' maximo para
cadeia saturada. )
max[Hidro] := (m[1] + 12) div 7;
For el := Fluor to Iodo do
max[el] := (m[1] + 12) div Round(Iso[el,1] + 6.0);

```

```
Procedure Classifica;
```

```
var d : Real;
```

```
Begin
```

```
  d := 0.0;
```

```
  For i := 1 to np do d := d + sqrt(alt[i] - altc[i]);
```

```
  d := 100.0 * sqrt(d / n2max); ( % de diferenca )
```

```
  If d < lista[ultimo].delta then
```

```
    begin
```

```
      With r do
```

```
        begin
```

```
          delta := d;
```

```
          For el := Carbono to Iodo do num[el] := cont[el];
```

```
        end;
```

```
        i := i;
```

```
        While (r.delta > lista[i].delta) do i := i + 1; ( ve a posicao )
```

```
        For j := MAXLISTA - 1 downto i do lista[j + 1] := lista[j]; ( desloca )
```

```
        lista[i] := r; ( atualiza )
```

```
        If ultimo < MAXLISTA then ultimo := ultimo + 1;
```

```
        For el := Carbono to Iodo do
```

```
          If cont[el] > 0 then Write(cont[el] * 4) else Write(' ');
```

```
          WriteIn(lista[i].delta :12:3, '%');
```

```
        end;
```

```
      End;
```

```
Procedure Calc;
```

```
var gi, nh : Integer;
```

```
Begin
```

```
  If keypressed then read(tbd, tecla);
```

```
  If tecla = ESC then Exit;
```

```
  nzc := 0.0;
```

```
  For el := Nitro to Iodo do nzc := nzc + nzccont[el];
```

```
  n := nzc[i] - nzc;
```

```
  If n >= 0.0 then
```

```
    begin
```

```
      mi := Round(m);
```

```
      cont[Carbono] := mi div 12;
```

```
      cont[Hidro] := mi mod 12;
```

```
      ( nh e' o maximo numero de H (cadeia saturada). )
```

```
      nh := 2 * cont[Carbono] + cont[Nitro] + cont[Fosforo] + 2 * cont[Silicio]
```

```
      - cont[Fuor] - cont[Cloro] - cont[Bromo] - cont[Iodo] + 3;
```

```
      While (cont[Carbono] >= min[Carbono]) and (cont[Hidro] <= max[Hidro])
```

```
        and (cont[Hidro] <= nh) do
```

```
        begin
```

```
          If cont[Hidro] >= min[Hidro] then
```

```
            If cont[Carbono] <= max[Carbono] then
```

```
              begin
```

```
                n := nzc + Iso[Carbono, i] * cont[Carbono]
```

```
                + Iso[Hidro, i] * cont[Hidro]; ( calcula nzc do primeiro pico )
```

```
                If Abs(n - nzc[i]) < erroznz then
```

```
                  begin
```

```
                    gi := cont[Nitro] + cont[Fosforo] - cont[Hidro]
```

```
                    - cont[Fuor] - cont[Cloro] - cont[Bromo] - cont[Iodo];
```

```
                    {gi := gi/2 + (cont[Carbono] + cont[Silicio]) * i }
```

```
                    If (not socatrad) or (not Odd(gi) and socatrad) then
```

```
                      begin
```

```
MontaCluster(cont, altc, nzc[i]);
```

```
If ClusterOk then Classifica;
```

```
end;
```

```
end;
```

```
end;
```

```
  cont[Carbono] := cont[Carbono] - 1;
```

```
  cont[Hidro] := cont[Hidro] + 12;
```

```
  nh := 2 * cont[Carbono] + cont[Nitro] + cont[Fosforo] +
```

```
  2 * cont[Silicio] - cont[Cloro] - cont[Bromo] - cont[Iodo] + 2;
```

```
end;
```

```
end;
```

```
End;
```

```
Procedure Analisa(elan : TipoElemento);
```

```
Begin
```

```
  cont[elan] := min[elan];
```

```
  nzccont[elan] := min[elan] * Iso[elan, i];
```

```
  Repeat
```

```
    If elan <> Nitro then Analisa(Pred[elan]) else Calc;
```

```
    nzccont[elan] := nzccont[elan] + Iso[elan, i];
```

```
    cont[elan] := cont[elan] + 1;
```

```
  Until (cont[elan] > max[elan]);
```

```
End;
```

```
Procedure Tempo(var t : Real);
```

```
type
```

```
  regpack = record
```

```
    ax, bx, cx, dx, bp, si, di, ds, es, flags: Integer;
```

```
  end;
```

```
var regpack: regpack;
```

```
  hora, min, seg : Byte;
```

```
Begin
```

```
  regpack.ax := $2000;
```

```
  intr($21, regpack);
```

```
  with regpack do
```

```
    begin
```

```
      hora := cx shr 8;
```

```
      min := cx mod 256;
```

```
      seg := dx shr 8;
```

```
      t := seg + min * 60 + hora * 3600;
```

```
    end;
```

```
  End;
```

```
Begin ( AnalisaCluster )
```

```
  analm := analisamenosi;
```

```
  If not Janormal then Normaliza;
```

```
  For i := 1 to MAXLISTA do ( prepara a lista )
```

```
    With lista[i] do
```

```
      begin
```

```
        delta := 100.0;
```

```
        ion := 'xx';
```

```
      end;
```

```
      ultimo := 1; ( lista vazia )
```

```
    end;
```

```
  NumMax;
```

```
  p := 1.0; ( Avisa sobre o numero de combinacoes possiveis )
```

```

For el := Nitro to Iodo do p := p * (1.0 + (ax[el] - min[el]));
WriteLn('Número de combinações superior a ', p :10:0);
WriteRes('testa margem de erro? (S/N)');
Repeat
  Read(kbd, tecla);
  tecla := UpCase(tecla);
Until tecla in ['S', 'N'];
If tecla = 'S' then TestaMargem := TRUE else TestaMargem := FALSE;
WriteRes('confirma? (S/N)');
Repeat
  Read(kbd, tecla);
  tecla := UpCase(tecla);
Until tecla in ['S', 'N'];
If tecla = 'S' then
  begin
    Tempo(teapot);
    ClrScr;
    WriteRes('ANALISE DE CLUSTER');
    WriteLn;
    TextColor(14);
    For el := Carbono to Iodo do
      begin
        If length(Simbolo[el]) = 1 then Write(' ');
        Write(' ', Simbolo[el]);
      end;
    WriteLn('Diferença');
    For el := Carbono to Iodo do
      If min[el] > 0 then Write(min[el] :4);
      else Write(' ');
    WriteLn;
    For el := Carbono to Iodo do
      If ax[el] > 0 then Write(ax[el] :4);
      else Write(' ');
    WriteLn;
    TextColor(7);
    Window(1, WhereY, 80, 25);
    tecla := 'x';
    { Utilizada para interrosper os calculos. }
    { Efectua os calculos. }
    Analisa(todo);
    Window(1, 1, 80, 25);
    If lista[ultimo].ion = 'xx' then ultimo := ultimo - 1; { tira o lixo }
    For i := 1 to ultimo do { grau de insaturacao e tipo de ion }
      With lista[i] do
        begin
          gi := 2 * (num[Carbono] + num[Silicio]) - num[Hidro]
            - num[Fluor] - num[Cloro] - num[Bromo] - num[Iodo]
              + num[Nitro] + num[Fosforo] + 2;
          If Odd(gi) then ion := cation else ion := cationrad;
          gi := gi div 2;
        end;
        jaanal := TRUE;
        Tempo(teapot);
        Write(#7);
      end;
    end;
  end;
End;

```

```

Procedure Resultados;
const SI = #15; { para imprimir condensado }
      SD = #14; { para imprimir expandido }
      DC2 = #18; { desativa condensacao }
      CR = #13;
      FF = #12;
var texto, marg : String[20];
    identanal : String[255];
    obs : Boolean;
    tempogasto : Integer;

Procedure ImprimeComposicoes;
var i, branco : Integer;
Begin
  For i := 1 to ultimo do
    begin
      With lista[i] do
        begin
          Write(1st,marg,i:2,della :10:3,'X',qi :4);
          If num[Silicio] + num[Enxofre] + num[Fosforo] > 0
            then texto := 'f'
            else texto := ' '; { tomar cuidado com GI }
          Write(1st,texto,ion,' ');
          For el := Carbono to Iodo do
            If num[el] > 0 then
              begin
                Write(1st,Simbolo[el]);
                If num[el] > 1 then
                  For branco := 1 to Trunc((Ln(num[el]))/Ln(10)) + 1 do
                    Write(1st,' ');
                end;
                Write(1st,ESC,'A',#4); { Ativa entrelinhamento 4/72. }
                Write(1st,marg,' ');
                For el := Carbono to Iodo do
                  begin
                    If num[el] > 0 then
                      begin
                        For branco := 1 to length(Simbolo[el]) do Write(1st,' ');
                        If num[el] > 1 then Write(1st,num[el]);
                      end;
                    end;
                    Write(1st,ESC,'A',#12); { Ativa entrelinhamento 12/72. }
                  end;
                end;
                Write(1st,ESC,'2');
                { Ativa entrelinhamento 1/6. }
            end;
          end;
        end;
      end;
    end;
  end;
  Write(1st,ESC,'2');
  identanal := '';
  Write('Qual a identificacao da analise? '); ReadLn(identanal);
  WriteLn('tecle para imprimir...');
  Read(kbd, tecla);
  marg := SI + ' '; { 15 espaços em branco }

```

```

For i := 1 to 4 do WriteIn(1st); { margem superior }
Write(1st,SI,SO,' Relatório de Análise de Cluster',CR);
WriteIn(1st,SI,SO,' Relatório de Análise de Cluster');
WriteIn(1st);
WriteIn(1st,marg,'Programa AC versao ',versao,' (CLL)');
WriteIn(1st);
If identanal < '' then
begin
  WriteIn(1st,marg,'Identificacao: ',identanal);
  WriteIn(1st);
end;
Write(1st,marg,'Cluster');
If analal then WriteIn(1st,' apos subtrair M - H: ') else WriteIn(1st);
WriteIn(1st,marg,' m/z Int. Rel. Abundancia ');
For i := 1 to np do
  WriteIn(1st,marg,mz[i],orig[i]:11,2,'%',alt[i]:14,5);
WriteIn(1st);
WriteIn(1st,marg,'CONDICOES');
If TestaMargem then
  WriteIn(1st,marg,'Intensidade : Erro Relativo = ',errorel:4,1,
    '% e Erro Absoluto = ',erroab:3,1);
WriteIn(1st,marg,'Erro em m/z = ',errozm:3,2);
If fatoresub < 0.0 then
begin
  Write(1st,marg,'Foi subtraído do cluster original ');
  For el := Carbono to Iodo do
    If regform[el] > 0 then
      begin
        Write(1st,Símbolo[el]);
        If regform[el] > 1 then Write(1st,regform[el]);
      end;
  WriteIn(1st,' com fracao igual a ',fatoresub:8,5);
end;
Write(1st,marg,'Composicao: min. ');
For el := Carbono to Iodo do
  If min[el] > 0 then Write(1st,' ',min[el],Símbolo[el]);
WriteIn(1st);
Write(1st,marg,' max. ');
For el := Carbono to Iodo do
  If max[el] > 0 then Write(1st,' ',max[el],Símbolo[el]);
WriteIn(1st);
Write(1st,marg,'Tempo de análise: ');
tempogasto := Round(tempot - tempoi);
If tempogasto = 0 then WriteIn(1st,' < 1 segundo. ')
else if tempogasto = 1 then WriteIn(1st,' 1 segundo. ')
else WriteIn(1st,' tempogasto, segundos. ');
WriteIn(1st);
Write(1st,marg,'COMPOSICOES +');
If socatrad then WriteIn(1st) else WriteIn(1st,' E +');
WriteIn(1st);
WriteIn(1st,marg,' # Diferença SI Ion Composicao');
ImprimeComposicoes;
If obs then
begin
  WriteIn(1st);

```

```

WriteIn(1st,marg,' # Cuidado ao utilizar este grau de insaturacao');
WriteIn(1st,marg,' na elucidacao da estrutura molecular. ');
end;
WriteIn(1st,DC2,FF);
End;

Procedure ImprimeResIngles;
Begin
  Write(1st,ESC,'2'); { Ativa entrelinhamento 1/6. }
  Identanal := '';
  Write('Qual a identificacao da analise? '); ReadIn(Identanal);
  Write('Tecle para imprimir...');
  Read(1st,tecla);
  marg := SI + ' ' ( 15 espaços em branco )
  For i := 1 to 4 do WriteIn(1st); { margem superior }
  Write(1st,SI,SO,' Isotope Pattern Analysis',CR);
  WriteIn(1st,SI,SO,' Isotope Pattern Analysis');
  WriteIn(1st);
  WriteIn(1st,marg,'Program AC versao ',versao,' (CLL)');
  WriteIn(1st);
  If identanal < '' then
  begin
    Write(1st,marg,'Identificacao: ',identanal);
    WriteIn(1st);
  end;
  Write(1st,marg,'Cluster');
  If analal then WriteIn(1st,' after subtracted M - H') else WriteIn(1st);
  WriteIn(1st,marg,' m/z Rel. Int. Abundance');
  For i := 1 to np do
    WriteIn(1st,marg,mz[i],orig[i]:11,2,'%',alt[i]:14,5);
  WriteIn(1st);
  WriteIn(1st,marg,'CONDICOES');
  If TestaMargem then
    WriteIn(1st,marg,'Intensity : Relative Error = ',errorel:4,1,
      '% and Absolute Error = ',erroab:3,1);
  WriteIn(1st,marg,'m/z Error = ',errozm:3,2);
  If fatoresub < 0.0 then
  begin
    Write(1st,marg,'Cluster subtracted');
    For el := Carbono to Iodo do
      If regform[el] > 0 then
        begin
          Write(1st,Símbolo[el]);
          If regform[el] > 1 then Write(1st,regform[el]);
        end;
    WriteIn(1st,' Proportion ',fatoresub:8,5);
  end;
  Write(1st,marg,'Composition: min. ');
  For el := Carbono to Iodo do
    If min[el] > 0 then Write(1st,' ',min[el],Símbolo[el]);
  WriteIn(1st);
  Write(1st,marg,' max. ');
  For el := Carbono to Iodo do
    If max[el] > 0 then Write(1st,' ',max[el],Símbolo[el]);
  WriteIn(1st);

```

```

Write(1st,arg,'Tiae: ');
tempgasto := Round(tempof - tempol);
If tempgasto = 0 then Write(1st,' 1 s')
else if tempgasto = 1 then Write(1st,' 1 s')
    else Write(1st,' tempgasto, s');
Write(1st);
Write(1st,arg,'COMPOSITIONS +');
If socatrad then Write(1st) else Write(1st,' AND +');
Write(1st);
Write(1st,arg,' # Difference DB Ion Composition');
IapiraeComposicoes;
If obs then
begin
    Write(1st);
    Write(1st,arg,
        '* Rings plus Double Bond can be incorrect, attention to S1, S or P. ');
end;
Write(1st,DC2,FF);
End;

```

```

Begin { Resultados }
obs := FALSE;
ClrScr;
Writes('COMPOSICOES MAIS PROVAVEIS');
If socatrad then Write(1st,'Soamente cation radicalar');
Write(1st);

```

```

TextColor(15);
Write(1st,' # Diferenca BI Ion Composicao');
TextColor(7);
For i := 1 to ultimo do
begin
    With lista[i] do

```

```

begin
    Write(1st,delta:10:3,'X',gi:4);
    If nua[Silicio] + nua[Enxofre] + nua[Fosforo] > 0
    then
    begin
        Write(' ');
        obs := TRUE;
    end
end

```

```

else Write(' '); { tomar cuidado com BI }
TextColor(14);
Write(' ',ion,' ');
For el := Carbono to Iodo do
    If nua[el] > 0 then
    begin
        TextColor(14);
        Write(simbolo[el]);
        TextColor(7);
        If nua[el] > 1 then Write(nua[el]);
    end;
end;
Write(1st);
end;
end;

```

```

end;
end;
If obs then

```

```

begin
    Write(1st);
    Write(1st,'* Cuidado ao utilizar este grau de insaturacao');
    Write(1st,' na elucidacao da estrutura molecular. ');
end;
Write(1st);
Write(1st,'Tempo de analise: ');
tempgasto := Round(tempof - tempol);
If tempgasto = 0 then Write(1st,' 1 segundo. ')
else if tempgasto = 1 then Write(1st,' 1 segundo. ')
    else Write(1st,' tempgasto, segundos. ');
end;
Write(1st);

```

```

Writes('Iapirae em Portugues');
Writes('Iapirae em Ingles');
Writes('ESC - abandona');
Repeat

```

```

Read(Kbd, tecla);
tecla := UpCase(tecla);
Until tecla in ['P','I','ESC'];
Case tecla of

```

```

'P' : IapiraeRes;
'I' : IapiraeResIngles;
end;
End;

```

```

Procedure AnalisaM;
var temp, a, raiz : TipoV20;
nr, n, nraiz : Integer;
ok : Boolean;
pbtemp, x0 : Real;
aztemp : TipoV20;
const precisao = 1.0E-7;
MAXFAT = 20.0;

```

```

Function Pol(var a : TipoV20;
             n : Integer;
             c : Real;
             var b : TipoV20) : Real; { vetor dividido por (x - c) }

```

```

var j : Integer;
Begin
    b[n-1] := a[n];
    For j := n-1 downto 1 do b[j-1] := a[j] + c * b[j];
    Pol := a[0] + c * b[0];
End;

```

```

Procedure MNR(var a,
              z : TipoV20;
              var n : Integer;
              var x : Real;
              var ok : Boolean); { coeficientes }
{ coeficientes um grau abaixo }
{ grau, volta um a menos }
{ sugestao inicial e volta raiz }
{ TRUE se existe raiz }

```

```

var P, dp, erro : Real;
b, d : TipoV20;
ntent : Integer;
Begin
    ntent := 0;

```

```

Repeat
  ntent := ntent + 1;
  p := Pol(a, n, x, b);
  dp := Pol(b, n - 1, x, 0);
  If dp < 0.0 then erro := p / dp else erro := 1.0;
  x := x - erro;
Until (Abs(erro) < precisao) or (ntent = 50);
If Abs(erro) < precisao then
  begin
    ok := TRUE;
    z := b;
    n := n - 1;
  end
  else ok := FALSE;
End;

```

```

Procedure AltoGrau;
var ntent : Integer;
  tearaiz : Boolean;
  x : Real;

```

```

  begin
    ntent := 0;
  Repeat
    ntent := ntent + 1;
    x := 20.0 * Random;
    MNR(a, a, n, x, tearaiz);
  Until tearaiz or (ntent = 10);
  If tearaiz then

```

```

    begin
      nraiz := nraiz + 1;
      raiz[nraiz] := x;
    end
    else n := 0;
  End;

```

```

Procedure SegGrau;
var delta : Real;

```

```

  begin
    delta := sqrt(a[1]) - 4 * a[2] * a[0];
    If delta > 0.0 then
      begin
        nraiz := nraiz + 1;
        raiz[nraiz] := (-a[1] + sqrt(delta)) / (2 * a[2]);
        nraiz := nraiz + 1;
        raiz[nraiz] := (-a[1] - sqrt(delta)) / (2 * a[2]);
      end
    else if delta = 0.0 then
      begin
        nraiz := nraiz + 1;
        raiz[nraiz] := -a[1] / (2 * a[2]);
      end;
    n := 0;
  End;

```

```

Procedure AnalisaMisto(or : Integer);

```

```

var f : Real;
  nuspico : Integer;

Procedure Normaliza2;
var intain,
  intmax : TipoV20;
  q,
  somamax,
  somamin,
  erroant,
  erro : Real;

  begin
    { Determina os erros das intensidades. }
    q := 1.0 / f;
    erroant := 0.0;
    If f < 1.0 then
      For i := np downto 1 do
        begin
          erro := erroant * temp[i + 1] / 100.0;
          If erro < erroab then erro := erroab;
          erro := erro + erroant;
          intmax[i] := orig[i] + erro;
          intmin[i] := orig[i] - erro;
          If intmin[i] < 0.0 then intmin[i] := 0.0;
          erroant := f * erro;
        end
      else
        For i := 1 to np do
          begin
            erro := erroant * f * temp[i] / 100.0;
            If erro < erroab then erro := erroab;
            erro := q * (erro + erroant);
            intmax[i] := orig[i] + erro;
            intmin[i] := orig[i] - erro;
            If intmin[i] < 0.0 then intmin[i] := 0.0;
            erroant := erro;
          end;
        end;
    WriteLn;
    WriteLn('Cluster a ser analisado:');
    For i := 1 to np do WriteLn('az[i] :', intmin[i] : 10:5, orig[i] : 10:5,
      intmax[i] : 10:5);
    WriteLn('Tecla para continuar. ');
    Repeat Until KeyPressed;
    { Calcula a intensidade maxima e minima de cada pico. }
    somamax := 0.0;
    somamin := 0.0;
    For i := 1 to np do
      begin
        somamax := somamax + intmax[i];
        somamin := somamin + intmin[i];
      end;
    For i := 1 to np do
      begin
        altmax[i] := intmax[i] / (somamax - intmin[i] + intmax[i]);
        altmin[i] := intmin[i] / (somamax - intmin[i] + intmax[i]);

```

```

end;
{ Normaliza em relacao a soma total. }
somaax := 0.0;
For i := 1 to np do somaax := somaax + orig[i];
For i := 1 to np do alt[i] := orig[i] / somaax;
janormal := TRUE;
Calcd2max;
End;

Begin { AnalisaMisto }
  numpicos := np; { Salva o numero de picos do cluster original. }
  atemp := az; { Salva m/z dos picos. }
  temp := orig; { Salva o cluster original. }
  pbase := picobase; { Salva o pico base. }
  np := np - 1; { O novo cluster tem um pico a menos. }
  For i := 1 to np do { Atualiza m/z dos picos. }
    az[i] := az[i + 1];
  f := raiz(nr); { Calcula as intensidades dos picos. }
  orig(np) := temp(np + 1);
  For i := np - 1 downto 1 do orig[i] := temp[i + 1] - f * orig[i + 1];
  { Se f > 1 entao analisa M - H ao inves de M. }
  If f > 1.0 then For i := 1 to np do orig[i] := orig[i] * f;
  For i := 1 to np do If orig[i] < 0.0 then orig[i] := 0.0;
  Normaliza2;
  jaanal := FALSE;
  AnalisaCluster(TRUE);
  If jaanal then Resultados;
  np := numpicos;
  az := atemp;
  orig := temp;
  picobase := pbase;
End;

Begin { AnalisaM }
  CirScr;
  Writeln('Analise de cluster M com M - H');
  Writeln;
  nraiz := 0;
  n := np - 1;
  For i := 0 to n do
    If Odd(i) then a[i] := -orig[i + 1] else a[i] := orig[i + 1];
  Repeat
    If n > 2 then AltoGrau;
    If n = 2 then SegGrau;
  Until n = 0;
  n := np - 1;
  { Apura as raizes encontradas. }
  For i := 0 to n do
    If Odd(i) then a[i] := -orig[i + 1] else a[i] := orig[i + 1];
  For i := 1 to nraiz do
    begin
      n := np - 1;
      Write(raiz[i]);
      x0 := raiz[i];
      MNR(a, temp, n, x0, ok);
      raiz[i] := x0;
    end;
  End;

  Writeln('raiz[i]');
end;
j := 0;
For i := 1 to nraiz do { Seleciona raizes razoveis. }
begin
  If raiz[i] <= MAXPAT then
    If raiz[i] >= 0.0 then
      begin
        j := j + 1;
        temp[j] := raiz[i];
      end;
    end;
  nraiz := j; { Atualiza }
  raiz := temp;
  If nraiz = 0 then Exit;
  If nraiz = 1 then { Analisa o cluster. }
    AnalisaMisto(1)
  else
    begin
      Repeat
        Writeln('O cluster pode ser formado com a(s) seguinte(s) intensidade(s)');
        Writeln('de M - H em relacao a M: ');
        For i := 1 to nraiz do Writeln(i, ' ', raiz[i] : 8:4);
        Writeln('O abandona. ');
      Until nraiz = 0;
    end;
  End;
End;

Procedure Formula;
var s, num : String255;
    elemento : String2;
    az1, cod, nf,
    n, ph, opc : Integer;
    m, f, erro,
    errof,
    limite, soma : Real;
    g, d : TipoV20;
const stabph = 'Ph';

Procedure Normaliza3;
var c,
    intain,
    intmax : TipoV20;
    somaax,
    somaain,
    erro : Real;
Begin
  { Retorna o cluster corrigido. }
  For i := 1 to np do d[i] := alt[i] * soma;

```

```

( Converte a intensidade do cluster calculado para avaliar o erro. )
For i := 0 to MAXPIC do c[i] := 0;
For i := 1 to npc do c[i] := altc[i] * soma;
( Determina os erros das intensidades. )
( 0 erro absoluto de di e dado por
Edi = Eorigi + f * Eci + ci * Et,
como Eci e considerado zero Edi = Eorigi + ci * Et. )
somaax := 0.0;
somaain := 0.0;
For i := 1 to np do
begin
erro := erroel * orig[i] / 100.0; ( Erro do cluster original. )
If erro < erroab then erro := erroab;
j := az[i] - azi + i;
If ( j < 1 ) or ( j > npc ) then j := 0; ( c[0] = 0 )
erro := erro + errof * c[j]; ( Inclui erro de f. )
intax[i] := d[i] + erro;
intain[i] := d[i] - erro;
If intain[i] < 0.0 then intain[i] := 0.0;
somaax := somaax + intax[i];
somaain := somaain + intain[i];
end;
( Calcula a intensidade maxima e minima de cada pico. )
For i := 1 to np do
begin
altain[i] := intain[i] / (somaax - intax[i] + intain[i]);
altax[i] := intax[i] / (somaax - intain[i] + intax[i]);
end;
( Normaliza em relacao a soma total. )
somaax := 0.0;
For i := 1 to np do somaax := somaax + d[i];
orig := di; ( Atualiza orig com o novo cluster. )
For i := 1 to np do alt[i] := orig[i] / somaax;
Janormal := TRUE;
Calcd2eax;
End;

```

```

Begin ( Formula )
ClrScr;
Writeln('FORMULA');
Writeln;
For el := Carbono to Iodo do Write(Siabolo[el], ' ');
Writeln(Siaboh);
Writeln;
Write('Entre a Formula: ');
Readln(s); s := s + ' ';
For el := Carbono to Iodo do cont[el] := 0;
ph := 0;
i := 1;
Repeat
If s[i] in ['A'..'Z'] then
begin
elemento := s[i];
i := i + 1;
If s[i] in ['a'..'z'] then

```

```

begin
elemento := elemento + s[i];
i := i + 1;
end;
num := '';
While s[i] in ['0'..'9'] do
begin
num := num + s[i];
i := i + 1;
end;
( If num = '' then num := '1';
Val(num, a, cod);
If elemento = Siaboh then ph := ph + n;
For el := Carbono to Iodo do
If elemento = Siabolo[el] then cont[el] := cont[el] + n;
end
else i := i + 1;
Until i > length(s);
cont[Carbono] := cont[Carbono] + 6 * ph; ( Converte fenil em C6H5. )
cont[Hydro] := cont[Hydro] + 5 * ph;
m := 0.0;
For el := Carbono to Iodo do m := m + cont[el] * Iso[el,i];
Writeln('m/z do primeiro pico = ', m :8:4);
azi := Round(m);
Writeln('m/z utilizada nos calculos ', azi);
For i := 0 to MAXPIC do altc[i] := 0.0;
MontaCluster(cont, altc, azi);
npc := MAXPIC;
While (altc[npc] = 0.0) and (npc > 1) do npc := npc - 1;
Writeln;
Writeln('Cluster calculado:');
Writeln(' # m/z altura');
For i := 1 to npc do
Writeln(i2, azi + i - 1, 4, altc[i] :10:5);
Writeln;
If tendados then
begin
( Normaliza em relacao a soma total. )
soma := 0.0;
For i := 1 to np do soma := soma + orig[i];
For i := 1 to np do alt[i] := orig[i] / soma;
limite := erroab / soma; ( Converte o erro absoluto para esta escala. )
For i := 1 to npc do ( Calcula o fator g para cada pico. )
begin
j := azi - m2[i] + i;
If ( j < 1 ) or ( j > np ) then
If altc[i] > limite then
g[i] := 0.0
else g[i] := 1E+6
else
If (altc[i] > 0.0) and (alt[i] > limite) then
g[i] := alt[i] / altc[i]
else g[i] := 1E+6;
end;
Writeln;

```

```

k := i + j - 1;
alt[i] := alt[k];
orig[i] := orig[k];
m[i] := m[k];
end;
Normaliza;
WriteIn;
WriteIn('Cluster com m/z corrigidas:');
WriteIn(' # m/z altura');
for i := 1 to np do WriteIn(i:2, m[i]:4, orig[i]:8:2, ' ');
end;
WriteIn;
WriteIn('Pressione qualquer tecla para continuar...');
repeat until keypressed;
End;

Procedure Menu;
Begin
  WriteIn('MENU');
  WriteIn;
  WriteRes(' - Picos do cluster');
  WriteRes(' - altera Condicoes');
  WriteRes(' - restaura as condicoes Iniciais');
  WriteRes(' - Analisa cluster');
  WriteRes(' - Analisa M - 1');
  WriteRes(' - Resultados');
  WriteRes(' - calcula a massa de uma Formula');
  WriteRes(' - ESC finaliza');
End;

BEGIN
  tendados := FALSE;
  FazPreCalc;
  CondicoesIniciais;
  Repeat
    Menu;
    Read(kbd, opcao);
    opcao := UpCase(opcao);
    Case opcao of
      'P' : EntraCluster;
      'C' : EntraCondicoes;
      'I' : CondicoesIniciais;
      'A' : If tendados then AnalisaCluster(FALSE);
      'M' : If tendados then AnalisaM;
      'R' : If jaanal then Resultados;
      'F' : Formula;
      ESC : begin end;
    else Write(#7);
    end;
  Until (opcao = ESC);
END.

```

```

WriteIn('Pressione qualquer tecla para continuar...');
repeat until keypressed;
WriteIn;
WriteIn('Fator calculado:');
WriteIn(' # m/z g');
For i := 1 to npc do
  begin
    Write(i:2, m[i] + i - 1:4);
    If g[i] < 1.0 then WriteIn(g[i]:9:5) else WriteIn(' > 1');
  end;
  WriteIn;
  f := g[i];
  nf := 1;
  (faz f igual ao menor g.)
  (nf e' usado nos calculos de erros.)
  For i := 2 to npc do
    If g[i] < f then
      begin
        f := g[i];
        nf := i;
      end;
  end;
  WriteIn('Fracao deste cluster no original: ', f:8:5);
  If f > 1.0 then
    WriteIn('Nao e possivel subtrair do cluster original');
  If f > 1.0 then Exit;
  WriteRes('deseja subtrair do cluster original? (S/N)');
  Repeat
    Read(kbd, tecla);
    tecla := UpCase(tecla);
  Until tecla in ['S', 'N'];
  If tecla = 'N' then Exit;
  fatorsub := f;
  (Transcreve para o programa principal.)
  reform := cont;
  (Registra a formula subtraida.)
  j := m[i] - m[nf] + nf;
  (Primeiro calcula o erro do fator.)
  erro := erroel * orig[j] / 100.0;
  If erro < erroab then erro := erroab;
  errof := erro / orig[j];
  (0 termo Ecnf / cnf e' zero pois Ecnf = 0.)
  For i := 1 to npc do
    begin
      j := m[i] - m[nf] + i;
      If (j > 1) and (j <= np) then
        begin
          alt[j] := alt[j] - f * alt[i];
          If alt[j] < 0.0 then alt[j] := 0.0;
        end;
      end;
    WriteIn;
    WriteIn('Cluster resultante:');
    WriteIn(' # m/z altura');
    For i := 1 to np do WriteIn(i:2, m[i]:4, alt[i]:10:5);
    WriteIn;
    j := 1;
    WriteIn('Qual o numero do primeiro pico do cluster? (atual: ', j, ')');
    ReadIn(j);
    np := np - j + 1;
    For i := 1 to np do
      begin

```

```

Programa FA3;
const MAXLIN = 25;
      Versao = '3.1';

type VetorLinha = Array[1..MAXLIN] of Real;
      Matriz0 = Array[1..MAXLIN] of VetorLinha;

var d, r, c, Nula, Id : Matriz0;
    nl, nc, n : Integer;

(d - matriz de dados (m/z X mistura) ou (nl X nc)
 r - matriz dos componentes (m/z X componente) ou (nl X n)
 c - matriz das concentrações (concentração X mistura) ou (n X nc)
 nl - número de linhas
 nc - número de colunas
 n - número de componentes)

($R+)

Procedure PrepararMatrizes;
var zero : VetorLinha;
    i : Integer;
Begin
  For i := 1 to MAXLIN do zero[i] := 0.0;
  For i := 1 to MAXLIN do Nula[i] := zero;
  Id := Nula;
  For i := 1 to MAXLIN do Id[i,i] := 1.0;
End;

Procedure Exibe0(var a : Matriz0; nl, nc, c, d : Integer);
(nl = número de linhas
 nc = número de colunas
 c = campo de apresentação
 d = número de casas decimais)
var i, j : Integer;
Begin
  For i := 1 to nl do
    begin
      For j := 1 to nc do Write(a[i,j]:c:d);
      WriteLn;
    end;
  Write('Tecla para continuar...');
  Repeat Until KeyPressed;
  SoXy(1,WhereY); ClrEol;
  WriteLn;
End;

Procedure Inverte(a : Matriz0; (Matriz de entrada.)
n : Integer; (Dimensão.)
var c : Matriz0; (Matriz inversa.)
var i, coluna,
    linha, troca
    fator
    testeok : Boolean;
Begin
  testeok := TRUE;
  For linha := 1 to n do
    If a[linha,linha] = 0.0 then
      begin
        testeok := False;
        troca := (linha mod n) + 1;
        While (a[troca,linha] = 0.0) do troca := (troca mod n) + 1;
        WriteLn('Trocando a equacao ',linha,' pela equacao ',troca);
        pilha := a[linha];
        a[linha] := a[troca];
        a[troca] := pilha;
        pilha := c[linha];
        c[linha] := c[troca];
        c[troca] := pilha;
      end;
    Until testeok;
    For linha := 1 to n do
      begin
        fator := a[linha,linha];
        If fator = 0.0 then fator := 1E-20;
        For coluna := 1 to n do a[linha,coluna] := a[linha,coluna] / fator;
        c[linha] := c[linha] / fator;
        For i := 1 to n do
          If i <> linha then
            begin
              fator := a[i,linha];
              For coluna := 1 to n do
                a[i,coluna] := a[i,coluna] - fator * a[linha,coluna];
                c[i] := c[i] - fator * c[linha];
              end;
            end;
        end;
      End;

Procedure Inverte(a : Matriz0; (Matriz de entrada.)
n : Integer; (Dimensão.)
var c : Matriz0; (Matriz inversa.)
var i, coluna,
    linha, troca
    fator
    testeok : Boolean;
Begin
  testeok := TRUE;
  For linha := 1 to n do
    If a[linha,linha] = 0.0 then
      begin
        testeok := False;
        troca := (linha mod n) + 1;
        While (a[troca,linha] = 0.0) do troca := (troca mod n) + 1;
        WriteLn('Trocando a equacao ',linha,' pela equacao ',troca);
        pilha := a[linha];
        a[linha] := a[troca];
        a[troca] := pilha;
        pilha := c[linha];
        c[linha] := c[troca];
        c[troca] := pilha;
      end;
    Until testeok;
    For linha := 1 to n do
      begin
        fator := a[linha,linha];
        If fator = 0.0 then fator := 1E-20;
        For coluna := 1 to n do a[linha,coluna] := a[linha,coluna] / fator;
        c[linha] := c[linha] / fator;
        For i := 1 to n do
          If i <> linha then
            begin
              fator := a[i,linha];
              For coluna := 1 to n do
                a[i,coluna] := a[i,coluna] - fator * a[linha,coluna];
                c[i] := c[i] - fator * c[linha];
              end;
            end;
        end;
      End;

Procedure Exibe0(var a : VetorLinha; n, c, d : Integer);
(n = número de elementos
 c = campo de apresentação
 d = número de casas decimais)
var i : Integer;
Begin
  For i := 1 to n do Write(a[i]:c:d);

```

```

For i := 1 to n1 do
  For j := 1 to n2 do
    Read(aq, d[i,j]);
  End;
End;

Procedure Transposta; Matriz0;
  nla, nca : Integer;
  var b : Matriz0;
  var n1b, ncb : Integer;

  var i, j : Integer;
  pilha : Real;
Begin
  n1b := nca;
  ncb := nla;
  For i := 1 to n1a do
    For j := 1 to nca do b[i,j] := a[i,j];
  End;
End;

Procedure Mult(var a : Matriz0;
  var al, ac : Integer;
  var b : Matriz0;
  var bl, bc : Integer;
  var c : Matriz0;
  var cl, cc : Integer);
  { Primeira Matriz }
  { Segunda Matriz }
  { Resultante }

  var i, j, k : Integer;
  soma : Real;
Begin
  If ac = bl then
    begin
      cl := al;
      cc := bc;
      For i := 1 to cl do
        For j := 1 to cc do
          begin
            soma := 0.0;
            For k := 1 to ac do soma := soma + a[i,k] * b[k,j];
            c[i,j] := soma;
          end;
        end;
      end;
    end
  else WriteLn('Matrizes incompativeis', #7);
End;

Procedure Mosento(var a : Matriz0;
  n1, nc : Integer;
  var z : Matriz0;
  var n : Integer);
  { Matriz de entrada. }
  { Mosento. }

  var i, j, k : Integer;
  soma : Real;
Begin
  n := nc;
  For i := 1 to n do
    For j := 1 to n do
      begin
        soma := 0.0;

```

```

  pilha : VetorLinha;
Begin
  c := Id; { Inicializa C. }
Repeat
  testeok := TRUE;
  For linha := 1 to n do
    If a[linha,linha] = 0.0 then
      begin
        testeok := False;
        troca := (linha mod n) + 1;
        While (a[troca,linha] = 0.0) do troca := (troca mod n) + 1;
        WriteLn('Trocando a linha ',linha,' pela linha ',troca);
        pilha := a[linha];
        a[linha] := a[troca];
        a[troca] := pilha;
        pilha := c[linha];
        c[linha] := c[troca];
        c[troca] := pilha;
      end;
    Until testeok;
  For linha := 1 to n do
    begin
      fator := a[linha,linha];
      If fator = 0.0 then fator := 1E-20;
      For coluna := 1 to n do
        begin
          a[linha,coluna] := a[linha,coluna] / fator;
          c[linha,coluna] := c[linha,coluna] / fator;
        end;
      For i := 1 to n do
        If i < linha then
          begin
            fator := a[i,linha];
            For coluna := 1 to n do
              begin
                a[i,coluna] := a[i,coluna] - fator * a[linha,coluna];
                c[i,coluna] := c[i,coluna] - fator * c[linha,coluna];
              end;
            end;
          end;
        end;
      end;
    end;
  For i := 1 to n do
    For j := 1 to n do
      begin
        d[i,coluna] := a[i,coluna] - fator * a[linha,coluna];
        c[i,coluna] := c[i,coluna] - fator * c[linha,coluna];
      end;
    end;
  end;
End;

Procedure Entrabados;
var nome : String[14];
arq : Text;
i, j : Integer;
Begin
  ClrScr;
  d := Nulla;
  WriteLn('Qual o nome do arquivo? (deve estar em B:)');
  ReadLn(nome);
  Assign(arq, 'b' + nome);
  Reset(arq);
  ReadLn(arq, n1); { Entra o numero de linhas. }
  ReadLn(arq, nc); { Entra o numero de colunas. }

```

```

For j := 1 to n do soma := soma + a[k,i] * a[i,j];
z[i,j] := soma;
end;

End;

Procedure G10( a : Matriz0;
n : Integer;
var e : VetorLinha;
var nev : Integer;
var v : Matriz0;
var nvec : Integer);
{ Onde:
a = matriz de entrada
n = dimensao da matriz a
e = vetor contendo os autovalores
nev = numero de autovalores
v = matriz com os autovetores
nvec = numero de autovetores }
const ERR0 = 1E-10;
ZERO = 1E-20;
var i, j, k, ag : Integer;
s, s2,
alfa, Ca,
norm, teste,
u, l, lambda : Real;
b, c, w, p, q : VetorLinha;
r : Matriz0;
Begin
r := a; { Faz copia reserva. }
{----- Foracao da TriDiagonal -----}
For i := 1 to n - 2 do
begin
s2 := 0.0;
For j := i + 1 to n do s2 := s2 + sqr(a[j,i]);
s := sqrt(s2);
If a[i + 1,i] < 0.0 then s := -s;
c[i] := a[i,i];
b[i] := -s;
If s < 0.0 then
begin
alfa := 1 / (s2 + a[i+1,i] * s);
w[i+1,i] := a[i+1,i] + s;
a[i+1,i] := w[i+1,i];
For j := i + 2 to n do w[j] := a[j,i];
For j := i + 1 to n do
begin
p[j] := 0.0;
For k := i + 1 to n do p[j] := p[j] + a[j,k] * w[k];
p[j] := alfa * p[j];
end;
Ca := 0.0;
For j := i + 1 to n do Ca := Ca + w[j] * p[j];
Ca := alfa * Ca / 2.0;
(1)
(2)
(3)
(4)
(5)
(6)
(7)
(8)
(9)
(10)
(11)
(12)
(13)
(14)
(15)
(16)
(17)
(18)
(19)
(20)
(21)
(22)
(23)
(24)
(25)
(26)
(27)
(28)
(29)
(30)
(31)
(32)
(33)
(34)
(35)
(36)
(37)
(38)
(39)
(40)
(41)
(42)
(43)
(44)
(45)
(46)
(47)
(48)
(49)
(50)
(51)
(52)
(53)
(54)
(55)
(56)
(57)
(58)
(59)
(60)
(61)
(62)
(63)
(64)
(65)
(66)
(67)
(68)
(69)
(70)
(71)
(72)
(73)
(74)
(75)
(76)
(77)
(78)
(79)
(80)
(81)
(82)
(83)
(84)
(85)
(86)
(87)
(88)
(89)
(90)
(91)
(92)
(93)
(94)
(95)
(96)
(97)
(98)
(99)
(100)

```

```

For j := i + 1 to n do q[j] := p[j] - Ca * w[j];
For k := i + 1 to n do
For j := i + 1 to n do
a[j,k] := a[j,k] - q[j] * w[k] + q[k] * w[j];
end;
end;
(1)
(2)
(3)
(4)
(5)
(6)
(7)
(8)
(9)
(10)
(11)
(12)
(13)
(14)
(15)
(16)
(17)
(18)
(19)
(20)
(21)
(22)
(23)
(24)
(25)
(26)
(27)
(28)
(29)
(30)
(31)
(32)
(33)
(34)
(35)
(36)
(37)
(38)
(39)
(40)
(41)
(42)
(43)
(44)
(45)
(46)
(47)
(48)
(49)
(50)
(51)
(52)
(53)
(54)
(55)
(56)
(57)
(58)
(59)
(60)
(61)
(62)
(63)
(64)
(65)
(66)
(67)
(68)
(69)
(70)
(71)
(72)
(73)
(74)
(75)
(76)
(77)
(78)
(79)
(80)
(81)
(82)
(83)
(84)
(85)
(86)
(87)
(88)
(89)
(90)
(91)
(92)
(93)
(94)
(95)
(96)
(97)
(98)
(99)
(100)

```

```

e[k] := lambda;      (20)
end;
Writeln;
(----- Calculo dos autovetores -----)
For k := 1 to nvec do
begin
  a := r; { Recupera a matriz inicial. }
  For i := 1 to n do
    begin
      b[i] := Random;      { Termo independente. }
      a[i,i] := a[i,i] - e[k]; { Subtrai o autovetor da diagonal. }
    end;
  Resol(a, b, n); { Determina os autovetores. }
  teste := Abs(b[i]); { Normaliza para maior = 1 }
  For i := 2 to n do
    If Abs(b[i]) > teste then teste := Abs(b[i]);
  For i := 1 to n do b[i] := b[i] / teste;
end;
End;

Procedure DetN(var e : VetorLinha;
               ne : Integer;
               var n : Integer);
var i, j, r, c : Integer;
s, min, RE : Real;
IND : VetorLinha;
Begin
  If n1 < nc then
    begin
      r := nc;
      c := n1;
      ne := n1;
    end
  else
    begin
      r := n1;
      c := nc;
    end;
  Writeln('Determinacao do numero de componentes');
  For i := 1 to ne - 1 do
    begin
      s := 0.0;
      For j := i + 1 to c do s := s + Abs(e[j]);
      RE := sqrt(s / (r * (c - i)));
      IND[i] := RE / sqrt(nc - i);
      Writeln(i2, ' RE = ', RE :6:2, ' IND = ', IND[i] :6:3);
    end;
    n := i;
    min := IND[n];
    For i := 2 to ne - 1 do

```

```

If IND[i] < min then
begin
  n := i;
  min := IND[i];
end;
Writeln('Encontrados ', n, ' componentes --> IND = ', IND[n] :10:4);
Writeln('Corrigir para --> ');
Readln(n);
End;

Procedure Normaliza(var v : Matriz0;
                   n1, nc : Integer);
var i, j : Integer;
s : Real;
Begin
  For i := 1 to n1 do
    begin
      s := 0.0;
      For j := 1 to nc do s := s + sqrt(v[i,j]);
      If s > 0.0 then s := 1.0 / sqrt(s);
      For j := 1 to nc do v[i,j] := s * v[i,j];
    end;
  End;

Function Det(var a : Matriz0;
             n : Integer) : Real;
var r : Matriz0; { Matriz de recursao. }
sinal,
soma : Real;
i, k, m : Integer;
Begin
  If j < n - 1 then
    begin
      soma := 0.0;
      For i := 1 to grau do
        If a[i,j] < 0.0 then
          begin
            m := i;
            For k := 1 to grau do
              If k < i then
                begin
                  { Prepara a nova matriz. }

```

```

jafoi := TRUE;
a[i] := v[k];
end;
End;

Procedure NormalizarC(var r, c : Matriz0;
  nl, nc, n : Integer);
var i, j : Integer;
  fator : Real;
Begin
  For j := 1 to n do
    begin
      fator := r[i,j];
      For i := 2 to nl do if r[i,j] > fator then fator := r[i,j];
      fator := 100.0 / fator;
      For i := 1 to nl do r[i,j] := fator * r[i,j]; { Normaliza o espectro. }
      For i := 1 to nc do c[i,j] := c[i,j] / fator; { Corrige a C. }
      end;
    end;
  End;

Procedure Tratados;
var a, v, vn, q, qt, z : Matriz0;
  e : VetorLinha;
Begin
  Momento(d, nl, nc, z, nc); { Forma a matriz de covarianca. }
  GIB(z, nc, e, nc, q, nc); { Determina os autovalores e autovetores. }
  DetN(e, nc, n); { Determina o numero de componentes. }
  Transposta(q, nc, nc, qt, nc, nc); { Calcula Q' }
  Multid, nl, nc, q, nc, nc, vn, nl, nc); { V = BQ (nl X nc) }
  vn := v; { Duplica V }
  Normaliza(vn, nl, n); { Normaliza as linhas de V. }
  DetA(vn, a, n, nl); { Encontra a matriz de rotacao A (n X n). }
  Mult(a, n, q, qt, n, nc, c, n, nc); { C = AQ' (n X nc) }
  Inverte(a, n, a); { R = VA-1 }
  Multiv, nl, n, a, n, r, nl, n);
  NormalizarC(r, c, nl, nc, n);
  WriteLn('Matriz concentracao (C):');
  ExibeM(c, n, nc, 7, 3);
  WriteLn('Matriz componentes (R):');
  ExibeM(r, nl, n, 7, 1);
End;

Procedure Resultados;
var arq : Text;
  s : String[64];
  car : Char;
Begin
  WriteLn('Matriz componentes (R):');
  ExibeM(r, nl, n, 7, 1);
End;

Procedure Manda(var a : Matriz0; nl, nc, c, d : Integer);
var i, j : Integer;
Begin
  For i := 1 to nl do
    begin
      For j := 1 to nc do Write(arq, a[i,j] :cid);
      WriteLn(arq);
    end;
  end;

```

```

CalcDet := a[1,n - 1] * a[2,n] - a[1,n] * a[2,n - 1];
end;
End;

Begin
  Det := CalcDet(a, 1, n);
End;

Procedure DetA( v : Matriz0;
  var a : Matriz0;
  n, nl : Integer);
var i, j, k : Integer;
  jafoi : Array[1..MAXLN] of Boolean;
  d, min, max : Real;
Begin
  For i := 1 to n do jafoi[i] := FALSE;
  k := 1;
  { Localiza a primeira linha chave. }
  For i := 2 to nl do if Abs(v[i,1]) < Abs(v[k,1]) then k := i;
  Write('Linha chave = ', k, ' -> '); ReadLn(k);
  jafoi[k] := TRUE; { Anota a linha ja selecionada. }
  a[i] := v[k];
  If n = 1 then Exit;
  If jafoi[1] then k := 2 else k := 1; { Localiza a segunda linha chave. }
  min := Abs(a[1,1]) * v[k,1] + a[1,2] * v[k,2];
  For i := 1 to nl do
    If not jafoi[i] then
      begin
        d := Abs(a[1,1] * v[i,1] + a[1,2] * v[i,2]);
        If d < min then
          begin
            min := d;
            k := i;
          end;
        end;
      end;
  Write('Linha chave = ', k, ' -> '); ReadLn(k);
  jafoi[k] := TRUE;
  a[2] := v[k];
  For i := 3 to n do
    { Localiza as linhas chaves restantes. }
    begin
      k := 1;
      While jafoi[k] do k := k + 1; { k aponta para uma linha nao analisada. }
      a[i] := v[k];
      max := Abs(Det(a, i));
      For j := 1 to nl do
        If not jafoi[j] then
          begin
            a[i] := v[j];
            d := Abs(Det(a, i));
            If d > max then
              begin
                max := d;
                k := j;
              end;
            end;
          end;
        end;
      Write('Linha chave = ', k, ' -> '); ReadLn(k);
    end;
  end;

```

```
End;

Begin (Resultados)
  WriteLn('deseja salvar? (S/N) '); ReadLn(car);
  If UpCase(car) <> 'S' then Exit;
  Write('Destino: '); ReadLn(s);
  Assign(arq,s);
  Rewrite(arq);
  WriteLn(arq,'Matriz concentracao (C):');
  Manda(C, n, nc, 7, 3);
  WriteLn(arq);
  WriteLn(arq,'Matriz componentes (R):');
  Manda(r, nl, n, 7, 1);
  Close(arq);
End;

BEGIN
  PrepararMatrizes;
  Entrabados;
  Tratados;
  Resultados;
END.
```

APÊNDICE 3

RESULTADOS DE ANÁLISES DESCRITAS NO CAPÍTULO VI

A seguir são apresentados os dados utilizados e resultados obtidos pelo programa FA3 no Capítulo VI. As matrizes concentração estão no formato componentes X misturas e as matrizes componentes no formato m/z X componentes puros. Tanto as misturas quanto os valores de m/z se referem a matriz de dados.

Cromatograma do Cloreto de Metileno: A primeira coluna da matriz de dados são os valores de m/z em daltons e as outras colunas correspondem respectivamente aos espectros 23, 25, 28, 33, 35, 40, 50 e 70.

29	84	53	26	0	0	0	0	0
31	121	53	0	0	0	0	0	0
35	256	539	757	482	328	221	175	209
36	0	0	41	51	72	64	44	32
37	66	173	251	105	77	55	36	73
41	0	0	0	39	49	71	43	38
47	72	215	429	464	441	312	265	306
48	109	167	237	282	244	181	150	202
49	2063	2523	2861	2345	1930	1650	1394	1594
50	51	72	120	114	94	66	42	59
51	792	971	825	667	598	479	405	486
82	480	485	318	125	118	30	37	51
83	1120	1525	1270	681	553	389	318	367
84	390	638	1072	1484	1527	1452	1215	1376
85	816	932	1138	561	456	321	247	295
86	110	192	708	971	932	971	816	956
87	156	175	243	107	109	82	46	100
88	0	59	83	190	181	170	127	171
93	0	0	35	0	27	0	0	0
94	0	45	44	0	0	0	0	0
95	75	103	91	0	23	0	0	0
96	198	257	150	57	0	0	0	0
97	283	447	311	66	24	0	0	0
98	120	169	98	31	0	0	0	0
99	165	304	198	46	36	0	0	0

- APÊNDICE 3 -

Matriz concentracao (C):

15.306	13.698	-1.085	-1.103	0.158	0.222	0.692	0.364
-1.146	1.351	13.289	19.350	12.628	18.116	15.072	17.385
6.020	10.689	16.415	4.654	0.838	-1.730	-1.860	-1.707

Matriz componentes (R):

3.9	0.0	1.5
6.1	-0.0	-0.0
6.8	15.8	35.2
0.0	3.0	0.0
1.3	3.9	12.0
1.2	2.7	-2.1
2.9	20.1	11.7
5.8	11.9	6.2
100.0	100.0	100.0
1.6	4.2	4.4
43.3	29.5	29.9
23.4	4.0	17.3
55.8	25.7	60.8
32.9	79.5	3.4
30.6	21.2	51.7
10.3	52.5	-0.3
5.7	5.1	10.1
2.7	9.4	-1.4
-0.8	0.3	1.5
0.0	0.0	2.7
3.2	0.3	5.1
10.0	0.6	9.7
14.0	0.9	19.3
6.2	0.3	6.4
8.9	0.8	12.5

Estudo sobre o Potencial de Ionização

A primeira tabela se refere aos espectros de massas completos obtidos a 11 valores de potencial de ionização: 10, 15, 20, 25, 30, 35, 40, 45, 50, 55 e 60eV. Na segunda tabela foram selecionados os valores de m/z que atingiram intensidade superior a 5% do pico base global. Os resultados (análise 1) referem-se a esta segunda tabela.

- APÊNDICE 3 -

m/z \ Ex	1	2	3	4	5	6	7	8	9	10	11
12 d	0.0	0.0	0.0	0.1	0.3	1.1	1.8	3.0	4.5	4.4	4.2
13 d	0.0	0.0	0.1	0.1	0.8	3.9	5.3	7.1	9.9	9.3	8.6
14 d	0.2	0.1	0.1	1.3	6.3	9.4	9.1	10.6	13.8	12.2	11.0
15 d	0.1	0.0	0.0	0.1	0.2	0.3	0.3	0.3	0.3	0.4	0.4
18 d	0.0	0.0	0.2	0.2	0.3	0.4	0.4	0.4	0.5	0.5	0.6
28 d	0.1	0.1	0.2	0.3	0.5	0.7	0.7	0.9	0.9	1.1	1.1
29 d	0.0	0.1	0.0	0.0	0.2	0.2	0.2	0.2	0.3	0.4	0.4
31 d	0.1	0.0	0.1	0.2	0.2	0.3	0.3	0.3	0.4	0.4	0.4
32 d	0.1	0.0	0.1	0.2	0.2	0.3	0.3	0.3	0.2	0.3	0.3
35 d	0.0	0.0	0.0	0.1	0.8	3.8	6.6	10.2	18.0	18.8	18.5
36 d	0.0	0.1	0.9	2.0	2.8	4.3	4.9	5.4	6.5	6.8	7.0
37 d	0.0	0.0	0.1	0.1	0.4	1.2	2.1	3.7	5.9	6.2	6.2
38 d	0.1	0.1	0.3	0.6	0.8	1.3	1.5	1.7	2.1	2.1	2.3
41 d	0.0	0.0	0.1	0.1	0.1	0.3	0.5	1.1	1.7	2.4	3.1
42 d	0.1	0.0	0.1	0.1	0.2	0.2	0.3	0.6	1.0	1.3	1.7
43 d	0.1	0.0	0.0	0.0	0.1	0.2	0.3	0.3	0.5	0.6	0.7
47 d	0.1	0.1	0.1	2.6	6.9	12.2	14.0	16.7	22.5	22.8	22.7
48 d	0.0	0.0	0.8	2.8	5.0	6.7	6.6	7.2	9.7	10.0	10.1
49 d	0.0	0.1	19.8	43.2	48.4	59.9	58.9	69.6	90.3	95.4	100.0
50 d	0.0	0.0	0.6	1.4	2.1	2.9	2.8	3.5	4.5	4.7	4.9
51 d	0.0	0.0	6.4	13.5	14.7	17.9	17.3	20.7	26.5	28.2	29.7
52 d	0.0	0.1	0.1	0.2	0.2	0.4	0.3	0.3	0.3	0.4	0.4
55 d	0.0	0.0	0.1	0.1	0.1	0.2	0.1	0.2	0.2	0.3	0.4
62 d	0.1	0.0	0.1	0.2	0.3	0.5	0.5	0.6	0.6	0.7	0.8
63 d	0.0	0.0	0.4	1.0	1.1	1.3	1.2	1.4	1.8	1.9	2.1
64 d	0.0	0.1	7.0	18.0	21.8	25.4	22.5	28.9	32.4	36.0	38.5
65 d	0.0	0.0	0.2	0.3	0.5	0.6	0.5	0.6	0.7	0.8	0.8
66 d	0.1	0.0	0.9	2.9	4.6	5.0	3.7	5.5	5.7	6.7	6.5

m/z \ Ex	1	2	3	4	5	6	7	8	9	10	11
13 d	0.0	0.0	0.1	0.1	0.8	3.9	5.3	7.1	9.9	9.3	8.6
14 d	0.2	0.1	0.1	1.3	6.3	9.4	9.1	10.6	13.8	12.2	11.0
35 d	0.0	0.0	0.0	0.1	0.8	3.8	6.6	10.2	18.0	18.8	18.5
36 d	0.0	0.1	0.9	2.0	2.8	4.3	4.9	5.4	6.5	6.8	7.0
37 d	0.0	0.0	0.1	0.1	0.4	1.2	2.1	3.7	5.9	6.2	6.2
47 d	0.1	0.1	0.1	2.6	6.9	12.2	14.0	16.7	22.5	22.8	22.7
48 d	0.0	0.0	0.8	2.8	5.0	6.7	6.6	7.2	9.7	10.0	10.1
49 d	0.0	0.1	19.8	43.2	48.4	59.9	58.9	69.6	90.3	95.4	100.0
51 d	0.0	0.0	6.4	13.5	14.7	17.9	17.3	20.7	26.5	28.2	29.7
64 d	0.0	0.1	7.0	18.0	21.8	25.4	22.5	28.9	32.4	36.0	38.5
66 d	0.1	0.0	0.9	2.9	4.6	5.0	3.7	5.5	5.7	6.7	6.5

- APÊNDICE 3 -

Matriz concentracao (C):

0.000	-0.000	0.000	0.001	0.008	0.038	0.066	0.102	0.180	0.188	0.185
0.002	0.000	0.023	0.076	0.121	0.129	0.096	0.144	0.151	0.169	0.172
0.032	0.001	0.001	0.013	0.063	0.094	0.091	0.106	0.138	0.123	0.110
-0.000	0.001	0.193	0.414	0.449	0.545	0.528	0.632	0.810	0.861	0.906
-0.000	0.001	0.024	0.064	0.084	0.134	0.148	0.165	0.201	0.206	0.215
0.000	0.000	0.001	0.001	0.009	0.038	0.053	0.071	0.099	0.092	0.086

Matriz componentes (R):

-0.0	-0.0	-0.0	0.0	0.0	100.0
-0.0	-0.0	100.0	0.0	0.0	0.0
100.0	-0.0	0.0	-0.0	0.0	0.0
0.0	-0.0	-0.0	0.0	32.6	0.0
27.2	10.3	-11.6	-0.5	-4.4	24.1
59.9	14.6	44.8	-13.4	100.0	-59.9
25.3	-3.1	42.8	0.3	36.6	-79.8
29.5	-14.3	40.0	100.0	36.8	-68.1
-0.0	0.0	-0.0	32.8	0.0	0.0
-9.8	100.0	-49.7	16.7	65.4	-8.4
-0.0	38.4	-0.0	-0.0	0.0	0.0

Análise 2: A região de m/z vai de 12 a 15d.

Matriz concentracao (C):

0.003	0.003	0.002	0.004	0.021	0.080	0.130	0.218	0.326	0.317	0.302
0.012	0.008	0.009	0.093	0.428	0.527	0.466	0.548	0.719	0.627	0.570
0.003	0.002	0.004	0.004	0.060	0.282	0.384	0.516	0.716	0.675	0.622

Matriz componentes (R):

100.0	0.0	-0.0
0.0	-0.0	100.0
-89.8	100.0	80.1
0.0	4.1	0.0

Análise 3: A região de m/z vai de 35 a 38d.

Matriz concentracao (C):

0.002	0.004	0.047	0.107	0.149	0.230	0.260	0.285	0.344	0.360	0.372
0.001	0.002	0.001	0.037	0.044	0.200	0.350	0.543	0.958	1.000	0.982
0.002	0.002	0.004	0.006	0.021	0.065	0.113	0.196	0.312	0.329	0.327

Matriz componentes (R):

0.0	100.0	-0.0
100.0	-0.0	0.0
-0.0	0.0	100.0
27.8	-7.8	28.4

- APÊNDICE 3 -

Análise 4: A região de m/z vai de 35 a 38d.

Matriz concentracao (C):

0.002	0.004	0.047	0.107	0.149	0.230	0.259	0.286	0.344	0.360	0.373
0.001	0.002	0.001	0.007	0.045	0.198	0.348	0.547	0.957	1.000	0.983

Matriz componentes (R):

0.0	100.0
100.0	-0.0
2.9	32.0
28.6	1.2

Análise 5: A região de m/z vai de 47 a 52d.

Matriz concentracao (C):

0.001	0.001	0.001	0.026	0.069	0.122	0.140	0.167	0.225	0.228	0.227
-0.000	-0.000	0.003	0.006	0.009	0.011	0.008	0.010	0.011	0.013	0.014
-0.000	0.001	0.199	0.420	0.458	0.557	0.538	0.643	0.825	0.877	0.925
-0.000	0.000	0.009	0.031	0.055	0.074	0.073	0.079	0.107	0.110	0.112

Matriz componentes (R):

100.0	-0.0	0.0	0.0
0.0	0.0	-0.0	90.6
4.3	-343.9	100.0	100.0
13.7	100.0	2.2	-14.2
-0.0	0.0	32.1	0.0
0.0	30.6	-0.0	0.0

Análise 6: A região de m/z vai de 82 a 86d.

Matriz concentracao (C):

0.001	0.001	0.003	0.006	0.007	0.014	0.013	0.016	0.016	0.018	0.022
0.004	0.000	0.046	0.152	0.238	0.262	0.194	0.286	0.295	0.349	0.337
-0.002	0.002	0.135	0.317	0.329	0.401	0.391	0.467	0.549	0.587	0.666

Matriz componentes (R):

100.0	-0.0	-0.0
-0.0	0.0	8.2
-14.4	100.0	100.0
1.3	3.0	1.7
-0.0	50.1	0.0

- APÊNDICE 3 -

Análise 7: A primeira tabela apresenta os espectros de 49 a 53d do benzeno a 7 valores de potencial de ionização: 50, 45, 40, 35, 30, 25 e 20eV.

8.7	6.4	3.3	0.6	0.5	0.0	0.4
78.3	79.0	72.8	49.9	21.7	6.9	2.4
94.2	98.5	100.0	87.5	60.2	25.0	2.4
90.7	93.2	93.3	83.4	77.3	55.3	28.5
4.7	4.2	4.3	3.5	3.8	2.5	2.0

Matriz concentracao (C):

0.087	0.064	0.033	0.006	0.005	-0.000	0.004
0.774	0.749	0.698	0.593	0.611	0.482	0.285
0.942	0.985	1.000	0.875	0.602	0.250	0.024
0.783	0.790	0.728	0.499	0.217	0.069	0.024

Matriz componentes (R):

100.0	-0.0	-0.0	-0.0
-0.0	-0.0	-0.0	100.0
0.0	0.0	100.0	0.0
-161.9	100.0	27.7	1.6
0.0	5.9	0.0	-0.0