

UNIVERSIDADE FEDERAL DE SANTA CATARINA

CENTRO DE CIÊNCIAS FÍSICAS E MATEMÁTICAS

CURSO DE PÓS - GRADUAÇÃO EM QUÍMICA

TESE DE DOUTORADO

**DESENVOLVIMENTO DE UM SISTEMA AUTOMÁTICO E
TERMOSTATIZADO PARA MEDIDAS DE
TENSÃO SUPERFICIAL**

ARILSON LEHMKUHL

ORIENTADOR: PROF. FARUK NOME

**FLORIANÓPOLIS
SANTA CATARINA - BRASIL
FEVEREIRO DE 1998**

**DESENVOLVIMENTO DE UM SISTEMA
AUTOMÁTICO E TERMOSTATIZADO PARA
MEDIDAS DE TENSÃO SUPERFICIAL**

Arilson Lehmkuhl

Esta tese foi julgada adequada para a obtenção do grau de Doutor em Química, especialidade em Química Analítica, e aprovada em sua forma final pelo Orientador e demais membros da Banca Examinadora.

Faruk Nome

Prof. Dr. Faruk Nome
Orientador

Valdir Soldi
Prof. Dr. Valdir Soldi
Coordenador

Banca Examinadora:

Faruk Nome

Prof. Dr. Faruk Nome
Orientador

Célio Pasquini

Prof. Dr. Célio Pasquini
Instituto de Química - UNICAMP

Boaventura Freire dos Reis
Prof. Dr. Boaventura Freire dos Reis
Centro de Energia Nuclear na Agricultura - USP

Ivan Gonçalves de Souza
Prof. Dr. Ivan Gonçalves de Souza
Departamento de Química - UFSC

Valdir Soldi
Prof. Dr. Valdir Soldi
Departamento de Química - UFSC

Florianópolis, 13 de fevereiro de 1998.

AGRADECIMENTOS

Ao professor Faruk Nome, que desde os tempos da minha graduação soube descobrir quando eu enrolava e quando trabalha sério.

Ao Célio Pasquini, Boaventura Freire dos Reis, Ivan Gonçalves de Souza e Valdir Soldi pela colaboração e apoio.

Aos meu amigos Vanderlei (“mas é um baita”) pelas correções de português, Cláudio (“vamos tomar uma cerveja? Mas tem que ser agora.”) pelas discussões sobre química em mesas de bar, Edgard pelas idéias e projetos e Vera (lagartixa), pelo auxílio e orientação.

Aos professores Faruk Nome, Maria da Graça do Nascimento, Marcos Caroli Rezende, Ivan Gonçalves de Souza, Antônio Carlos Jousef, Sonia Probst e Hugo Gallardo que, sabendo ou não, influenciaram em muitas decisões importantes da minha vida. Sorte a minha...

Aos demais professores do departamento de química, que contribuíram para a minha formação, em especial ao Marcos Aires de Brito, Eduardo Pinheiro, Alfredo Tibúrcio, Roldão Queiróz, Marilene Krug, Frank Quina e Bertolino, além dos citados anteriormente. Aos professores-colegas Dilma Marconi e José Gesser.

Aos colegas do 303/305, Álvaro, Alex, Edson (gordo), Josiel, Edson (cabeludo), Ângelo (passarinho), Santiago, Sandro, Gean, Fabiana, Márcio, Francine, Sônia, Adriano e Sérgio.

Aos colegas e amigos Paulo (negão, “eu não estou puto, mas é muita sacanagem), Pedro (pedrinho”, “Maria, dá três pulinhos”), Donizete (alemão), Jair, Neide e Sandra do 301.

Ao César, ao Bento, ao Reinaldo, à Angélica e à Bertília, que tanto me viram trabalhando e tomando café pelo 209.

Aos colegas, amigos e companheiros dos outros laboratório, Jacir ("el magro"), Elisane, Antoninho, Fábio (borboleta), Robério, Anderson, Simone, Tânia, Guedes, Chico e Niero.

Aos demais colegas e amigos que eu deveria lembrar, porém, digitando este texto às 0:15 hs, a minha memória já não faz justiça.

Aos funcionários do departamento e do Centro, em especial a Tânia, ao Jadir e a Graça.

Ao CNPq pelo auxílio financeiro, sem o qual eu jamais teria realizado este trabalho.

Aos meus pais e irmão que sempre deram apoio nos momentos mais importantes.

À Elisângelia, Gabriela e Letícia que sentiram muita falta do pai nas horas de ausência.

Á Elisa, a qual eu nem tenho palavras para agradecer....

A Ele que, de uma forma ou de outra, foi o verdadeiro autor de tudo isto.

**“Ganhar o Prêmio Nobel não deixa a pessoa mais inteligente,
mas a torna mais visível”**

José Saramago, Prêmio Nobel da literatura em 1998

“Deus não joga dados”

**Albert Einstein, ao discutir com Bohr a respeito
do Princípio da Incerteza de Heisenberg**

“Não cabe a nós prescrever que métodos

Ele usa para reger o universo”

Bohr, em resposta à Einstein

APRESENTAÇÃO

Esta tese de doutorado foi dividida em 6 (seis) capítulos, de acordo com a descrição apresentada a seguir.

O primeiro capítulo (*introdução*) contempla os objetivos da tese, apresentando inúmeros trabalhos da literatura que justificam os objetivos. Em seguida, algumas propriedades de interesse, dos surfactantes e das suas misturas, são descritas. Para finalizar, é apresentado o método do peso da gota, para medidas de tensão superficial e para determinação da CMC e de várias outras propriedades físico-químicas dos surfactantes.

O segundo capítulo (*automação*) inicia com um breve comentário a respeito das técnicas de fluxo. Em seguida, é apresentada a potencialidade de um sistema microprocessado e de algumas linguagens de programação, sendo, inclusive, descrito o funcionamento das rotinas de interrupção e de outras rotinas controladas por intervalo de tempo. Conclui-se este capítulo com uma abordagem a respeito da comunicação paralela e serial. Exemplos de circuitos eletrônicos de "drive" de potência e de detetor ótico são apresentados, bem como programas exemplo para escrita e leitura da interface paralela e para leitura, via interface serial.

O capítulo 3 (*experimental*) relata os reagentes e os equipamentos utilizados e o sistema para medidas de tensão superficial, incluindo o seu funcionamento. Aqui, são apresentadas as equações utilizadas, os parâmetros necessários para se introduzir no programa e a forma como é feita a análise dos dados. O fluxograma do programa de controle, aquisição e tratamento de dados também é apresentado (os códigos-fontes dos programas desenvolvidos encontram-se nos apêndices 2, 3 e 4). Na sequência, o sistema de termostatização, que foi especialmente montado para o tensiômetro, é descrito, incluindo todos os adaptadores e a linha termostatizada de fluxo. Alguns resultados, referentes à calibração do raio do capilar, são apresentados no final deste capítulo.

O capítulo 4 (*resultados e discussões*) inicia com a comprovação das medidas de tensão superficial e concentração. A primeira avaliação do sistema é feita por medidas de erro no cálculo da concentração, avaliado por espectrofotometria. Posteriormente, medidas de massa da gota da água e de tensão superficial do SDS, em função da vazão, são realizadas. Os valores da CMC, Γ , a e ΔG_{MIC}^0 do SDS, em função da vazão, também são determinados e os resultados discutidos. O CTABr também foi estudado. Para finalizar a avaliação do sistema, medidas de tensão superficial da água, em várias temperaturas, foram realizadas e comparadas com os dados disponíveis na literatura.

Ainda no capítulo 4, foram feitas várias determinações, com misturas de surfactantes. O sistema CTABr / TTABr foi estudado apresentando um comportamento ideal. Os sistemas CTABr / SB3-14, TTABr / SB3-14 e DOTAB / SB3-14 apresentaram comportamento muito próximo ao ideal. Porém, os sistemas SB3-10 / SDS, SB3-12 / SDS, SB3-14 / SDS, SB3-10 / SDOS, SB3-12 / SDOS e SB3-14 / SDOS apresentaram grande interação sinérgica entre os seus componentes. Para cada mistura, foi determinado o valor para o parâmetro de interação β , de Holiand e Rubingh.

No capítulo 5 (*considerações gerais*), é feita uma abordagem geral a respeito de todas as características do instrumento desenvolvido e dos resultados experimentais das misturas de surfactantes.

O último capítulo (*referências bibliográficas*) apresenta toda a bibliografia utilizada neste trabalho, incluindo os manuais de operações, os catálogos e similares, tão comuns na área de automação.

O Apêndice 1 contém a figura de um conector DB-25 e a sua pinagem na configuração para comunicação paralela da impressora e da interface MQI12/8PCC e da interface serial. Os Apêndices 2, 3 e 4 apresentam os códigos-fonte dos programas desenvolvidos neste trabalho. O Apêndice 5 contém o manual de instruções do sistema desenvolvido, que servirá de apoio a outros pesquisadores quando esses forem realizar medidas de tensão superficial.

Particularmente, eu espero que este instrumento seja de grande utilidade e que muitos trabalhos sejam nele desenvolvidos. Para o futuro, pretendo melhorar este equipamento diversificando o sistema de medidas.

SUMÁRIO

	PAG.
CAPÍTULO I - INTRODUÇÃO	1
1.1 - OBJETIVOS.....	2
1.2 - REVISÃO DA LITERATURA.....	2
1.3 - SURFACTANTES	3
1.4 - CONCENTRAÇÃO MICELAR CRÍTICA	4
1.5 - MISTURAS DE SURFACTANTES.....	5
1.5.1 - MISTURAS IDEAIS	5
1.5.2 - MISTURAS NÃO IDEAIS.....	6
1.6 - TENSÃO SUPERFICIAL.....	7
CAPÍTULO II - AUTOMAÇÃO	11
2.1 - TÉCNICAS DE FLUXO.....	12
2.2 - SISTEMAS MICROPROCESSADOS	12
2.3 - PROGRAMAÇÃO	14
2.4 - ROTINAS NORMAIS E ROTINAS DE INTERRUPTÃO	15
2.5 - COMUNICAÇÃO PARALELA	18
2.5.1 - INTERFACE PARALELA COMO SAÍDA.....	18
2.5.2 - INTERFACE PARALELA COMO ENTRADA.....	21
2.6 - COMUNICAÇÃO SERIAL.....	23
CAPÍTULO III - EXPERIMENTAL.....	27
3.1 - REAGENTES.....	28
3.2 - SOLUÇÕES.....	28
3.3 - EQUIPAMENTOS	29
3.4 - DESCRIÇÃO DO TENSÍMETRO	29
3.5 - CÁLCULOS MATEMÁTICOS	32
3.6 - PROGRAMA DE CONTROLE, AQUISIÇÃO E TRATAMENTO DE DADOS	33
3.7 - SISTEMA DE TERMOSTATIZAÇÃO.....	36
3.8 - DETERMINAÇÃO DO RAIOS DO CAPILAR	41

CAPÍTULO IV - RESULTADOS E DISCUSSÕES.....	42
4.1 - COMPROVAÇÃO DO SISTEMA DE TENSÃO SUPERFICIAL	43
4.1.1 - ERRO NO CÁLCULO DA CONCENTRAÇÃO.....	43
4.1.2 - MASSA DA GOTA DE ÁGUA EM FUNÇÃO DA VAZÃO	44
4.1.3 - TENSÃO SUPERFICIAL DO SDS EM VÁRIAS VAZÕES.....	46
4.1.4 - TENSÃO SUPERFICIAL E OUTRAS PROPRIEDADES DO CTABr.....	49
4.1.5 - MEDIDAS DE TENSÃO SUPERFICIAL EM VÁRIAS TEMPERATURAS.....	50
4.2 - MISTURAS DE SURFACTANTE	52
4.2.1 - SISTEMA IDEAL	52
4.2.2 - MISTURA DE SURFACTANTES CATIÔNICOS E SULFOBETAÍNAS.....	54
4.2.3 - MISTURA DE SDS E SULFOBETAÍNAS	58
4.2.4 - MISTURA DE SDOS E SULFOBETAÍNAS	62
4.2.5 - VALORES DE β PARA AS MISTURAS DE SURFACTANTES	66
 CAPÍTULO V - CONSIDERAÇÕES GERAIS	 68
5.1 - TENSÍÔMETRO.....	69
5.2 - MISTURA DE SURFACTANTES	71
5.3 - OUTROS SISTEMAS QUÍMICOS	72
 CAPÍTULO VI - REFERÊNCIAS BIBLIOGRÁFICAS.....	 73
 APÊNDICES.....	 83

ÍNDICE DE FIGURAS

	PAG.
Figura 1 - Representação esquemática de uma micela esférica preenchida com óleo.....	4
Figura 2 - Forças de atração que atuam nas partículas de um líquido	8
Figura 3 - Tensão superficial, γ , em função do logaritmo da concentração de surfactante.....	9
Figura 4 - Sistema microcontrolado para gerenciamento e leitura de dados	13
Figura 5 - Exemplo de uma rotina de interrupção em Turbo Pascal®	16
Figura 6 - Rotina acessada com base no tempo em Turbo Pascal®	17
Figura 7 - Rotina acessada com base no tempo em Qbasic®	18
Figura 8 - Circuito eletrônico do “drive” de potência para utilização em conjunto à interface paralela	19
Figura 9 - Programa para acionamento da interface paralela da impressora	21
Figura 10 - Circuito para leitura de um detetor ótico	22
Figura 11 - Programa para leitura da interface paralela da impressora.....	22
Figura 12 - Rotina de interrupção do programa de leitura da interface serial	24
Figura 13 - Rotina de preparação do protocolo para comunicação serial	25
Figura 14 - Programa de leitura da interface serial (COM2) em 1200 bauds	26
Figura 15 - Representação esquemática do tensiômetro automatizado com duas bombas de seringa	30
Figura 16 - Representação esquemática do tensiômetro automático	31
Figura 17 - Fluxograma do programa para o tensiômetro	34
Figura 18 - Adaptador de acrílico da linha termostatzada (L)	36
Figura 19 - Sistema de termostatzação para testar a linha de fluxo	37

Figura 20 - Curva de calibração do NTC com o termômetro de mercúrio (T1). Relação entre a temperatura do banho (T1) e a temperatura em T2	38
Figura 21 - Sistema de termostatização para o tensiômetro	39
Figura 22 - Sistema completo para medidas de tensão superficial	40
Figura 23 - Massa da gota de água em função da vazão, a 25°C	45
Figura 24 - Tensão superficial em função da concentração do SDS em diferentes vazões, a 25°C.	46
Figura 25 - Ampliação da região da CMC do gráfico de tensão superficial em função da concentração do SDS em diferentes vazões, a 25°C	47
Figura 26 - Valores de CMC e da variação da energia livre do processo de formação das micelas ($\Delta G^0_{MIC.}$) do SDS em função da vazão	48
Figura 27 - Valores do excesso na superfície (Γ) e da área superficial molecular (a) do SDS em função da vazão	48
Figura 28 - Tensão superficial do CTABr em função da concentração, a 25°C	49
Figura 29 - Temperatura selecionada no sistema de termostatização e temperatura real da gota	51
Figura 30 - (a) Tensão superficial em função da concentração total da mistura [TTABr + CTABr], a 25°C. (b) Valores das CMCs experimentais da mistura em função da fração molar de TTABr	53
Figura 31 - (a) Tensão superficial em função da concentração total da mistura [DOTAB + SB3-14], a 25°C. (b) Valores das CMCs experimentais da mistura em função da fração molar de DOTAB	55
Figura 32 - (a) Tensão superficial em função da concentração total da mistura [TTABr + SB3-14], a 25°C. (b) Valores das CMCs experimentais da mistura em função da fração molar de TTABr	56

- Figura 33 - (a) Tensão superficial em função da concentração total da mistura [CTABr + SB3-14], a 25°C.
(b) Valores das CMCs experimentais da mistura em função da fração molar de CTABr57
- Figura 34 - (a) Tensão superficial em função da concentração total da mistura [SDS + SB3-10], a 25°C.
(b) Valores das CMCs experimentais da mistura em função da fração molar de SDS.....59
- Figura 35 - (a) Tensão superficial em função da concentração total da mistura [SDS + SB3-12], a 25°C.
(b) Valores das CMCs experimentais da mistura em função da fração molar de SDS.....60
- Figura 36 - (a) Tensão superficial em função da concentração total da mistura [SDS + SB3-14], a 25°C.
(b) Valores das CMCs experimentais da mistura em função da fração molar de SDS.....61
- Figura 37 - (a) Tensão superficial em função da concentração total da mistura [SDOS + SB3-10], a 25°C.
(b) Valores das CMCs experimentais da mistura em função da fração molar de SDOS.....63
- Figura 38 - (a) Tensão superficial em função da concentração total da mistura [SDOS + SB3-12], a 25°C.
(b) Valores das CMCs experimentais da mistura em função da fração molar de SDOS.....64
- Figura 39 - (a) Tensão superficial em função da concentração total da mistura [SDOS + SB3-14], a 25°C.
(b) Valores das CMCs experimentais da mistura em função da fração molar de SDOS.....65
- Figura 40 - Valores do parâmetro β para as misturas de DOTAB, TTABr e CTABr com o SB3-14 em função do número de átomos de carbono do surfactante catiônico66

Figura 41 - Valores do parâmetro β para as misturas de SB3-10, SB3-12 e SB3-14 com o SDS e SDOS em função do número de átomos de carbono da sulfobetaina.....67

ÍNDICE DE TABELAS

	PAG.
Tabela 1 - Erro de concentração avaliado por espectrofotometria de UV-Vis.....	43
Tabela 2 - Medidas de massa de gotas para água pura em várias vazões.....	44
Tabela 3 - Valores da CMC, Γ , a e ΔG^0_{MC} para o CTABr.....	50
Tabela 4 - Valores de γ , experimentais e da literatura, em várias temperaturas.....	51

ÍNDICE DE APÊNDICES

	PAG.
APÊNDICE 1 - CONECTOR DB-25	84
APÊNDICE 2 - PROGRAMA DO TENSÍÔMETRO.....	85
APÊNDICE 3 - PROGRAMA PARA CÁLCULO DE BETA	93
APÊNDICE 4 - BIBLIOTECA DE ROTINAS.....	94
APÊNDICE 5 - MANUAL DO TENSÍÔMETRO	115

RESUMO

Esta tese de doutorado descreve o desenvolvimento de um tensiômetro automático de fluxo, baseado no método do peso da gota, no qual é realizada a variação contínua da concentração. O tensiômetro é termostatizado, a partir da utilização de uma "linha termostatizada de fluxo" e de vários adaptadores, especialmente confeccionados para essa finalidade. O instrumento é completamente gerenciado por um microcomputador, o qual realiza o controle dos periféricos, a leitura e o tratamento matemático dos dados adquiridos.

A veracidade das medidas realizadas pelo tensiômetro foi comprovada por medidas de massa da gota da água, em função da vazão; tensão superficial da água em função da temperatura; tensão superficial, CMC, e outras propriedades físico-químicas do SDS, em função da vazão; tensão superficial, CMC, e outras propriedades físico-químicas do CTABr; e medidas de concentração, por método espectrofotométrico, para fins de comparação.

Foram realizadas, também, medidas com misturas de surfactantes, e o parâmetro de interação β , de Holiand e Rubingh, foi avaliado, para cada mistura. O sistema CTABr / TTABr apresentou comportamento ideal. Os sistemas CTABr / SB3-14, TTABr / SB3-14 e DOTAB / SB3-14 apresentaram comportamento muito próximo ao ideal. Porém, os sistemas SB3-10 / SDS, SB3-12 / SDS, SB3-14 / SDS, SB3-10 / SDOS, SB3-12 / SDOS e SB3-14 / SDOS apresentaram grande interação sinérgica entre os seus componentes.

ABSTRACT

This work describes the development of an automated flow tensiometer based on the drop weight method, in which a continuous variation of concentration is carried out. In the proposed system, the tensiometer is thermostatted using a "thermostatted flow line" and several adapters specially built for this purpose. The tensiometer is totally managed by a personal computer, which controls peripheral equipment and processes the acquired data mathematically.

The following measurements were performed for verification of the accuracy of the measurements made with the tensiometer: water drop mass measurements with relation to flow rate; surface tension of the water with relation to temperature; surface tension, CMC, and other chemical and physical properties of SDS with relation to flow rate; surface tension, CMC, and other chemical and physical properties of CTABr. Concentration measurements were also performed using spectrophotometry for comparison purposes.

Measurements with several surfactant mixtures were also performed. Holland and Rubingh's β interaction parameter was assessed for each mixture. The CTABr / TTABr system presented an ideal behaviour. The CTABr / SB3-14, TTABr / SB3-14 and DOTAB / SB3-14 systems presented an almost ideal behaviour. However, high synergy was detected among the components in the SB3-10 / SDS, SB3-12 / SDS, SB3-14 / SDS, SB3-10 / SDOS, SB3-12 / SDOS and SB3-14 / SDOS systems.

SÍMBOLOS E ABREVIATURAS

g	- Aceleração da gravidade.
A/D	- Analógico - digital.
A	- Área.
a	- Área superficial molecular.
CTABr	- Brometo de cetiltrimetilamônio.
DOTAB	- Brometo de dodeciltrimetilamônio.
TTABr	- Brometo de tetradeciltrimetilamônio.
CI	- Circuito integrado.
CMC_A, CMC_B	- CMC dos componentes individuais de uma mistura.
f_A, f_B, f_N	- Coeficiente de atividade.
λ_{max}	- Comprimento de onda relativo ao máximo de absorção.
C_c, C_{c-1}	- Concentração corrigida da gota.
C_g	- Concentração da gota.
$C_a, C_b, C1$	- Concentração da solução da seringa.
C2	- Concentração inicial da solução da câmara de mistura.
CMC	- Concentração micelar crítica.
R	- Constante universal dos gases.
SB3-10	- Decil-N,N-Dimetil-3-propanosulfonato de amônio.
SDOS	- Decil-oxietilenosulfonato de sódio.
RDS	- Desvio padrão relativo.
D1	- Diodo.
LED	- Diodo emissor de luz.
SB3-12	- Dodecil-N,N-Dimetil-3-propanosulfonato de amônio.
SDS	- Dodecilsulfato de sódio.
Γ	- Excesso na superfície.
f, f_{-1}	- Fator de correção para o método do peso da gota.
Qf	- Fototransistor.
X_A, X_B, X_N	- Fração molar da forma micelizada.

Y_A, Y_B	- Fração molar da forma monomérica.
χ_A, χ_B	- Fração molar total.
L_f, L_i	- Leitura da balança.
m	- Massa.
μc	- Microcomputador.
NTC	- <i>Negative coefficient temperature.</i>
N^a	- Número de Avogadro.
n	- Número de mols de partículas por mol de surfactante.
nP_a, nP_b	- Número de passos da bomba de seringa.
β	- Parâmetro de interação micelar de Rubingh.
P	- Peso.
π	- $\pi \cong 3,14$.
rV	- Quociente entre raio efetivo e raiz cúbica do volume da gota.
r	- Raio do capilar.
R1	- Resistor.
R_L	- Resistor de carga.
T	- Temperatura.
γ	- Tensão superficial.
SB3-14	- Tetradecil-N,N-Dimetil-3-propanosulfonato de amônio.
W	- Trabalho.
Q1, Q2	- Transistor.
CPU	- Unidade Central de Processamento.
ΔG^0_{MIC}	- Variação de energia livre padrão para o processo de formação da micela.
f_{va}, f_{vb}	- Vazão da seringa.
V	- Volume.
V_m	- Volume da câmara de mistura.
V_g, V_{g-1}	- Volume da gota.
V_r	- Volume residual no capilar.
V_n, V_{n-1}	- Volume total adicionado na câmara de mistura.

Capítulo I

Introdução

**“Uma garrafa meio vazia de vinho é também meio cheia,
porém uma meia mentira de modo algum é meia verdade”
Jean Cocteau**

**“O problema de resistir a uma tentação é que
talvez não haja uma segunda chance”
Aurence Peter**

**“Se os fatos contradizem os profetas,
pior para os fatos”
Nelson Rodrigues**

1.1. Objetivos

O objetivo principal deste trabalho é construir um tensiômetro automático* baseado na determinação do peso de uma gota, que permita a caracterização de sistemas micelares. Por meio de experimentos envolvendo tensão superficial (γ), podemos determinar a concentração micelar crítica (CMC), o excesso na superfície (Γ), a área superficial molecular (a), a variação da energia livre do processo de formação das micelas (ΔG_{MIC}^0), entre outras.

1.2. Revisão da Literatura

A obtenção confiável dos dados de γ exige que o experimento seja realizado em condições de equilíbrio, sendo necessário várias horas de trabalho experimental. Em consequência disto, nos últimos anos, vários trabalhos envolvendo a montagem de sistemas automáticos para a determinação da γ têm sido publicados na literatura.⁽¹⁻¹²⁾ Hayami⁽¹⁾ estudou o efeito da variação do ângulo capilar pelo método da gota pendente. McMillan⁽²⁾ utilizando o mesmo método, desenvolveu um novo aparelho baseado em fibra ótica. Faour⁽³⁾, Lin⁽⁴⁾ e Anastasiadis⁽⁵⁾ acoplaram um processador de imagem e um computador ao seu instrumento analítico. Liggieri⁽⁶⁾ utilizou o método da pressão capilar no estudo do envelhecimento da superfície e Soos⁽⁷⁾ realizou medidas sob condições estáticas e dinâmicas. Ambos os trabalhos utilizaram transdutores de pressão. Para o método do peso da gota foram montados vários sistemas. Henderson⁽⁸⁾ e Matsuki⁽⁹⁾ desenvolveram um aparelho com boa acuracidade de temperatura e de volume de gota e estudaram o efeito da vazão. O sistema montado por Fainerman⁽¹⁰⁾ foi utilizado para estudar o efeito hidrodinâmico de formação de gota em altas vazões, e o sistema montado por Nunez-Tolin⁽¹¹⁾ permite realizar a variação contínua da concentração de forma automática. O instrumento

* Seguindo a tendência de outros autores, nesta tese não será feita distinção entre os termos "automático" e "mecanizado". Ver SKOOG, D.A.; LEARY, J.J.; *Principles of instrumental analysis*. 4ed. New York: Saunders College Publishing, 1992.

desenvolvido por Tornberg⁽¹²⁾ tem sido utilizado em trabalhos recentes da literatura. Para finalizar, Mollet⁽¹³⁾ realizou um estudo comparativo do efeito da formação *in situ* das moléculas do surfactante sobre os valores de γ e Motomura⁽¹⁴⁾ avaliou o efeito da temperatura.

Na continuação, será feita uma breve introdução a respeito de alguns aspectos relevantes deste trabalho, em relação aos surfactantes e suas propriedades e em relação às técnicas de automação.

1.3. Surfactantes

Os surfactantes são agentes tensoativos, ou seja, aumentam a molhabilidade da solução sobre um sólido.⁽¹⁵⁾ Em geral, são muito importantes devido às suas inúmeras aplicações, já que podem dissolver óleos e gorduras insolúveis em água pura⁽¹⁶⁾ e atuar como catalisadores ou inibidores no estudo cinético de reações químicas.⁽¹⁷⁻²¹⁾ As interações micelares dos agregados por eles formados podem ser estudadas termodinamicamente.^(22,23)

Surfactantes são compostos orgânicos anfífilos que possuem em sua estrutura molecular uma parte polar hidrofílica e outra apolar hidrofóbica (cadeias alquílicas, anéis aromáticos, etc.).

De acordo com a sua região hidrofílica, os surfactantes são classificados em 4 tipos.^(20,21)

- a) catiônicos: Região hidrofílica possui carga positiva [$C_{16}H_{33}N(CH_3)_3^+ Br^-$];
- b) aniônicos: Região hidrofílica possui carga negativa [$C_{12}H_{25}SO_4^- Na^+$];
- c) não-iônicos: Região hidrofílica possui grupos oxietileno ou sulfóxido [$H(-O-CH_2-CH_2)_n - R$], [$C_{12}H_{23}SOCH_3$];
- d) zwitteriônicos: Há na região hidrofílica pelo menos dois grupos com cargas opostas [$C_{12}H_{25}N^+(CH_3)_2-C_3H_6-SO_3^-$].

1.4. Concentração Micelar Crítica

As soluções aquosas de surfactantes, quando ultrapassam uma determinada concentração, a CMC (Concentração Micelar Crítica), formam micelas. Na região da CMC são observadas bruscas mudanças em várias propriedades físico-químicas,⁽¹⁵⁾ como tensão superficial, condutividade elétrica, densidade, viscosidade, turbidez e outras.

Uma micela é um agregado de vários monômeros de surfactante e possui tamanho típico de um colóide (1nm - 1 μ m).⁽²⁴⁾ Hartley⁽²⁵⁾ sugeriu uma forma esférica para as micelas, porém existem micelas das mais variadas formas, dependendo do tipo de surfactante, concentração, força iônica, solvente, temperatura, etc. A Figura 1,⁽²⁶⁾ mostra uma micela esférica, cujo interior está preenchido com óleo.

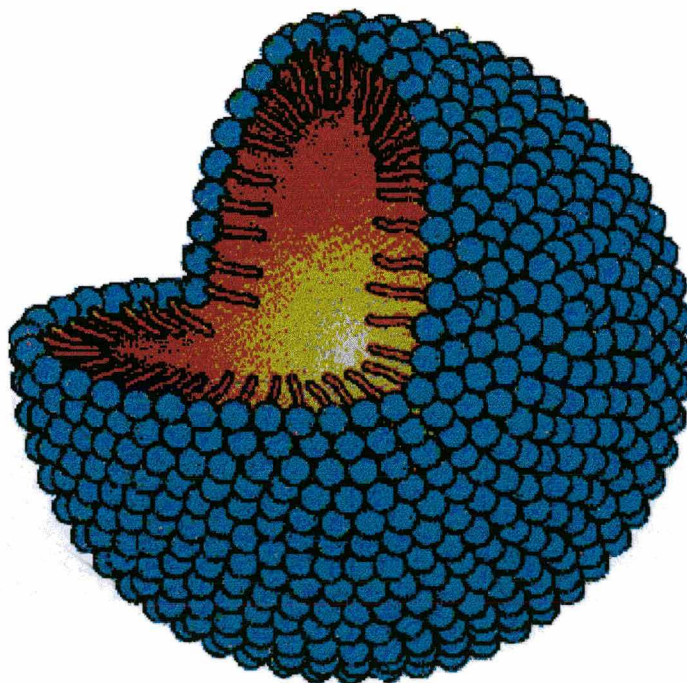




Figura 1 - Representação esquemática de uma micela esférica preenchida com óleo.

Nesta figura, a parte hidrofílica, também conhecida por grupo cabeça, está representada por , enquanto a parte hidrofóbica, também conhecida por cauda, está representada por . A CMC é a máxima concentração possível de monômeros livres em solução, ou seja, é a concentração mínima necessária para que se formem as micelas. Estes agregados são termodinamicamente estáveis, pois sua energia livre de formação é negativa.⁽²⁷⁾ A estabilidade é facilmente entendida, uma vez que, na forma de agregado, a parte hidrofóbica do surfactante está isolada da água, no interior da micela.

A quantidade de unidades monoméricas integrantes de uma micela é o número de agregação micelar, que é dependente⁽²⁸⁾ da concentração do surfactante; do comprimento da cadeia hidrocarbonada do surfactante; do tipo e estrutura do grupo hidrofílico; da natureza do contra-íon; da temperatura; da concentração e do tipo de aditivo que se encontra em solução.

As técnicas para se determinar a CMC de surfactantes são várias.⁽²⁹⁾ Dentre elas, pode-se citar tensão superficial, condutividade elétrica, densidade, turbidez, viscosidade, etc. Todas elas baseiam-se na determinação de uma propriedade físico-química qualquer, como função da concentração de surfactante.

1.5. Misturas de Surfactantes

1.5.1. Misturas Ideais

Misturas de dois ou mais surfactantes, que possuem estruturas semelhantes, apresentam características de mistura ideal⁽³⁰⁾ e podem ser estudadas com base no modelo da pseudofase.^(20,21) O equilíbrio entre monômeros e fase micelar é considerado como sendo análogo ao equilíbrio vapor-líquido em uma mistura de líquidos ideais. Sendo assim, a composição da mistura pode ser modelada utilizando-se as equações 1 e 2, onde CMC é a concentração total de monômeros, CMC_A e CMC_B são as CMC dos componentes

individuais, Y_A e Y_B são as frações molares da forma monomérica e X_A e X_B são as frações molares da forma micelizada. Os índices A e B referem-se aos surfactantes A e B, respectivamente.

$$CMC = \frac{CMC_A \cdot CMC_B}{(Y_A \cdot CMC_B + Y_B \cdot CMC_A)} \quad (1)$$

$$CMC = X_A \cdot CMC_A + X_B \cdot CMC_B \quad (2)$$

Na CMC a relação apresentada na equação 3 também é válida.

$$X_A = \frac{Y_A \cdot CMC}{CMC_A} \quad (3)$$

1.5.2. Misturas Não Ideais

Misturas de surfactantes com grupos hidrofílicos opostos, como, por exemplo, um surfactante catiônico e outro aniônico, apresentam desvios negativos em relação ao comportamento ideal devido à atração eletrostática entre os grupos hidrofílicos.⁽³¹⁾ Por outro lado, em misturas de surfactantes em que um deles possui uma cadeia hidrocarbônica e o outro uma cadeia de fluorcarboneto, o desvio em relação ao comportamento ideal é positivo.⁽³²⁾ O estudo desses tipos de misturas é realizado a partir da introdução do parâmetro ajustável β , desenvolvido por Rubingh.⁽³³⁾ Este parâmetro é adimensional e serve como um indicativo do tipo e do grau de interação entre os surfactantes.

Pelo modelo da pseudofase, a CMC de uma solução mista de surfactante é relacionada com a CMC dos componentes individuais através da equação 4,^(23,34,35) onde χ_A e χ_B são as frações molares totais e f_A e f_B são os coeficientes de atividade.

$$\frac{1}{CMC} = \frac{\chi_A}{f_A \cdot CMC_A} + \frac{\chi_B}{f_B \cdot CMC_B} \quad (4)$$

Os coeficientes de atividade são calculados a partir da equação 5, sendo que os valores de β e X_A são obtidos resolvendo iterativamente a equação 6 e aplicando o valor de X_A na equação 7.

$$f_N = \exp[\beta \cdot X_N^2] \quad (5)$$

$$X_A^2 \cdot \ln\left(\frac{\chi_A \cdot \text{CMC}}{X_A \cdot \text{CMC}_A}\right) = (1 - X_A)^2 \cdot \ln\left[\frac{\chi_B \cdot \text{CMC}}{(1 - X_A) \cdot \text{CMC}_B}\right] \quad (6)$$

$$\beta = \frac{\ln\left(\frac{\text{CMC} \cdot \chi_A}{\text{CMC}_A \cdot X_A}\right)}{X_B^2} \quad (7)$$

Esse tratamento tem sido usado como modelo para ajustar dados experimentais de CMC, de uma grande variedade de sistemas mistos, que apresentam desvios em relação ao comportamento ideal. ^(23,31,34,36-40)

1.6. Tensão Superficial

Ao se observar uma gota de um líquido qualquer caindo livremente, percebe-se que a mesma apresenta-se sob um formato curvo, aproximando-se de uma esfera. Tal fenômeno é devido à tensão superficial (γ) que é definida como sendo o trabalho (W) necessário para aumentar uma superfície em uma unidade de área (A), em um processo isotérmico e reversível. ⁽¹⁵⁾

$$\gamma = \frac{dW}{dA} \quad (8)$$

A tensão superficial pode ser explicada como sendo a diferença entre as forças de Van der Waals (ou outras mais fortes), de curto alcance, que atuam nas moléculas da superfície, e as forças que atuam nas moléculas do interior do líquido, conforme mostrado na Figura 2. ⁽²⁴⁾ No interior do líquido, as forças atuam

em todos os sentidos, e a resultante é nula. Na superfície, há uma resultante não nula que possui direção ortogonal a esse plano e cujo sentido está orientado da superfície para o interior da solução. Devido a essa resultante não nula é que a superfície tende a se curvar.

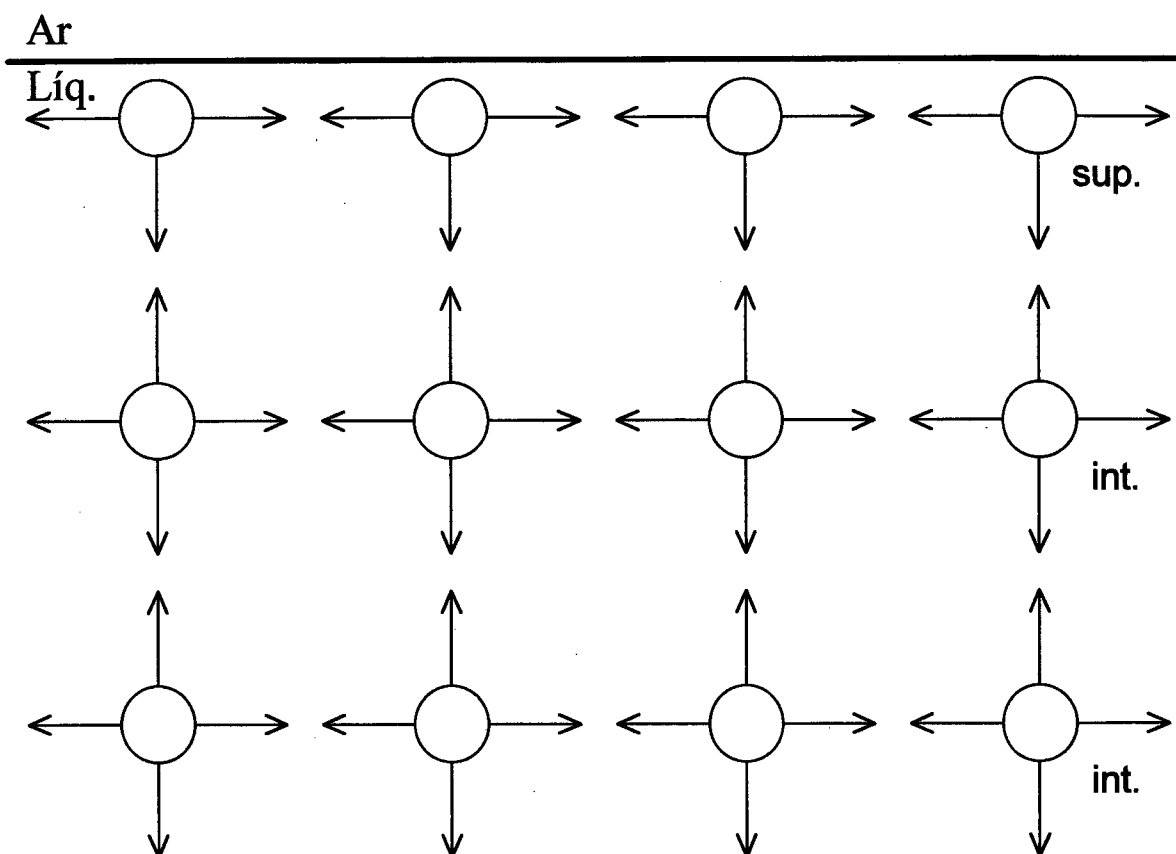


Figura 2 - Forças de atração que atuam nas partículas de um líquido. As partículas do interior (int.) possuem resultante nula e as da superfície (sup.) possuem resultante não nula.

Dentre os métodos conhecidos para determinar-se γ , o método do peso da gota é um dos mais antigos,⁽⁴¹⁾ e, sem dúvida, o mais conhecido deles. Além disso, esse método é de fácil manuseio e apresenta um baixo erro experimental ($\pm 1\%$). O método consiste em determinar a força peso (P) que atua sobre uma gota do líquido, que se encontra na superfície inferior de um tubo capilar de vidro.⁽²⁴⁾ O cálculo de γ é feito a partir da lei de Tate (Equação 9),⁽⁴¹⁾ onde m , γ e P são a massa, tensão superficial e peso da gota, respectivamente, e r é o raio

efetivo do capilar. A aceleração da gravidade é denotada por g , enquanto π é uma constante matemática ($\cong 3,14$). A lei de Tate descreve o balanço entre a força peso ($m.g$) e à força tensão superficial ($\gamma.2.\pi.r$), no exato momento em que ocorre o destacamento da gota a partir do capilar, ou seja, a força de adesão entre a gota e o capilar é igual a força gravitacional que atua sobre a gota.

$$P = m.g = 2.\pi.r.\gamma \quad (9)$$

A fórmula acima foi modificada por Harkins e Brown,⁽⁴²⁾ acrescentando-se um fator de correção empírico f , que descobriram ser função de $r/V^{1/3}$ ($f = f(r / V^{1/3})$). Na literatura é possível encontrar tabelas⁽¹⁵⁾ ou gráficos,⁽²⁴⁾ relacionando f com $r / V^{1/3}$. Lando e Oakley⁽⁴³⁾ calcularam várias funções matemáticas de f como função de $r / V^{1/3}$, cada uma válida dentro de certos limites de f .

Ao se fazer uma correlação entre γ e Log [surfactante] obtém-se uma abrupta mudança em γ , atribuído à formação das micelas (CMC), como pode ser mostrado na Figura 3.

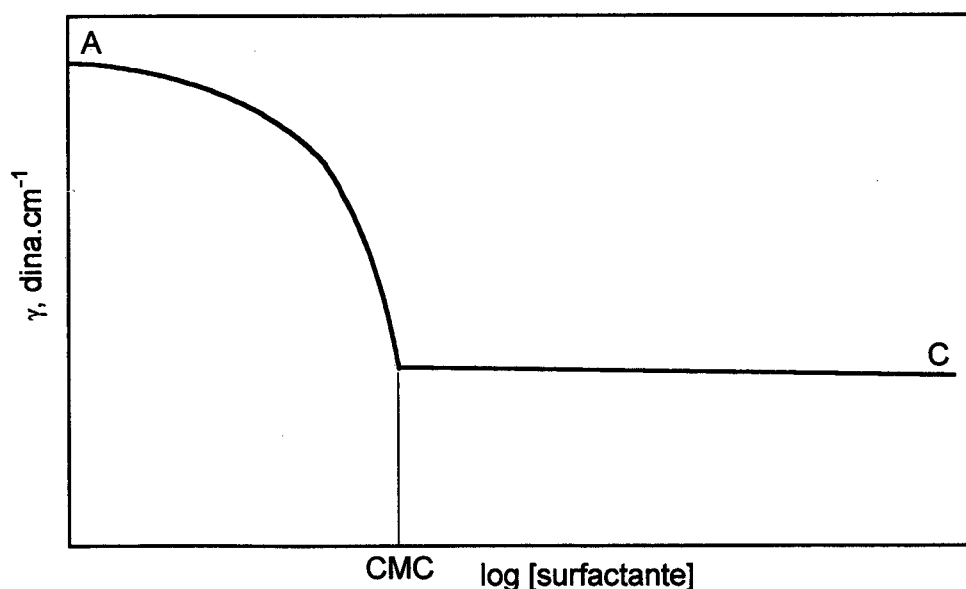


Figura 3 - Tensão superficial, γ , em função do logaritmo da concentração de surfactante.

Esse fenômeno ocorre porque a atividade superficial do surfactante é bem superior à atividade superficial da micela. Neste gráfico, do ponto A até a CMC, todo o surfactante adicionado se encontrará livre em solução ou, preferencialmente, preenchendo a superfície. Da CMC ao ponto C, todo o surfactante adicionado passará para a fase micelar, não havendo grandes mudanças no valor de γ . A determinação de γ em função da concentração de surfactante permite determinar os valores de excesso na superfície (Γ), da área superficial molecular (a) e da variação da energia livre padrão de formação das micelas (ΔG_{MIC}^0), por meio da aplicação das equações 10, 11 e 12, respectivamente. Nessas equações, R é a constante universal dos gases, t é a temperatura, n é o número de moles de partícula por mol de surfactante e N^a o número de Avogadro.

$$\Gamma = -\frac{1}{n.R.t} \left(\frac{d\gamma}{d \ln C} \right) \quad (10)$$

$$a = \frac{1}{\Gamma.N^a} \quad (11)$$

$$\Delta G_{\text{MIC}}^0 = -R.t.\ln(\text{CMC}) \quad (12)$$

Capítulo II

Automação

**"Se o cérebro fosse tão simples que nós pudéssemos entender,
então seríamos tão simples que nós não íamos entendê-lo"**
Lyll Watson

**"Na prosperidade nossos amigos nos conhecem,
mas na adversidade somos nós que os conhecemos"**
Churton Coilins

**"Nossa cabeça é redonda para permitir
ao pensamento mudar de direção"**
Francis Picabia

A automação dos procedimentos conhecidos de laboratório traz grandes vantagens. Um sistema com processamento eletrônico pode realizar várias tarefas, como movimentar bombas peristálticas⁽⁴⁴⁾ ou de seringa, realizar leituras de absorvância ou de potencial⁽⁴⁵⁻⁴⁹⁾ acionar válvulas solenóides⁽⁴⁴⁻⁵¹⁾ dentre outras. A possibilidade de utilizar um microcomputador⁽⁵²⁾ aumenta o grau de automação, tendo em vista que um programa pode tornar um sistema auto-suficiente, inclusive tomando várias decisões que, normalmente, seriam tarefas rotineiras do analista.^(48,49)

2.1. Técnicas de Fluxo

As técnicas de fluxo têm sido amplamente discutidas e aplicadas a trabalhos práticos nos últimos anos.⁽⁵³⁾ A alta frequência analítica e o uso de pouca vidraria favorecem a expansão dessa técnica e, por se tratar de um sistema fechado, é de grande importância para evitar a contaminação do meio ambiente, da amostra e do analista.^(46,47) Uma atenção especial deve ser dada à técnica de análise por injeção em fluxo (FIA), pois a mesma tem atraído o interesse de muitos pesquisadores desde a sua criação.⁽⁵³⁻⁵⁵⁾ A união das técnicas de fluxo aos sistemas microcontrolados constitui uma poderosa combinação para aqueles que necessitam de automação em laboratório.^(56,57)

2.2. Sistemas Microprocessados

Um exemplo possível de ser alcançado em um processo de automação está representado na Figura 4. Neste exemplo, o computador controla a velocidade de uma bomba peristáltica por meio da porta de comunicação serial. Um condutivímetro, um pH-metro, um espectrofotômetro, um registrador, entre outros sistemas de detecção podem ter os seus dados analógicos lidos e convertidos para dados digitais, por uma interface conversora analógico - digital (A/D), para posterior análise.

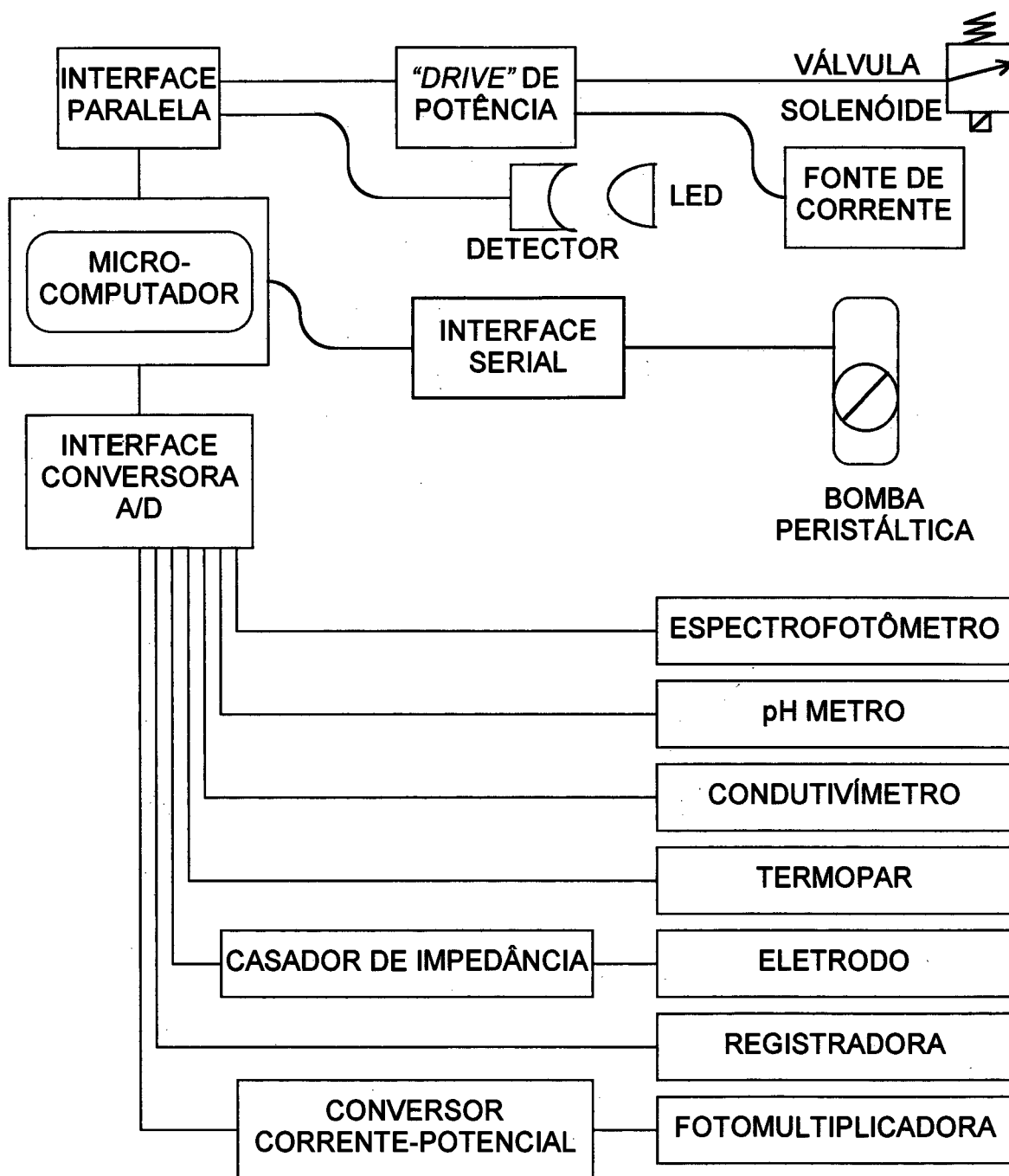


Figura 4 - Sistema microcontrolado para gerenciamento e leitura de dados.

A interface paralela pode facilmente ser utilizada como entrada, comunicando-se, por exemplo, com um detector ótico, ou como saída, acionando um "drive" de potência que, por sua vez, alimenta outros circuitos que consomem grande quantidade de energia. O gerenciamento de cada periférico descrito acima é realizado por um programa especialmente desenvolvido para tal finalidade, sendo

possível o tratamento matemático dos dados e o seu armazenamento em disco. Uma etapa pode ser incluída junto ao tratamento dos dados para realizar um teste estatístico que verifique a existência de alguma discrepância entre os diversos valores lidos. Confirmada a presença de algum dado discrepante, o mesmo é eliminado e repetido de forma automática e sem a intervenção do operador. Nessa etapa também pode ser acrescentada uma rotina de otimização de resultados que determinará os valores ótimos para vários parâmetros experimentais como concentração da amostra, vazão do fluido carregador, volume injetado, concentração de fundo e quantidade de aditivos.

Uma abordagem mais detalhada a respeito da interface paralela e da interface serial, bem como de algumas rotinas específicas de programação, será realizada a seguir, uma vez que as mesmas tiveram aplicação prática neste trabalho.

2.3. Programação

O controle do computador sobre todos os periféricos ilustrados na Figura 4 somente será possível utilizando um programa que seja capaz de acionar as devidas interfaces responsáveis pelo evento desejado. Existem várias linguagens de programação de alto nível (linguagens que o programador interpreta facilmente), dentre as quais pode-se citar :⁽⁵⁸⁾ Turbo Pascal[®], Basic[®], Fortran[®], C[®], sendo que uma grande maioria existe em várias versões para os ambientes DOS[®] e Windows 3.11/95[®]. A linguagem Fortran[®] é muito utilizada em programas que realizam extensos cálculos matemáticos, sendo o Basic[®] o mais conhecido e difundido, apesar de gerar códigos executáveis relativamente lentos. As linguagens C[®] e Turbo Pascal[®] são muito utilizadas em programas desenvolvidos para o interfaceamento computador-periféricos, como em sistemas nos quais o computador controla ou recebe sinal de algum dispositivo externo e em programas de comunicação. Ambas são linguagens que possuem uma extensa biblioteca de rotinas e funções pré-definidas, permitem fácil implementação de novas rotinas através de bibliotecas definidas pelo usuário, utilizam sintaxe de código-fonte relativamente parecida, são linguagens orientadas a objeto,⁽⁵⁸⁾ são

linguagens estruturadas, geram programas executáveis poderosos e rápidos e são muito conhecidas no meio acadêmico nas áreas de automação e de engenharia. Há ainda a possibilidade de utilizar a linguagem *Assembler*, que dentre todas é a que gera os códigos executáveis mais rápidos, mas que, por se tratar de uma linguagem de baixo nível, exige grande conhecimento do “*hardware*” e necessita de um tempo muito elevado para o desenvolvimento de um novo programa. O mais indicado é a utilização de uma linguagem de alto nível e a introdução da linguagem *Assembler* apenas nas rotinas que exigem alta velocidade de processamento.

2.4. Rotinas Normais e Rotinas de interrupção

Em processos de automação, muitas vezes faz-se necessário que determinadas ações sejam realizadas em intervalos de tempo regulares e precisos. Nesses casos, as rotinas de interrupção são ferramentas quase que obrigatórias. Rotinas de interrupção são aquelas nas quais o processamento somente será realizado após um dispositivo adequado sinalizar à unidade central de processamento (CPU). Imediatamente, esta interrompe a execução da rotina normal, salva todos os seus registradores na memória e passa a executar a rotina de interrupção, podendo esta ser natural (do próprio computador) ou desenvolvida pelo usuário em substituição à interrupção natural. Porém, em ambas as situações, após a sua conclusão, a CPU retoma os registradores previamente salvos e continua a executar a rotina normal, no mesmo ponto onde havia parado. Deve-se ter em mente que nem todas as interrupções do computador estão disponíveis para serem substituídas livremente pelo usuário. Por exemplo, se o vetor de interrupção do teclado (interrupção 9) for alterado, e na nova rotina definida pelo usuário não for executada uma chamada à rotina original, simplesmente o computador não responderá mais aos toques do teclado. Por outro lado, se a interrupção modificada for a do relógio (interrupção 8), por exemplo, não haverá atualização da data e da hora do computador. A Figura 5 ilustra o exemplo de um código-fonte completo que aciona uma rotina de

interrupção definida pelo usuário, em linguagem Turbo Pascal[®], no sistema operacional DOS[®].

```
PROGRAM EX_INTERR; (1)
USES DOS,CRT; (2)
VAR velho_vetor_de_interrupcao:POINTER; (3)
    contador:WORD; (4)
    (5)
PROCEDURE NOVA_ROTINA_DE_INTERRUPCAO; (6)
INTERRUPT; (7)
BEGIN (8)
    INC(contador); (9)
    WRITELN('Esta interrupcao foi acessada ',contador,' vezes. '); (10)
    PORT[$20]:=$20; (11)
END; (12)
    (13)
BEGIN (14)
    contador:=0; (15)
    GETINTVEC($1C,velho_vetor_de_interrupcao); (16)
    SETINTVEC($1C,@NOVA_ROTINA_DE_INTERRUPCAO); (17)
    REPEAT (18)
        WRITELN('Esta é a rotina normal que esta sendo interrompida. '); (19)
    UNTIL KEYPRESSED; (20)
    SETINTVEC($1C,velho_vetor_de_interrupcao); (21)
END. (22)
```

Figura 5 - Exemplo de uma rotina de interrupção em Turbo Pascal[®].

O programa inicia na linha 15 zerando o valor da variável “contador”. Na linha 16, a função “getintvec” salva na variável “velho_vetor_de_interrupcao”, o endereço de memória para onde a interrupção de número 28 ($28_{10} = 1C_{16}$) desloca o processamento da CPU. Na linha 17, a função “setintvec” modifica o deslocamento da interrupção para o endereço da rotina “nova_rotina_de_interrupcao”. Antes do seu término, na linha 21, o programa deve restaurar o vetor de interrupção que foi salvo na linha 16.

Uma forma de não trabalhar com interrupção e conseguir uma base de tempo confiável para executar uma determinada tarefa é utilizando avaliações “booleanas”, cujo resultado será função de alguma base de tempo. Por exemplo, o programa pode ficar esperando até que o relógio do computador seja atualizado (na frequência de 18,2065 Hz) para, então, realizar uma determinada

tarefa. O programa em linguagem Turbo Pascal[®], apresentado na Figura 6, utiliza exatamente essa forma de controle. A variável "tempo_atual" é declarada como residindo no mesmo endereço da memória (segmento \$0000 e deslocamento \$046C), onde a interrupção do relógio do computador registra a hora do sistema. Na linha 16, o valor atual da hora é armazenado na variável "tempo_inicial" para posterior comparação na linha 18. Enquanto nenhuma interrupção do relógio for executada, o programa fica em laço entre as linhas 17 e 18. Ao ser executada uma interrupção do relógio, o valor contido naquele endereço (\$0000:\$046C) de memória será alterado e o resultado da comparação realizada na linha 18 tornar-se-á verdadeiro. Então, o programa passa para a linha seguinte, executando a rotina "SIMULA_INTERRUPCAO". Um programa equivalente, em linguagem Qbasic[®], é ilustrado na Figura 7.

```
PROGRAM EXRELOG; (1)
USES CRT; (2)
VAR tempo_atual:LONGINT ABSOLUTE $0000:$046C; (3)
    tempo_inicial:LONGINT; (4)
    contador:WORD; (5)
    (6)
PROCEDURE SIMULA_INTERRUPCAO; (7)
BEGIN (8)
    INC(contador); (9)
    WRITELN('Rotina de falsa interrupcao acessada ',contador,' vezes. '); (10)
END; (11)
    (12)
BEGIN (13)
    contador:=0; (14)
    REPEAT (15)
        tempo_inicial:= tempo_atual; (16)
        REPEAT (17)
            UNTIL (tempo_atual <> tempo_inicial); (18)
        SIMULA_INTERRUPCAO; (19)
    UNTIL KEYPRESSED; (20)
END. (21)
```

Figura 6 - Rotina acessada com base no tempo em Turbo Pascal[®].


```
DEF SEG = &H0
contador = 0
DO WHILE INKEY$ = ""
    inicial = PEEK(&H46C)
    DO WHILE (inicial <> PEEK(&H46C))
        LOOP
    contador = contador + 1
    PRINT "Esta rotina foi acionada"; contador; "vezes."
LOOP
```

Figura 7 - Rotina acessada com base no tempo em Qbasic®.

2.5. Comunicação Paralela

Uma interface paralela é útil nas situações em que são realizados acionamentos ou leituras de estados lógicos. Vários exemplos podem ser citados: acionamento de motor de passo,⁽⁵⁹⁾ de válvula solenóide,⁽⁴⁴⁻⁵¹⁾ de motor dc, de lâmpadas, de fonte de corrente^(50,51) e leitura de chave ótica.^(48,49) A conexão física da interface paralela ao periférico é normalmente realizada utilizando-se um conector do tipo DB-25. A função de cada pino desse conector na interface paralela varia de um fabricante para outro, sendo necessária a utilização de um manual apropriado. O Apêndice 1 ilustra um conector desse tipo e uma listagem contendo as funções de todos os pinos de duas interfaces paralelas encontradas no comércio.^(60,61)

2.5.1. Interface Paralela como Saída

A interface paralela, quando utilizada como saída, geralmente necessita de um "drive" de potência, uma vez que a maioria dos dispositivos utilizados consomem maior energia que aquela suportada pela interface. Em termos básicos, o "drive" de potência é um transistor (Q1) operando como chave aberto-fechado que possibilita o consumo de maior energia, de acordo com o circuito representado na Figura 8A.

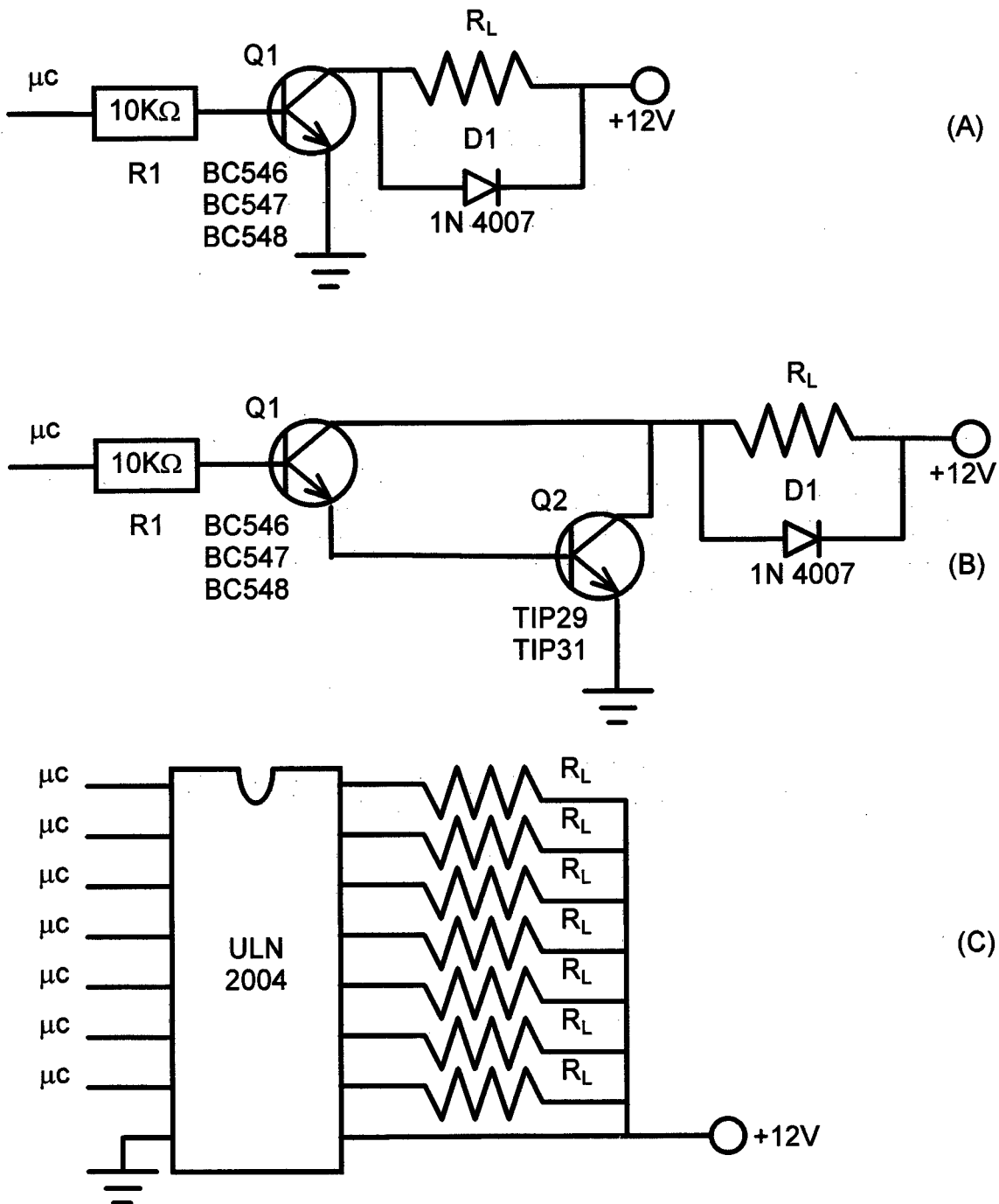


Figura 8 - Circuito eletrônico do “drive” de potência para utilização em conjunto à interface paralela. “Drive” comum (A), em configuração Darlington (B) e circuito integrado ULN 2004 (C). Os símbolos μC representam as saídas da interface paralela do computador.

O resistor R1 limita a corrente de base em Q1, pois a mesma não pode ser superior a 2 mA,^(62,63) garantindo a sua integridade. Sendo a carga R_L do tipo indutiva (motor de passo, válvula solenóide, relê, etc.), é obrigatória a presença do diodo D1 que, nesse caso, é conhecido por “diodo roda-livre”. Esse circuito suporta até 100 mA e, havendo necessidade de maior consumo de energia, um outro circuito pode ser utilizado, de acordo com a representação na Figura 8B. A configuração Darlington apresentada suporta uma carga de 3 A. Um circuito mais compacto e que possui 7 “drives” iguais aos descritos anteriormente, suportando até 500 mA cada um, encontra-se representado no circuito integrado ULN 2004 da Figura 8C. Neste exemplo, cada “drive” representado possui internamente um diodo roda-livre.

O acionamento de qualquer um desses circuitos e, conseqüentemente, da carga é obtido fazendo-se uma equivalência direta entre os bits de 0 a 7 da interface paralela e os dígitos de ordem 1 a 8 de um valor numérico de saída (em binário). Esse valor numérico deve conter o dígito 1 (um) correspondendo aos bits da interface, cujos pinos no conector DB-25 serão alimentados com +5V (nível lógico alto) e o dígito 0 (zero) correspondendo aos bits, cujos pinos não serão alimentados (nível lógico baixo). Na Figura 9 está representado um exemplo de acionamento dos pinos 3 e 6 do conector DB-25 (bit 1 e bit 4) da interface paralela da impressora, nas linguagens Turbo Pascal[®], Qbasic[®] e Assembler. No exemplo dado, o programa em Assembler funciona corretamente em qualquer processador baseado no 8086, como o 80286, o 80386, o 80486, etc.

Uma atenção especial deve ser dada à base numérica utilizada, pois os números 18_{10} , 12_{16} e 00010010_2 , representam o mesmo valor. O endereço da memória \$0040:\$0008 contém o número da porta lógica da impressora paralela. Sendo assim, a declaração para a variável ADDRLPT1 em Turbo Pascal[®] e as 4 linhas iniciais do Qbasic[®] e do Assembler para o processador 8086 têm exatamente essa função: resgatar o endereço da porta de saída da interface paralela. Outra interface paralela pode ser utilizada em adição a essa e, nesse caso, o endereço de saída, bem como uma eventual programação adicional da interface, deve ser obtido com o próprio fabricante através dos manuais. Em

qualquer caso, os comandos utilizados para escrita na interface paralela são os mesmos, ou seja:

- "PORT[endereço da porta paralela]:=código de saída;", em Turbo Pascal®;
- "OUT (endereço da porta paralela), código de saída", em Qbasic®;
- "OUT endereço, saída", em Assembler.

```
PROGRAM SAIDALPT;  
VAR addript1:WORD ABSOLUTE $0040:$0008;  
BEGIN  
    PORT[addrlpt1]:=18;  
END.
```

ROTINA EM
TURBO PASCAL®

```
DEF SEG = (&H40)  
lowaddr = PEEK(&H08)  
hiaddr = PEEK(&H09)  
addript1 = lowaddr + hiaddr*16*16  
OUT (addrlpt1),18
```

ROTINA EM
QBASIC®

```
MOV SI,40  
MOV ES,SI  
MOV AX,ES:[0008]  
MOV DX,AX  
MOV AX,12h  
OUT DX,AX
```

ROTINA EM
ASSEMBLER PARA
O PROCESSADOR
8086

Figura 9 - Programa para acionamento da interface paralela da impressora.

2.5.2. Interface Paralela como Entrada

A utilização da interface paralela como entrada, a partir de um detetor ótico Qf (fototransistor), pode ser conseguida montando-se o circuito ilustrado na Figura 10. A introdução de um obstáculo como um objeto físico opaco ou uma gota de um líquido qualquer entre o LED e Qf provocará a interrupção da passagem de luz. Assim sendo, o fototransistor estará em corte e o sinal na

entrada da interface passará para alto (+5V), situação que pode facilmente ser reconhecida por qualquer programa.

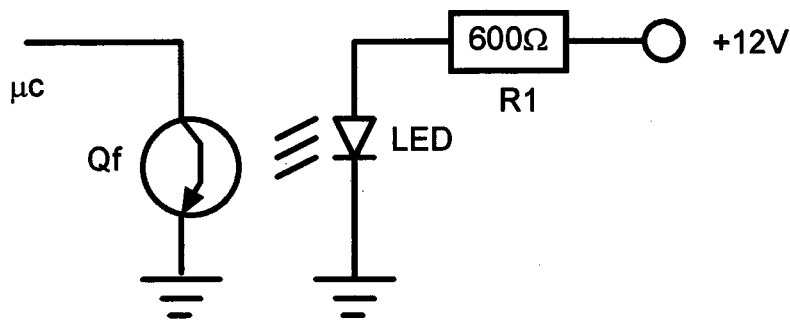


Figura 10 - Circuito para leitura de um detetor ótico.

A Figura 11 ilustra 3 programas que verificam a entrada da porta paralela da impressora.

```
PROGRAM ENTRALPT;
VAR  addr1pt1:WORD ABSOLUTE $0040:$0008;
     entrada:byte;
BEGIN
  entrada:=PORT[addr1pt1+1];
  Writeln(entrada);
END.
```

ROTINA EM
TURBO PASCAL®

```
DEF SEG = (&H40)
lowaddr = PEEK(&H08)
hiaddr = PEEK(&H09)
addr1pt1 = lowaddr + hiaddr*16*16
entrada=INP (addr1pt1+1)
```

ROTINA EM
QBASIC®

```
MOV SI,$40
MOV ES,SI
MOV AX,ES:[0008]
INC AX
MOV DX,AX
IN AL,DX
MOV entrada,AL
```

ROTINA EM
ASSEMBLER PARA
O PROCESSADOR
8086

Figura 11 - Programas para leitura da interface paralela da impressora.

Observa-se que a porta lógica de saída somada a 1 (um) fornece o valor da porta lógica de entrada. Novamente, a rotina em Assembler funciona corretamente em qualquer processador baseado no 8086.

É fácil perceber uma grande similaridade com o exemplo anterior, sendo o algoritmo representado por:

- “código de entrada:=PORT[endereço da porta paralela];”, em Turbo Pascal®;
- “código de entrada=INP(endereço da porta paralela)”, em Qbasic®;
- “IN entrada, endereço”, em Assembler.

2.6. Comunicação Serial

O sistema de comunicação serial é geralmente utilizado para transferência de dados e exige um circuito bem mais complexo que os apresentados até o momento. Normalmente, essa comunicação é realizada pela interface serial RS-232C. Em geral, os equipamentos (balança, espectrofotômetro, pH-metro, etc.) possuem em seu interior todo o circuito necessário para a comunicação serial. No Apêndice 1, apresentam-se as funções de cada saída em um conector DB-25, utilizado para comunicação serial e a sua equivalência em um conector de 9 pinos.

O acionamento da porta serial do computador exige programação específica do controlador de interrupção que, na maioria dos computadores pessoais, é o circuito integrado (CI) 8259. Também é necessária a elaboração de um protocolo de comunicação, que conterà informações diversas a respeito da comunicação estabelecida, entre as quais podem-se citar: velocidade de transmissão de dados, porta lógica de comunicação e tamanho do dado transmitido. A linguagem Qbasic® possui o comando “OPEN” para implementar o protocolo e preparar o CI 8259, porém a linguagem Turbo Pascal® não possui um comando equivalente e a montagem do protocolo deve ser feita por uma rotina do usuário. As Figuras 12, 13 e 14 apresentam um programa-exemplo elaborado em Turbo Pascal®, que estabelece comunicação serial de leitura entre o computador e um periférico qualquer, utilizando a 2ª porta serial com uma taxa de transmissão

de 1200 bauds (bits por segundo). Esse programa pode ser utilizado para leitura de dados a partir de qualquer equipamento que utilize uma interface serial RS-232C.

```
program serial;
uses crt,dos;
var dadostr:string;
    pr:word;
    save0c:pointer;
    save8259,com:byte;
    buffer:array[0..500] of byte;
const com1 = 1;
        com2 = 2;
        base : array [1..2] of word = ($03f8,$02f8);
        vect : array [1..2] of byte = ($0c,$0B);
        mask : array [1..2] of byte = ($EF,$F7);
procedure serial_recepcao;
interrupt;
var i:integer;
begin
    buffer[pr]:=port[base[com]];
    buffer[pr+1]:=0;
    if (buffer[pr] = 10) and (buffer[pr-1] = 13) then begin
        for i:=1 to pr-1 do DadoStr[i]:=chr(buffer[i]);
        DadoStr[0]:=chr(pr-2);
        pr:=0;
    end;
    inc(pr);
    if (pr>450) then pr:=0;
    port[$20]:=$20;
end;
```

Figura 12- Rotina de interrupção do programa de leitura da interface serial.

```
procedure serial_protocolo (com,baud:word;paridade:char;stop,palavra:byte);
var controle:byte;
    reg:registers;
begin
  case baud of
    110:controle:= 0;
    150:controle:= 1;
    360:controle:= 2;
    600:controle:= 3;
    1200:controle:= 4;
    2400:controle:= 5;
    4800:controle:= 6;
    9600:controle:= 7;
  end;
  case upcase(paridade) of
    'N' : controle:=(controle shl 2) or 0;
    'O' : controle:=(controle shl 2) or 2;
    'E' : controle:=(controle shl 2) or 3;
  end;
  case stop of
    1 : controle:=(controle shl 1) or 0;
    2 : controle:=(controle shl 1) or 1;
  end;
  case palavra of
    7 :controle:=(controle shl 2) or 2;
    8 :controle:=(controle shl 2) or 3;
  end;
  reg.al:=Controle;
  reg.ah:=0;
  reg.dx:=com-1;
  intr($14,reg);
end;
```

Figura 13- Rotina de preparação do protocolo para comunicação serial.


```
procedure habilita_reg_serial(com:word);
begin
  port[base[com]+4] := $0f;
  port[base[com]+1] := $01;
  port[$21] := port[$21] and Mask[Com];
end;
procedure inicia_comunicacao(porta,bau:word;par:char;stop,pal:byte);
begin
  dadostr:="";
  Com:=Porta;
  pr:=1;
  serial_protocolo(com,bau,par,stop,pal);
  getintvec(vect[com],save0c);
  save8259:=port[$21];
  setintvec(vect[com],@serial_recepcao);
  habilita_reg_serial(com);
end;
procedure termina_Comunicacao;
begin
  setintvec(Vect[com],save0c);
  port[$21]:=save8259;
end;
begin
  inicia_comunicacao(com2,1200,'n',1,8);
  repeat
    writeln(dadostr);
  until keypressed;
  termina_comunicacao;
end.
```

Figura 14 - Programa de leitura da interface serial (COM2) em 1200 bauds.

Capítulo III

Experimental

**“Não há nenhum vento favorável para aquele
que não sabe o porto a se dirigir”
Arthur Schopenhauer**

**“O passado é uma lição para se meditar,
não para se reproduzir”
Mário de Andrade**

**“No fim , tudo dá certo. Se não deu,
é porque ainda não chegou ao fim”
Fernando Sabino**

3.1. Reagentes

As sulfobetaínas decil-N,N-dimetil-3-amônio-1-propanosulfonato (SB3-10), dodecil-N,N-dimetil-3-amônio-1-propanosulfonato (SB3-12) e tetradecil-N,N-dimetil-3-amônio-1-propanosulfonato (SB3-14), com pureza de 98% e procedentes da Sigma, foram purificadas através de refluxo em extrator do tipo "soxhlet", utilizando como solvente éter de petróleo na fração 40° (Merck), por um período de 48 horas. Os surfactantes dodecilsulfato de sódio (SDS), brometo de cetiltrimetilamônio (CTABr), brometo de tetradeciltrimetilamônio (TTABr) e brometo de dodeciltrimetilamônio (DOTAB), com pureza de 99%, comprados da Sigma, foram utilizados sem nenhum tratamento prévio. O surfactante decil-oxietileno-2-sulfonato de sódio (SDOS) foi cedido pelo Dr. Hugo Gallardo,⁽⁶⁴⁾ da Universidade Federal de Santa Catarina. A água utilizada no preparo de todas as soluções foi destilada e purificada por um sistema de troca iônica do tipo MILLI-Q Water System, cuja condutividade específica era menor que $0,1 \mu\text{S}\cdot\text{cm}^{-1}$. O corante 1-o-toluil-azo-2-naftol (ORANGE-OT) foi gentilmente cedido pela Dra. Iolanda Cuccovia,⁽⁶⁵⁾ da Universidade de São Paulo.

3.2. Soluções

As soluções estoques de todos os surfactantes utilizados nas medidas de tensão superficial foram preparadas em uma concentração de 5 a 8 vezes o valor da CMC dos mesmos. As soluções estoques que continham uma mistura de dois surfactantes foram obtidas a partir da mistura de alíquotas adequadas das soluções estoques dos surfactantes puros, de forma que a concentração total da mistura se encontrasse na mesma faixa acima descrita, com relação à CMC da mistura. Em nenhuma dessas soluções foi feita adição de sal ou tampão.

3.3. Equipamentos

O computador utilizado na automação do tensiômetro foi um IBM-compatível 386 DX-40 com 4 MB de RAM. Todos os programas desenvolvidos foram escritos em linguagem Turbo Pascal[®], versão 7.0⁽⁵⁸⁾. A balança utilizada para medir a massa das gotas era dotada de uma interface serial RS-232C (Chyo, modelo JL-180, precisão 0,0001g). A vazão foi mantida constante por uma bomba de seringa (Microquímica, modelo MQBSG1/302), o controle de bomba de seringa e a leitura do detetor ótico foram realizados por uma interface de entrada / saída paralela (Microquímica, modelo MQI 12/8pcc). Um banho termocriostático (Microquímica, modelo MQBTZ99 - 20), cuja precisão é de $\pm 0,1^{\circ}\text{C}$, foi usado na termostatização do sistema, sendo que um agitador magnético (Corning) foi utilizado para a homogeneização da câmara de mistura. Tubos de tygon, de polietileno e de silicone e peças de acrílico e de alumínio foram utilizados para confeccionar o sistema de termostatização. Um termômetro de mercúrio, um NTC de 3 mm, um multímetro comercial e uma bomba peristáltica, desenvolvida no próprio laboratório, foram também utilizados para calibrar o sistema de termostatização. Um espectrofotômetro de UV-Vis (Hewlett - Packard 8452^A) foi utilizado para analisar possíveis erros de concentração.

3.4. Descrição do Tensiômetro

O diagrama esquemático do tensiômetro desenvolvido inicialmente está ilustrado na Figura 15. Controladas por um microcomputador e acionadas por uma interface paralela, duas bombas de seringa com motor de passo são movimentadas com velocidades variadas. O programa para controle e acesso das entradas e saídas digitais da interface obedeceu às instruções contidas no respectivo manual.⁽⁶⁰⁾

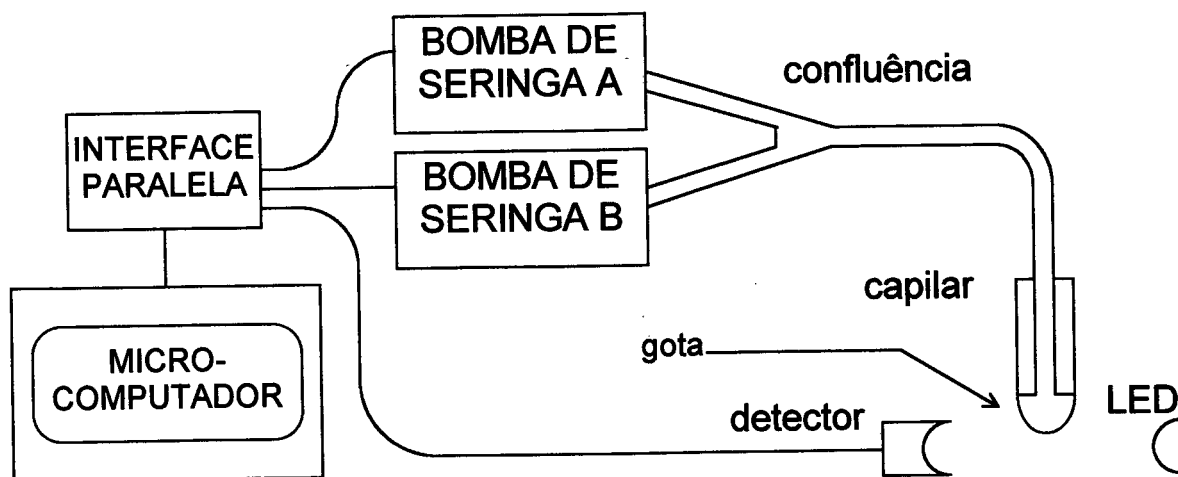


Figura 15 - Representação esquemática do tensiômetro automatizado com duas bombas de seringa.

Na confluência, as duas soluções são misturadas. No capilar de vidro, a gota é formada e, ao ocorrer o seu desprendimento, esta interromperá um feixe de luz emergente de um LED que incide sobre o detetor ótico. Nesse momento, um sinal é enviado ao computador, que efetua os cálculos de concentração e volume desta gota, utilizando para isso as Equações 13 e 14,

$$C_g = \frac{(C_a \cdot nP_a + C_b \cdot nP_b)}{(nP_a + nP_b)} \quad (13)$$

$$V_g = nP_a \cdot f_{va} + nP_b \cdot f_{vb} \quad (14)$$

onde C_a e C_b correspondem às concentrações do surfactante nas seringas das bombas A e B e nP_a e nP_b aos passos dados pelos motores de passo das bombas A e B, respectivamente. C_g e V_g são a concentração e o volume da gota, respectivamente; f_{va} e f_{vb} são dois fatores de proporcionalidade intrínsecos de cada seringa utilizada. f_{va} e f_{vb} têm dimensão de volume de solução / passo do motor.

Os testes realizados com esse sistema para 27 medidas consecutivas de água pura apresentaram tensão superficial igual a $(72,37 \pm 4,95)$ dina.cm⁻¹. Em virtude do alto valor do desvio padrão foi desenvolvido o tensiômetro ilustrado na

Figura 16, que utiliza uma bomba de seringa, uma câmara de mistura e, para determinar a massa da gota, uma balança.

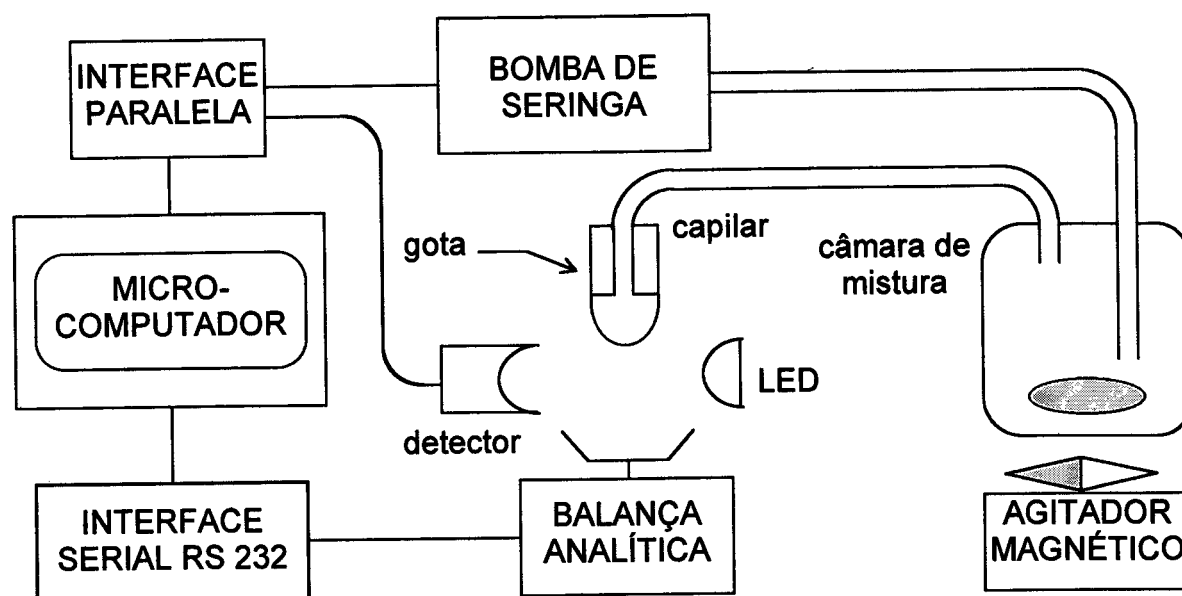


Figura 16 - Representação esquemática do tensiômetro automático.

O computador controla a velocidade da bomba de seringa e realiza as leituras do detetor ótico e dos dados enviados pela balança por meio da interface serial RS-232C.⁽⁶⁶⁾ A comunicação do computador com a bomba de seringa é realizada por uma interface de entrada e saída (interface paralela).⁽⁶⁰⁾ A seringa injeta uma solução de concentração C_1 em uma câmara de mistura, de concentração inicial C_2 e volume V_m . Um agitador magnético é responsável pela homogeneização do sistema.

A massa da gota (m_g), formada na superfície inferior do capilar, será a diferença entre as leituras da balança após e antes a passagem da gota pelo detetor ótico ($m_g = L_f - L_i$). O volume da gota (V_g) é calculado por meio da densidade (d), que é função da sua concentração ($d = f(C_g)$). Uma função matemática de segundo grau pode ser ajustada pelo programa,⁽⁶⁷⁾ sendo necessário que o usuário digite uma tabela de dados de concentração e densidade da solução. A concentração da gota é calculada através da equação 15,^(11,68,69)

$$C_g = C_1 + \frac{V_m}{V_g} \cdot (C_1 - C_2) \cdot \left[\exp\left(\frac{-V_n}{V_m}\right) - \exp\left(\frac{-V_{n-1}}{V_m}\right) \right] \quad (15)$$

onde V_n e V_{n-1} são o volume total adicionado até a gota atual e anterior, respectivamente.

3.5. Cálculos Matemáticos

Em ambos os sistemas propostos, os cálculos matemáticos abaixo foram implementados. A tensão superficial de cada gota é calculada usando a equação 9. O fator de correção é calculado pela equação 16^(11,42,43,70,71) quando rV encontra-se no intervalo entre 0,65 e 0,95 ou pela equação 17^(42,43,70,71) quando o intervalo é de 0,30 a 1,20, onde rV é obtido pela equação 18.

$$f = 0,9267 - 0,7694 \times rV + 0,4512 \times rV^2 \quad (16)$$

$$f = 0,9054 - 0,7249 \times rV + 0,4293 \times rV^2 \quad (17)$$

$$rV = \frac{r}{V_g^{1/3}} \quad (18)$$

A concentração real da gota é afetada pelo resíduo da gota anterior, ou seja, pela porção da gota que permanece no capilar após a queda da última gota. O volume residual V_r é calculado pela equação 19 e a concentração corrigida (C_c) pela equação 20, onde f_{-1} é o fator de correção da gota anterior, V_{g-1} é o volume da gota anterior e C_{c-1} é a concentração corrigida da gota anterior.

$$V_r = \frac{V_{g-1} \cdot (1 - f_{-1})}{f_{-1}} \quad (19)$$

$$C_c = \frac{C_{c-1} \cdot V_r + C_g \cdot [(V_g/f) - V_r]}{V_g} \cdot f \quad (20)$$

Ao final do experimento, o programa inicia a etapa de análise dos dados, a qual fornece os valores de CMC além de Γ , a e ΔG_{MIC}^0 .

3.6. Programa de Controle, Aquisição e Tratamento de Dados

Foi desenvolvido um programa específico, em linguagem Turbo Pascal[®] versão 7.0,⁽⁵⁸⁾ que realiza o controle da velocidade da bomba de seringa, leitura do detetor ótico, leitura dos dados enviados pela balança, apresentação do gráfico de γ x Log [Surfactante], tratamento matemático e armazenamento dos dados em disco. O código fonte desse programa encontra-se no Apêndice 2 e o fluxograma está ilustrado na Figura 17. A determinação dos valores de β foi realizada utilizando um programa próprio, cujo código-fonte encontra-se no Apêndice 3. Ambos os programas utilizam uma ampla biblioteca de rotinas e funções próprias que encontra-se no Apêndice 4. O Apêndice 5 contém um manual de operações onde os passos para realizar os principais procedimentos estão descritos.

O programa utiliza uma ampla gama de parâmetros experimentais, o que aumenta a versatilidade do tensiômetro. Dois experimentos, um realizado a 0° de latitude (linha do equador) e outro a 90° (pólos) utilizariam como valor da aceleração da gravidade⁽⁷²⁾ 9,78039 e 9,83217 m.s^{-2} , respectivamente. Solução aquosa de NaCl 0,103 mol.dm^{-3} , solução aquosa de KBr 0,127 mol.dm^{-3} e água pura apresentam densidade relativa de 1,0043, 1,0108 e 1,0000 g.cm^{-3} , respectivamente. A diferença na aceleração da gravidade é de 0,52%, na densidade da solução de NaCl é de 0,43% e na densidade da solução de KBr é de 1,08%, em relação à água pura. A utilização dos valores corretos para densidade e aceleração da gravidade, aceitos como parâmetros experimentais, permite fazer medidas absolutas nos mais variados sistemas químicos e ambientes geográficos.

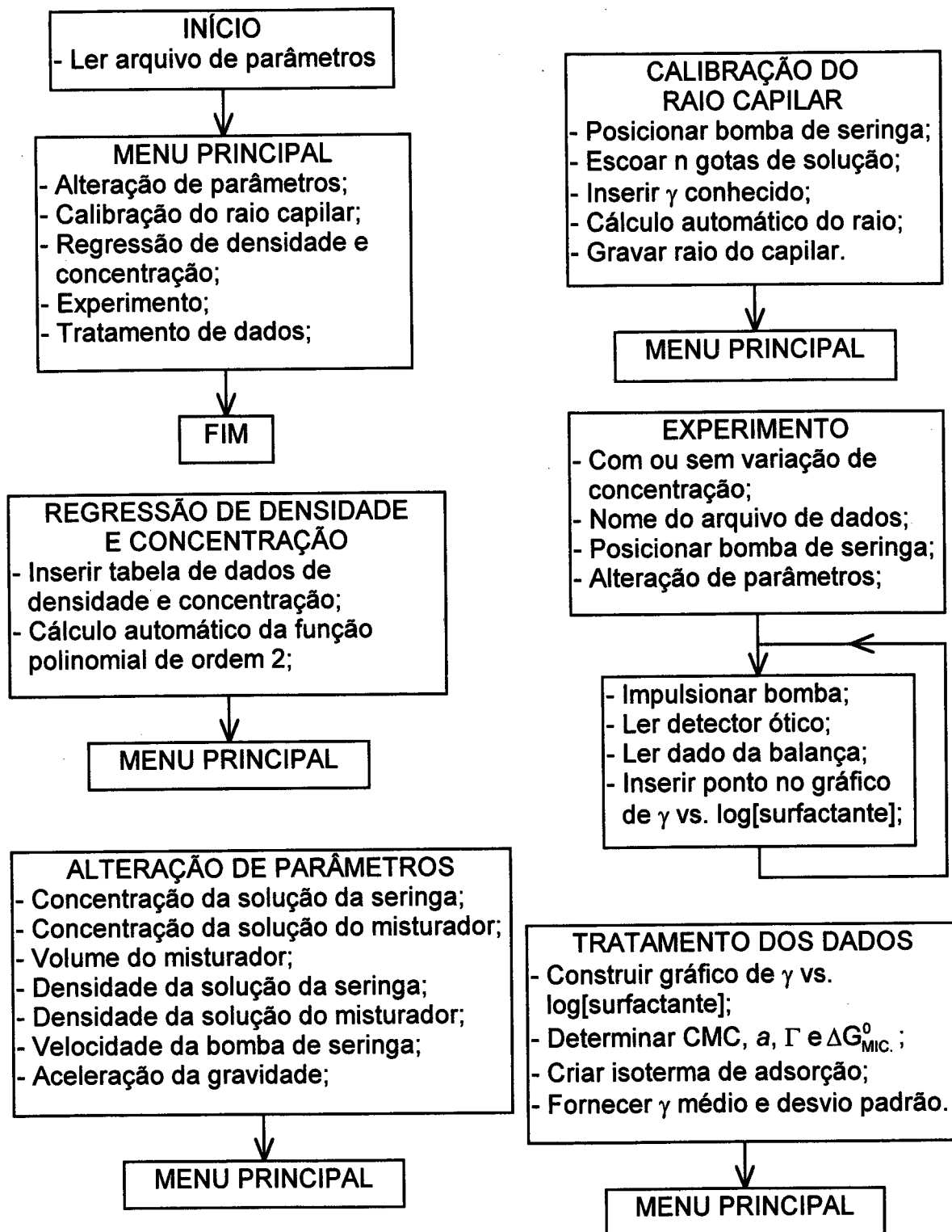


Figura 17 - Fluxograma do programa para o tensiômetro.

Os parâmetros que devem ser introduzidos são :

- vazão da bomba de seringa;
- concentração da solução da seringa;
- concentração inicial da solução da câmara de mistura;
- volume da câmara de mistura;
- raio do capilar;
- temperatura experimental;
- densidade da solução da seringa;
- densidade inicial da solução da câmara de mistura;
- aceleração da gravidade no local do experimento;
- variação da densidade da solução com a concentração.

O último parâmetro citado, ou seja, a variação da densidade da solução como função da concentração, somente poderá ser aplicado com a introdução prévia de uma tabela de dados de densidade e concentração. Por meio dessa tabela, o programa realiza uma regressão polinomial de ordem 2,⁽⁶⁷⁾ para calcular a função matemática que melhor corrige a densidade da gota.

Uma rotina para calibrar o raio do capilar com uma solução, cuja tensão superficial seja conhecida, também está presente no programa. O programa sugere o valor da tensão superficial da água pura, se essa for a substância escolhida, já corrigido o efeito da temperatura experimental.

A etapa de análise dos dados envolve várias rotinas. Inicialmente, o gráfico de tensão superficial em função de Log [surfactante] é construído. O usuário insere, muda a inclinação e posiciona adequadamente até 5 curvas, sobrepondo-as aos pontos experimentais. Uma delas deve ser inclinada para baixo e ficará sobreposta aos pontos anteriores à CMC e outra normalmente será horizontal e ficará sobreposta aos pontos posteriores à CMC. Nesse instante, pode-se determinar a CMC, Γ , a e ΔG^0_{MIC} . A construção de uma isoterma de adsorção é realizada a partir da escolha do ponto final da isoterma (próximo à CMC) e da determinação da ordem da regressão polinomial dos dados. Nos experimentos em que não há mudança na concentração, o programa calcula o valor médio da tensão superficial e o desvio padrão da média. Além disso, é realizado o teste de Smirnov-Grubbs⁽⁷³⁾ que detecta e elimina automaticamente os

valores experimentais discrepantes, com um grau de confiança de 95%. Esse teste também é aplicado na etapa de calibração do raio do capilar. Na maioria dos experimentos realizados, nenhum dado foi eliminado. Como será visto adiante, o desvio padrão sempre foi inferior a 0,5%.

3.7. Sistema de Termostatização

Inicialmente, foi montada uma linha flexível de fluxo termostatizado,⁽⁷⁴⁾ cuja confecção utiliza dois adaptadores iguais aos descritos na Figura 18, um deles atuando como entrada de fluxo e outro como saída. Todas as conexões são mantidas por pressão, não havendo necessidade de utilizar cola.

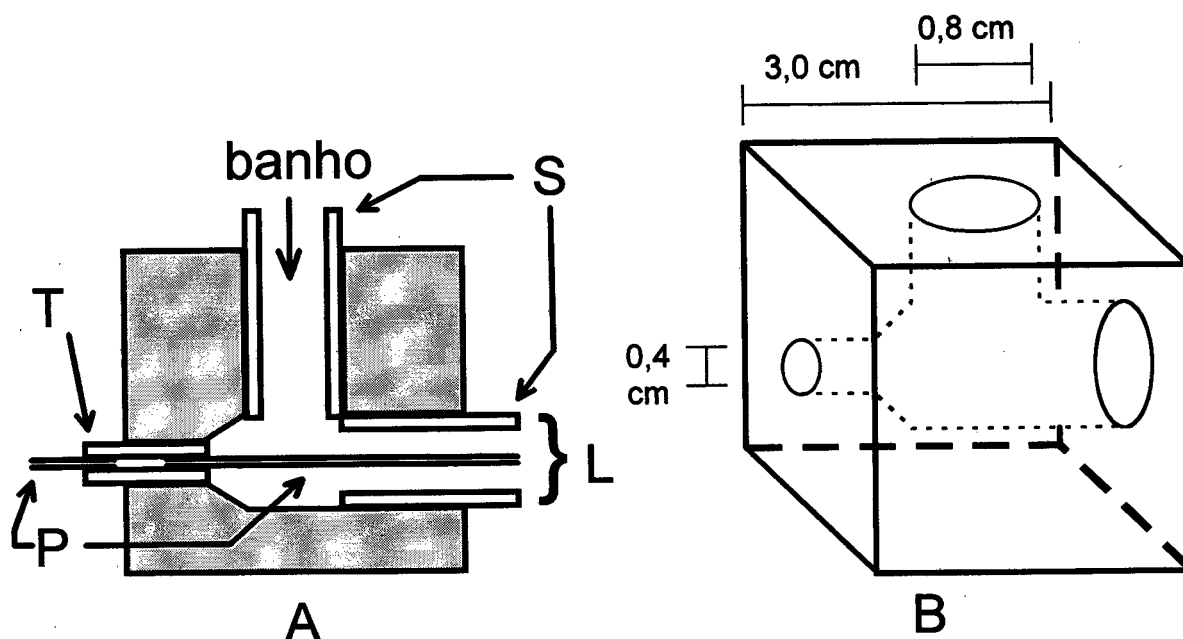


Figura 18 - Adaptador de acrílico da linha termostatizada (L). A figura da esquerda está em corte (A) e a da direita em perspectiva (B). Tubos de silicone (S), de polietileno (P) e de tygon (T).

O teste de eficiência da referida linha de fluxo foi realizado com o sistema descrito na Figura 19.⁽⁷⁴⁾ Um NTC (*negative temperature coefficient*) de 3 mm foi utilizado como termômetro (T2) no final da linha de fluxo. O mesmo foi

previamente calibrado com o termômetro de mercúrio (T1), na faixa de temperatura entre 8 e 42 °C, como apresentado na Figura 20. Medidas de temperatura da água (amostra) em T2 e em T1 servem como parâmetro de eficiência. Finalmente, a termostatização do tensiômetro é feita por um sistema totalmente adaptado (Figura 21) e mais complexo do que o exposto anteriormente.

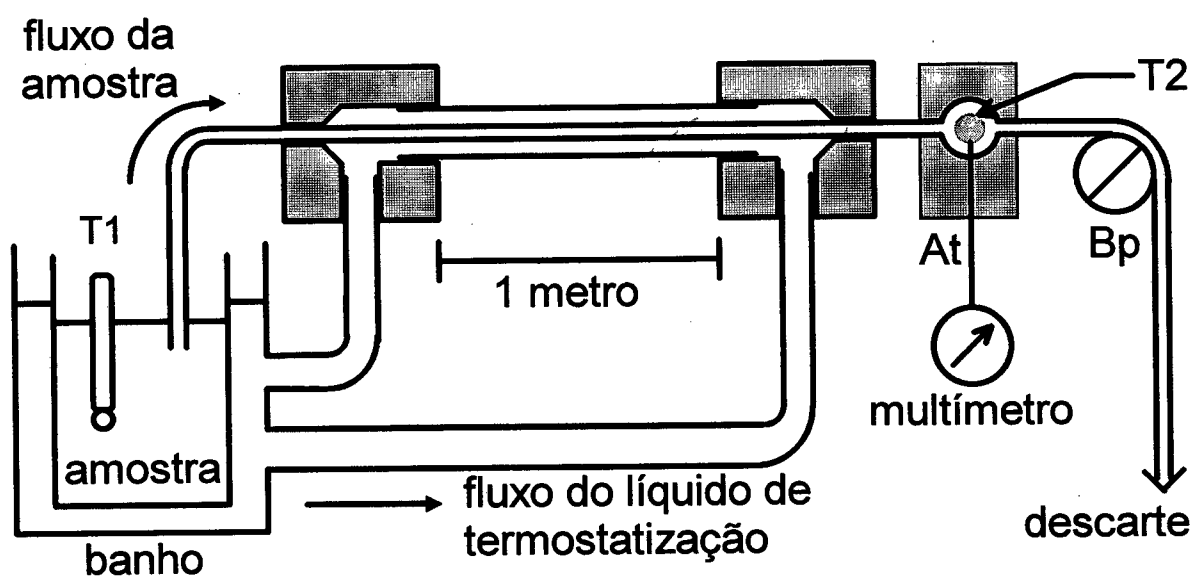


Figura 19 - Sistema de termostatização para testar a linha de fluxo. Termômetro de mercúrio (T1), termômetro baseado em NTC (T2), bomba peristáltica (Bp) e adaptador de acrílico do NTC (At).

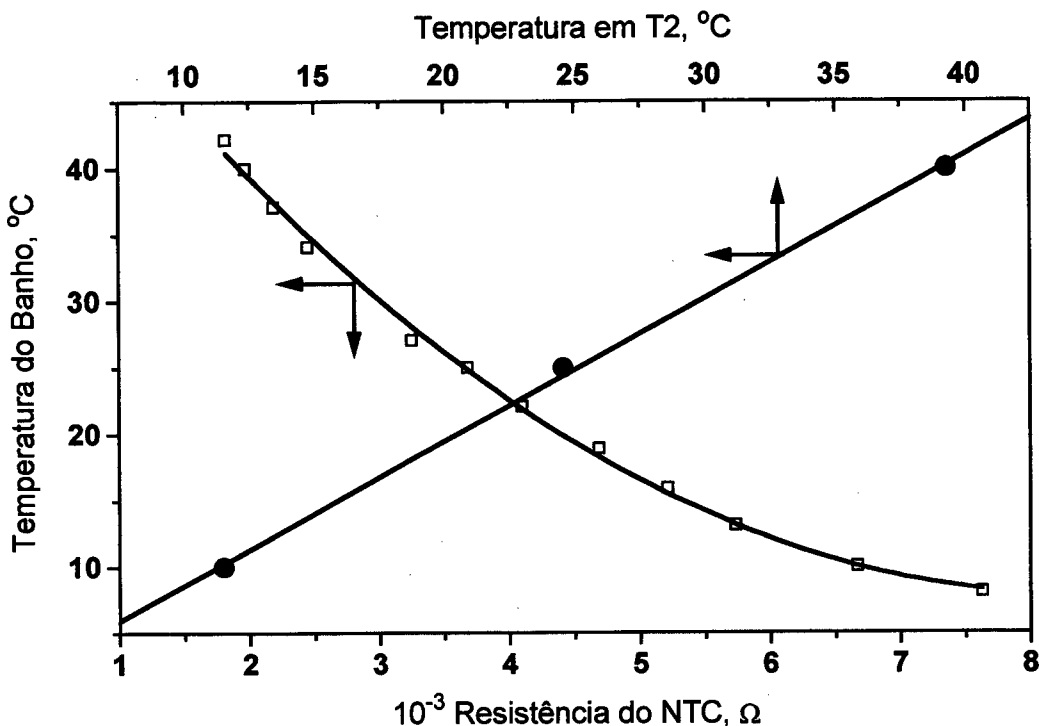


Figura 20 - Curva de calibração do NTC com o termômetro de mercúrio (T1)(□). Relação entre a temperatura do banho (T1) e a temperatura em T2 (●).

Nessa mesma versão do tensiômetro, a câmara de mistura está imersa em uma cela termostatizada de vidro e a seringa está no interior de uma camisa de vidro (CS). A cela termostatizada de alumínio permite um rápido equilíbrio térmico com a região onde há a formação da gota, é utilizada como encaixe para o capilar de vidro e acomoda em sua região inferior o LED e o detetor ótico. O orifício horizontal permite a visualização do capilar e de todo o processo de formação da gota. Os adaptadores de acrílico e as linhas termostatizadas (LT) são os mesmos apresentados nas figuras 18 e 19, respectivamente. Um dos adaptadores de acrílico possui um capilar de vidro em substituição ao tubo de polietileno, na sua parte inferior. A Figura 22 apresenta uma visão completa do sistema de controle e de fluxo do tensiômetro.

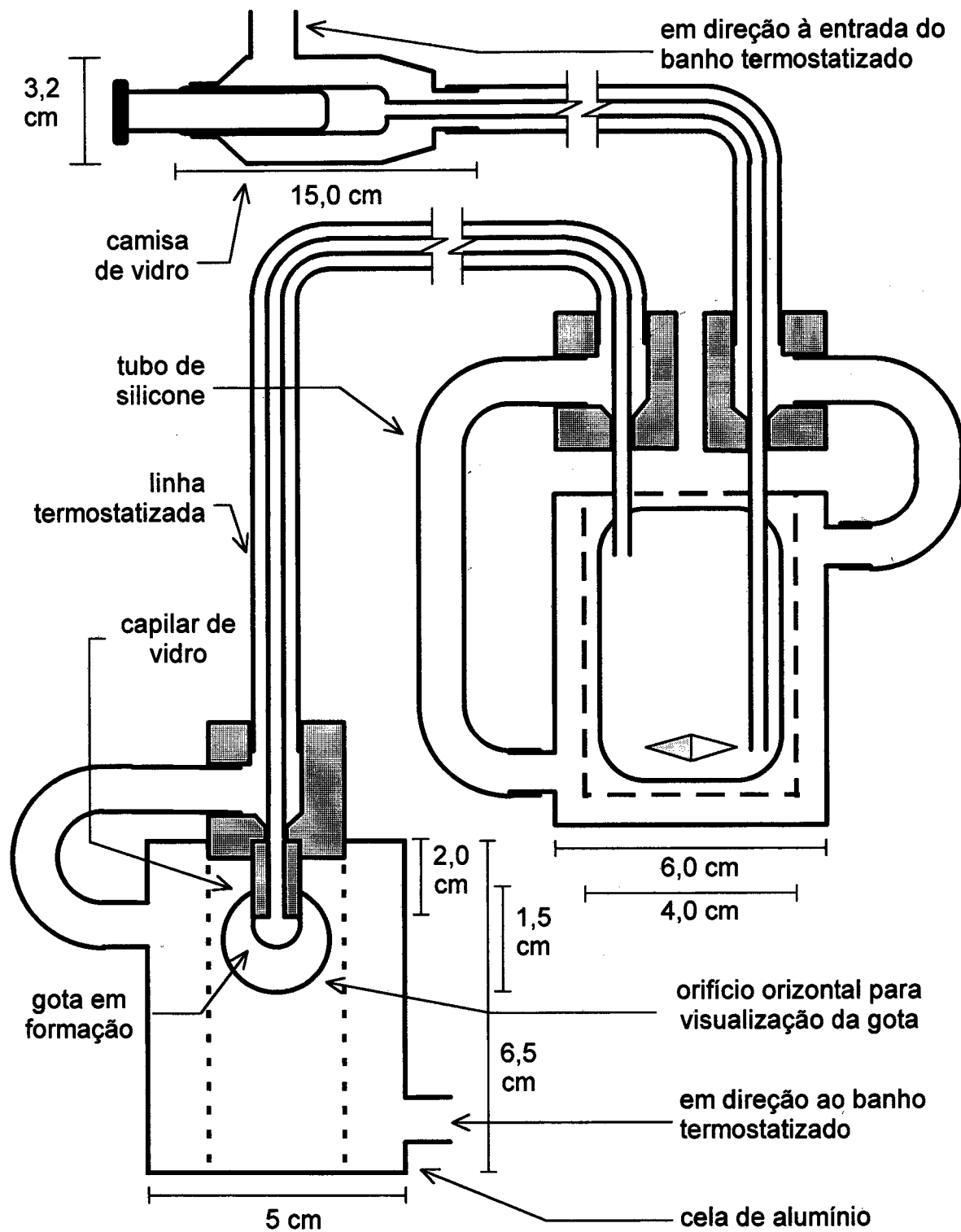


Figura 21 - Sistema de termostatização para o tensiômetro.

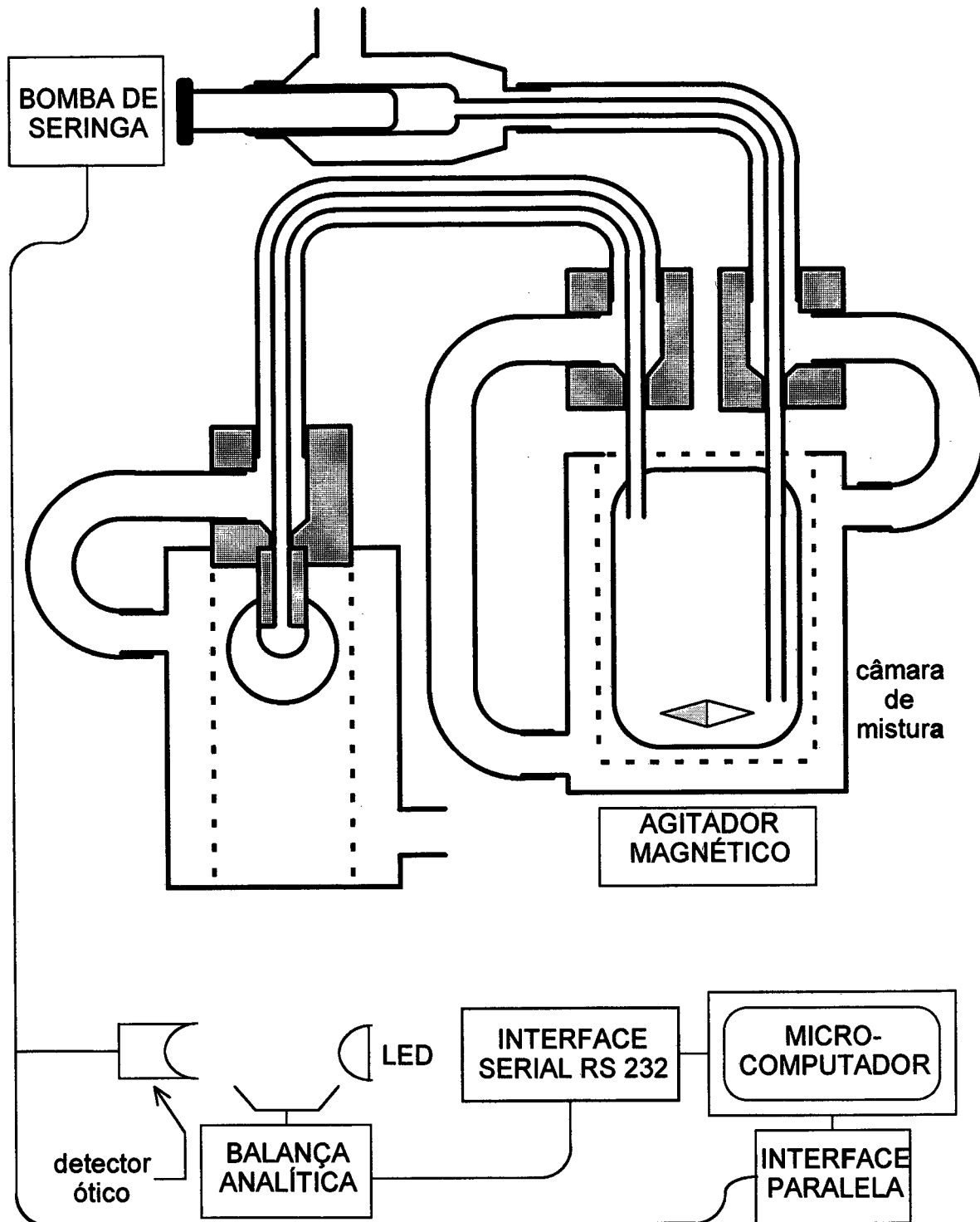


Figura 22 - Sistema completo para medidas de tensão superficial.

3.8. Determinação do Raio do Capilar

O raio do capilar utilizado em todas as medidas foi calibrado de forma freqüente, utilizando sempre uma vazão de $12 \cdot 10^{-3} \text{ mL} \cdot \text{min}^{-1}$ e temperatura da água igual a 25°C . Apenas com finalidade ilustrativa, são apresentados a seguir os resultados experimentais obtidos em uma das calibrações. O número de gotas utilizadas para calibração foi 16, a média da massa das gotas foi $(84,37 \pm 0,14) \text{ mg}$ (RDS=0,17%) e o valor do raio efetivo calculado para essa medida foi de $(0,2992 \pm 6 \cdot 10^{-4}) \text{ cm}$ (RDS=0,2%). Os erros na determinação da massa da gota e do raio efetivo do capilar determinam a precisão máxima possível de atingir com o tensiômetro construído.

Capítulo IV

Resultados e Discussões

“Se estou certo da minha teoria, por que motivos haveria de perder tempo ouvindo uma outra pessoa que, por ter idéias diferentes, tem de estar errada?”

Rubem Alves

“A consciência é parecida com o mar. Por mais violenta que seja a tempestade à superfície, o fundo permanece calmo”

Victor Hugo

“Todos nascemos louco. Alguns continuam sempre assim”

Samuel Beckett

4.1. Comprovação do Sistema de Tensão Superficial

O tensiômetro foi otimizado em relação aos diferentes parâmetros que influenciam o cálculo da tensão superficial e da CMC, isto é, massa da gota em função da vazão do líquido, temperatura da gota e cálculo de concentrações nas misturas formadas.

4.1.1. Erro no Cálculo da Concentração

A metodologia do cálculo da concentração foi avaliada com medidas de espectrofotometria de UV-Vis, utilizando uma solução do corante hidrofóbico Orange-OT ($\lambda_{\text{max}} = 495 \text{ nm}$). Dois experimentos, com variação de concentração, foram realizados e as absorbâncias das soluções finais, contidas nas câmaras de mistura, foram medidas e comparadas com aquelas obtidas a partir de uma curva analítica preparada para essa finalidade. A Tabela 1 apresenta os resultados experimentais obtidos. As concentrações calculadas pelo programa, utilizando a equação 15 descrita anteriormente, e considerando os volumes da câmara de mistura, são praticamente idênticas àquelas calculadas por meio do método espectrofotométrico.

Tabela 1 - Erro de concentração avaliado por espectrofotometria de UV-Vis.

Concentração ^a calculada pelo programa, mol.dm ⁻³	Concentração determinada por espectrofotometria, mol.dm ⁻³	Número de gotas	Volume da câmara de mistura, mL	erro, %
0,0644	0,0646	120	4,9000	0,3
0,0331	0,0333	275	17,0000	0,6

a) Vazão de $18 \cdot 10^{-3} \text{ mL} \cdot \text{min}^{-1}$.

Deve-se ter em mente que a equação 15, que descreve o processo de diluição que ocorre com a solução contida na câmara de mistura, somente será verdadeira em baixas vazões, uma vez que em vazões mais elevadas a solução

não se encontrará homogênea. Os testes aqui realizados garantem que a vazão escolhida é adequada para manter a veracidade da equação.

4.1.2. Massa da Gota de Água em Função da Vazão

Os valores encontrados para a massa da gota de água em função da vazão, a 25°C, estão apresentados na Tabela 2. Os mesmos dados desta tabela se encontram em forma gráfica na Figura 23.

Tabela 2 - Medidas de massa de gotas para água pura em várias vazões.

10^3 vazão, mL.min ⁻¹	massa, mg	desvio absoluto	desvio relativo
300	85,94	0,11	0,13
240	85,92	0,08	0,09
180	85,68	0,08	0,09
120	85,42	0,18	0,21
60	85,28	0,04	0,05
48	85,32	0,13	0,15
36	85,14	0,15	0,18
30	85,10	0,07	0,08
24	85,12	0,11	0,13
18	84,98	0,11	0,13
12	84,78	0,04	0,05
6	84,76	0,09	0,11

Cada resultado apresentado na tabela é a média aritmética da leitura de 5 gotas diferentes. Em vazões que variam de $6 \cdot 10^{-3}$ a $300 \cdot 10^{-3}$ mL.min⁻¹ (17 segundos/gota a 14 minutos/gota), observa-se que o máximo do desvio absoluto das medidas de massa da gota é de 0,18 mg, o que representa um desvio

percentual de 0,21%. Nas medidas de líquidos puros, ou seja, quando não se tratar de misturas, esse erro pode ser reduzido aumentando-se o número de replicatas. Em concordância com os resultados de Pierson e Whitaker,⁽⁷⁵⁾ a massa da gota diminui à medida que a vazão diminui, provavelmente como consequência da formação de gotas satélites de menor tamanho, ou seja, de gotas pequenas associadas à gota principal. Alguns trabalhos da literatura^(8,76-78) introduzem uma equação empírica para corrigir a massa da gota em função da vazão em estudos realizados em altas vazões. Uma rotina para realizar tal correção poderia ser facilmente implementada. Entretanto, neste trabalho procurou-se realizar medidas em condições de equilíbrio, ou seja, a vazões baixas, com soluções aquosas de surfactantes, sendo desnecessária a correção.

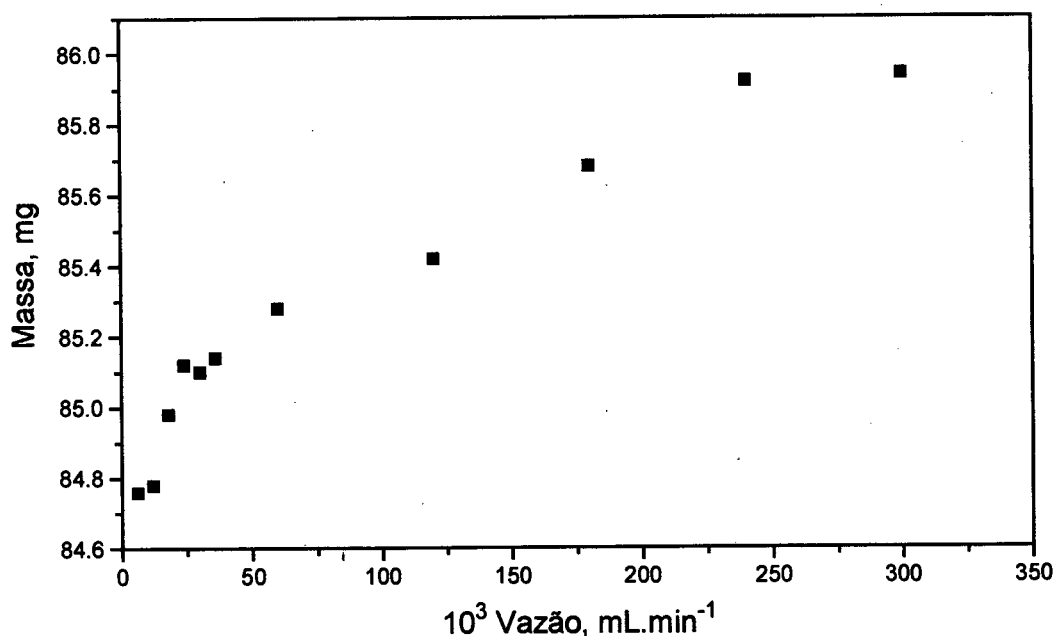


Figura 23 - Massa da gota de água em função da vazão, a 25°C.

4.1.3. Tensão Superficial do SDS em Várias Vazões

Várias determinações de tensão superficial em função da concentração do SDS, em vazões diferentes, foram realizados a 25°C e os resultados estão apresentados nas Figuras 24 e 25.

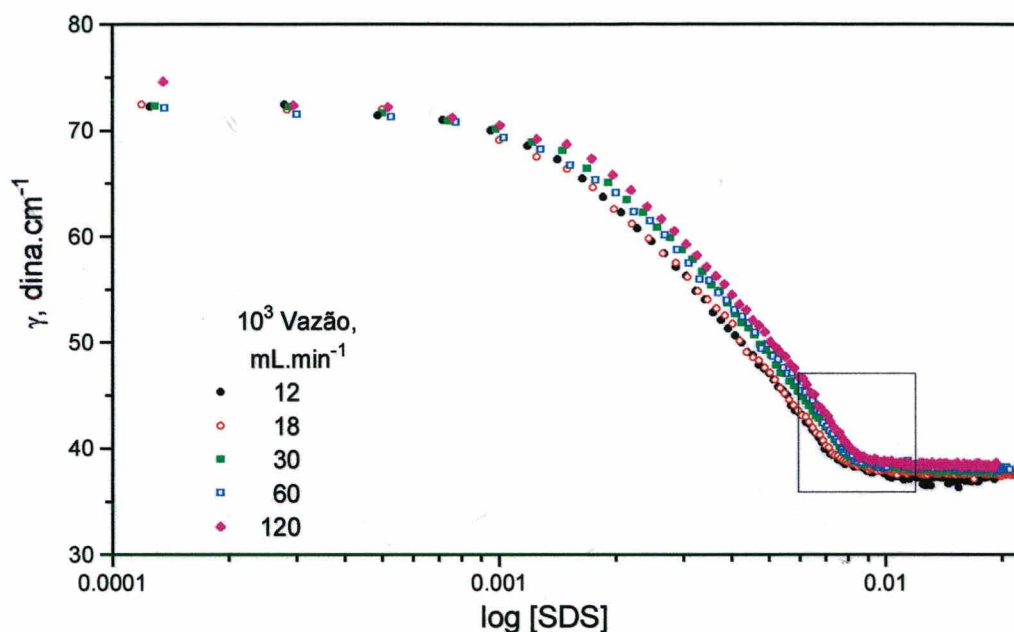


Figura 24 - Tensão superficial em função da concentração do SDS em diferentes vazões, a 25°C. A região marcada está ampliada na figura 25.

A partir da observação das figuras 24 e 25, é fácil concluir que para uma concentração constante de SDS a tensão superficial é maior para vazões mais elevadas. Uma possível justificativa é que em vazões elevadas a quantidade de gotas satélites associadas à gota principal é maior e, portanto, a sua massa será mais elevada. Outra importante observação é que o valor da CMC varia drasticamente em função da vazão aplicada, de tal forma que a CMC calculada é maior para vazões mais elevadas. Novamente, esse efeito pode ser explicado com base no envelhecimento da gota, porém não pode ser descartada a consideração a respeito da dificuldade na homogeneização da solução contida na câmara de mistura em altas vazões.^(11,68,69) Sendo os experimentos realizados

de forma a aumentar a concentração na câmara de mistura, um atraso na homogeneização faz com que a solução emergente da câmara de mistura tenha uma concentração inferior aquela prevista, e nesse caso o ponto de CMC somente será detectado em concentrações mais elevadas.

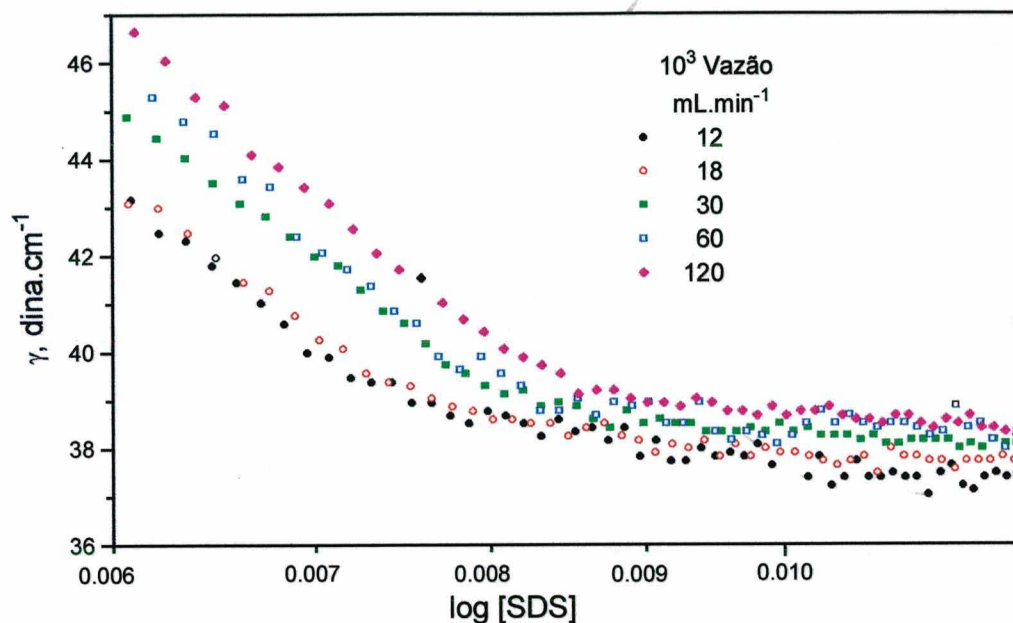


Figura 25 - Ampliação da região da CMC do gráfico de tensão superficial em função da concentração do SDS em diferentes vazões, a 25°C.

As Figuras 26 e 27 ilustram a discrepância que pode ser obtida nos resultados experimentais, caso não seja utilizada uma vazão adequada para a realização das medidas. Em vazões mais baixas, os valores de Γ e de a tornam-se concordantes com aqueles apresentados pela literatura.^(11,29,79-82) Uma atenção especial deve ser dada às curvas de vazão 0,012 e 0,018 mL.min⁻¹ (Figura 25), uma vez que os dois perfis são praticamente sobreponíveis. Existe uma dificuldade para a determinação gráfica dos valores de CMC e Γ , uma vez que uma grande variação de tangentes pode ser adotada a partir da curva de tensão superficial em função da concentração.⁽²⁹⁾

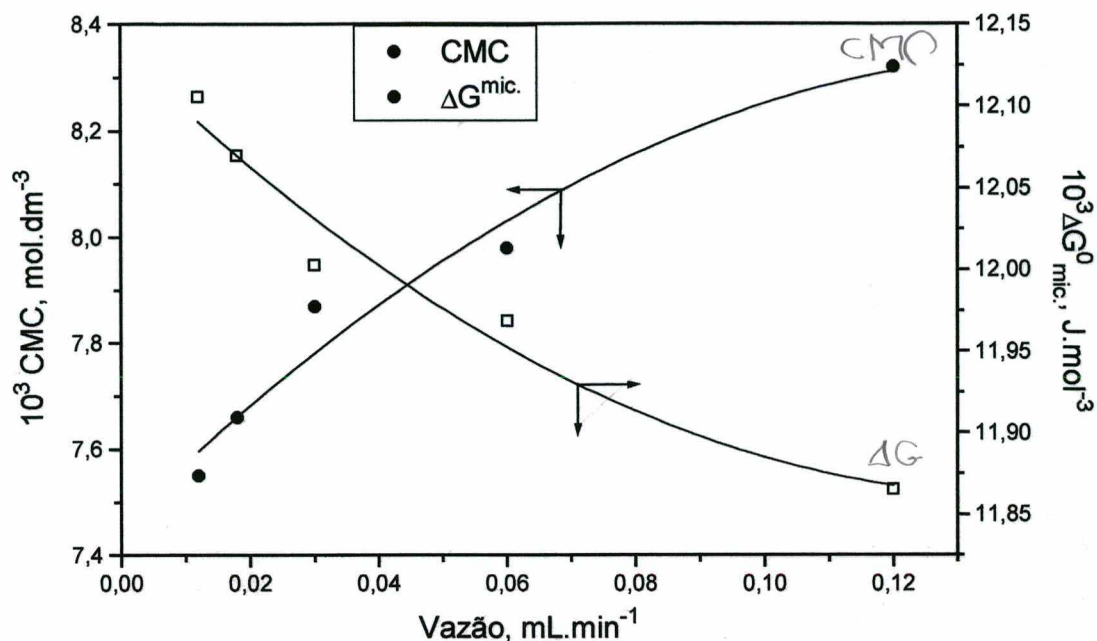


Figura 26 - Valores da CMC e da variação da energia livre do processo de formação das micelas (ΔG^0_{mic}) do SDS em função da vazão.

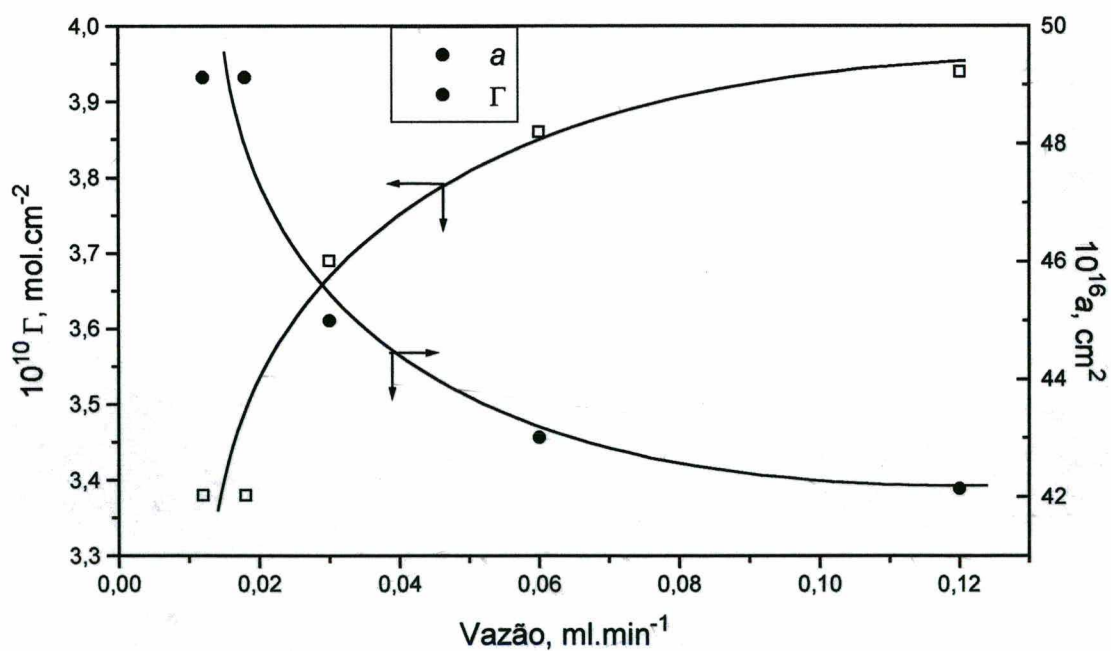


Figura 27 - Valores do excesso na superfície (Γ) e da área superficial molecular (a) do SDS em função da vazão.

Uma forma de diminuir as incertezas é a utilização de uma rotina específica que produza uma função matemática que obedeça o comportamento da curva, a partir de uma regressão polinomial dos dados experimentais. Os valores de $d\gamma / d\ln C$, ou seja, os coeficientes angulares das curvas, podem ser obtidos utilizando a 1ª derivada desta função e, finalmente, a isoterma de adsorção pode ser construída, a partir desses dados.

4.1.4. Tensão Superficial e Outras Propriedade do CTABr

A Figura 28 apresenta um gráfico de tensão superficial do CTABr em função da concentração. O experimento foi realizado em condições ótimas para os parâmetros vazão ($12 \cdot 10^{-3} \text{ mL} \cdot \text{min}^{-1}$), volume da câmara de mistura (17,00 mL) e concentração da solução na seringa ($4 \cdot 10^{-3} \text{ mol} \cdot \text{dm}^{-3}$).

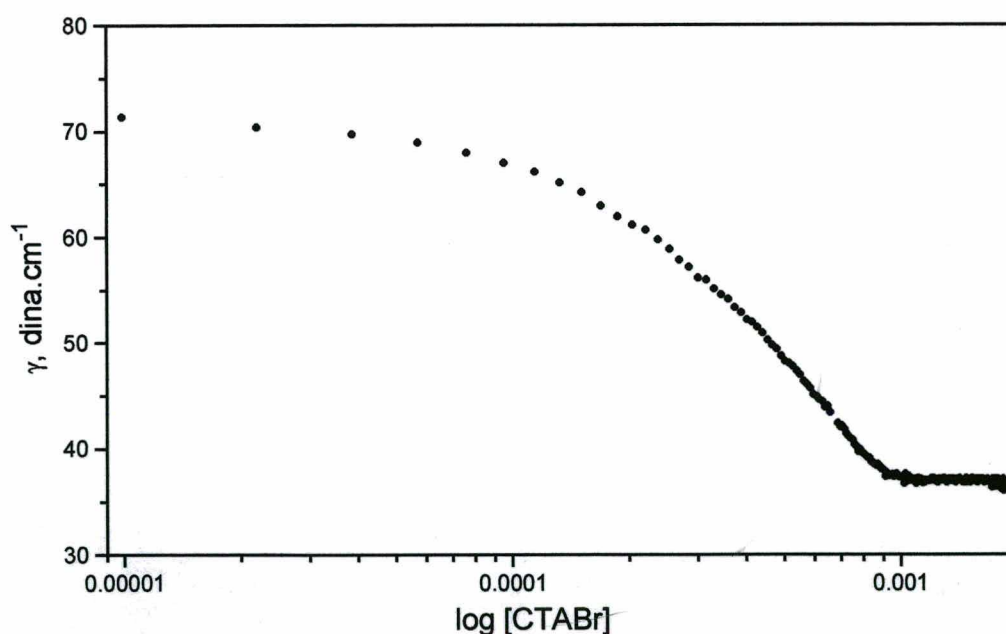


Figura 28 - Tensão superficial do CTABr em função da concentração, a 25°C. A vazão, o volume da câmara de mistura e a concentração da solução da seringa foram $12 \cdot 10^{-3} \text{ mL} \cdot \text{min}^{-1}$, 17,00 mL e $4 \cdot 10^{-3} \text{ mol} \cdot \text{dm}^{-3}$, respectivamente.

A representação gráfica obtida é excelente, havendo uma boa distribuição dos pontos experimentais anteriores e posteriores à CMC. Os valores de Γ , a , CMC e ΔG^0_{MIC} obtidos a partir da análise deste gráfico encontram-se na Tabela 3, todos concordantes com a literatura.^(11,29,83) Foram necessários 15 minutos para preparar o equipamento e 4 horas para a aquisição de dados. Durante o processo de aquisição de dados, a presença do pesquisador não se faz necessária e, portanto, outras atividades podem ser desenvolvidas simultaneamente.

Tabela 3 - Valores da CMC, Γ , a e ΔG^0_{MIC} para o CTABr.

Autor	10^4 CMC, mol.dm ⁻³	10^{10} Γ , mol.cm ⁻²	10^{16} a , cm ²	ΔG^0_{MIC} , Kj.mol ⁻¹
Este trabalho	9,3 ± 0,2	3,84 ± 0,04	43,29 ± 0,51	-27,16 ± 0,04
Paredes ⁽¹¹⁾	9,02 ± 0,03	3,28 ± 0,04	50,46 ± 0,61	-27,33 ± 0,03
Mukerjee ⁽²⁹⁾	9,20 ± 0,09	-	-	-
Zana ⁽⁸³⁾	9,0	-	-	-

4.1.5. Medidas de Tensão Superficial em Várias Temperaturas

Foram realizadas determinações de tensão superficial para água pura em temperaturas diferentes, para comprovar o sistema de termostatização completo. Com finalidade comparativa, na Tabela 4 encontram-se os resultados obtido, bem como aqueles da literatura.⁽⁷²⁾ A equação [$\gamma_{LIT.} = -15,90 + 1,218 \cdot \gamma_{EXP.}$] representa a relação entre esses valores, com um coeficiente de correlação (r) de 0,990. Ainda utilizando os mesmos dados, pode-se obter a equação [$T_{selec.} = -66,38 + 1,228 \cdot T_{real}$] que relaciona a temperatura selecionada no sistema de termostatização com a temperatura real da gota, determinada pelo valor da γ . Na Figura 29, essa mesma relação é encontrada em forma gráfica.

Tabela 4 - Valores de γ , experimentais e da literatura, em várias temperaturas.

Temperatura, K	γ literatura, dina.cm ⁻¹	γ experimental, dina.cm ⁻¹
283,0	74,22	73,86
285,5	-	73,71
288,0	73,49	73,30
290,5	-	73,14
291,0	73,05	-
293,0	72,75	72,94
295,5	-	72,68
298,0	71,97	72,25
300,5	-	71,69
303,0	71,18	71,36
308,0	-	70,85
313,0	69,56	-

Observa-se portanto, que a cela de alumínio (Figura 22) e todo o restante do sistema de termostatização, apesar de não ser um sistema ideal, é um sistema de termostatização bastante eficiente, pois a sua ausência modificaria significativamente o resultado das medidas realizadas.

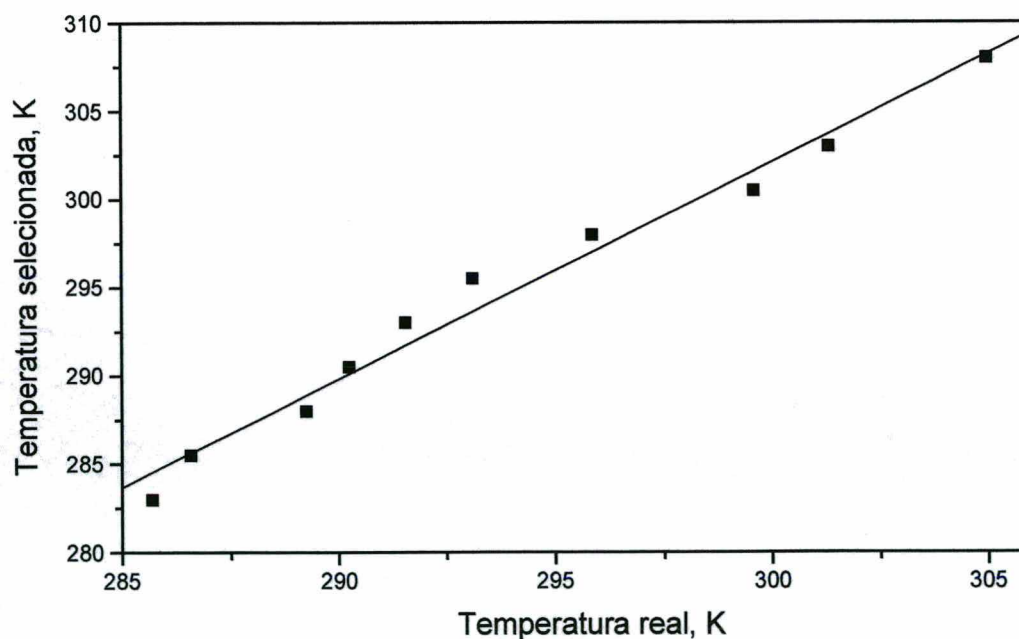


Figura 29 - Temperatura selecionada no sistema de termostatização e temperatura real da gota.

4.2. Misturas de Surfactantes

O instrumento desenvolvido tem mostrado resultados promissores para surfactantes puros e será agora aplicado para misturas de dois surfactantes. Vários sistemas envolvendo os surfactantes CTABr, TTABr, DOTAB, SDS, SDOS, SB3-14, SB3-12 e SB3-10 foram estudados, sendo que os tipos de misturas representam comportamento ideal e não-ideal.

4.2.1. Sistema Ideal

Trata-se de misturas cujos constituintes possuem cargas similares em termos de grupo cabeça e contra-íon. Foi selecionado o sistema envolvendo a mistura dos surfactantes TTABr e CTABr para estudo e os resultados encontram-se apresentados no gráfico da Figura 30-a. A Figura 30-b apresenta os valores das CMCs experimentais e os valores esperados para uma mistura com comportamento ideal, estimados a partir das equações 4, 5 e 6.^(33,34) Por meio da equação 7 foi determinado o valor de β para cada fração da mistura e a média aritmética calculada foi aproximadamente 0,8. O comportamento ideal da mistura é evidenciado tanto pelo valor de β como pela correlação entre os valores das CMCs experimentais e aqueles calculados com a equação para sistemas ideais, que é de 0,996. Esse comportamento já era esperado, uma vez que os dois surfactantes possuem estruturas semelhantes, diferindo apenas por dois átomos de carbono na cauda.⁽⁸⁴⁾ Outros sistemas químicos, cujos componentes da mistura são estruturalmente semelhantes (SDS + dodecanoato de sódio), também apresentam valores de β muito baixo (0,45).⁽²¹⁾

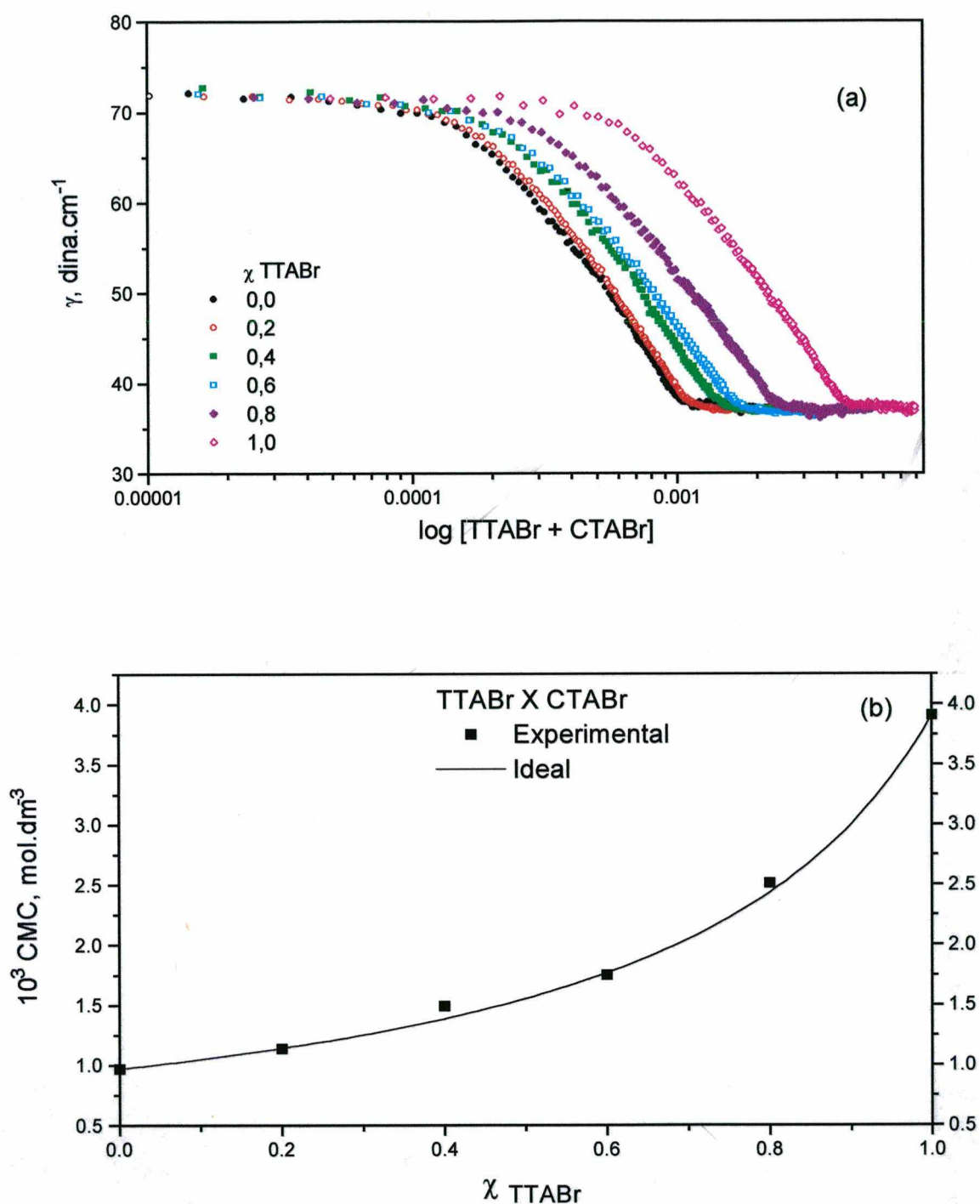


Figura 30 - (a) Tensão superficial em função da concentração total da mistura [TTABr + CTABr], a 25°C. (b) Valores das CMCs experimentais da mistura (■) em função da fração molar de TTABr. A linha contínua representa o comportamento de uma mistura ideal.

4.2.2. Misturas de Surfactantes Catiônicos e Sulfobetáinas

Os sistemas envolvendo as misturas dos surfactantes catiônicos DOTAB, TTABr e CTABr com a sulfobetáina SB3-14 foram estudados e os resultados serão apresentados a seguir .

A Figura 31-a apresenta os valores da tensão superficial da mistura dos surfactantes DOTAB e SB3-14 em função da concentração total, para várias frações de DOTAB. A Figura 31-b indica uma boa correlação entre os valores das CMCs experimentais e os esperados para uma mistura ideal. Porém, os valores das CMCs experimentais são ligeiramente menores que os valores para uma mistura ideal. Aplicando o modelo de Rubingh para essa mistura obtém-se um valor médio para β igual a -1,1, indicando uma leve interação entre os surfactantes.

A Figura 32-a apresenta os valores da tensão superficial da mistura dos surfactantes TTABr e SB3-14 em função da concentração total, para várias frações de TTABr. A Figura 32-b apresenta uma correlação muito boa entre os valores das CMCs experimentais e os esperados para uma mistura ideal. O valor médio para o parâmetro β é igual a -0,7, o que representa uma interação quase desprezível entre os constituintes da mistura.

A Figura 33-a apresenta os valores da tensão superficial da mistura dos surfactantes CTABr e SB3-14 em função da concentração total, para várias frações de CTABr. Na Figura 33-b estão representados os valores da CMC experimental e os valores esperados para uma mistura ideal. Nota-se que todos os valores experimentais são inferiores aos valores calculados para uma mistura ideal. O parâmetro β calculado para esse sistema é de -1,8, o que indica um desvio negativo em relação ao comportamento ideal um pouco mais acentuado. O valor das CMCs dos surfactantes catiônicos puros está em concordância com aqueles da literatura⁽⁸³⁾ (DOTAB = $1,53 \cdot 10^{-2}$, TTABr = $3,6 \cdot 10^{-3}$, CTABr = $9 \cdot 10^{-4}$ mol.dm⁻³).

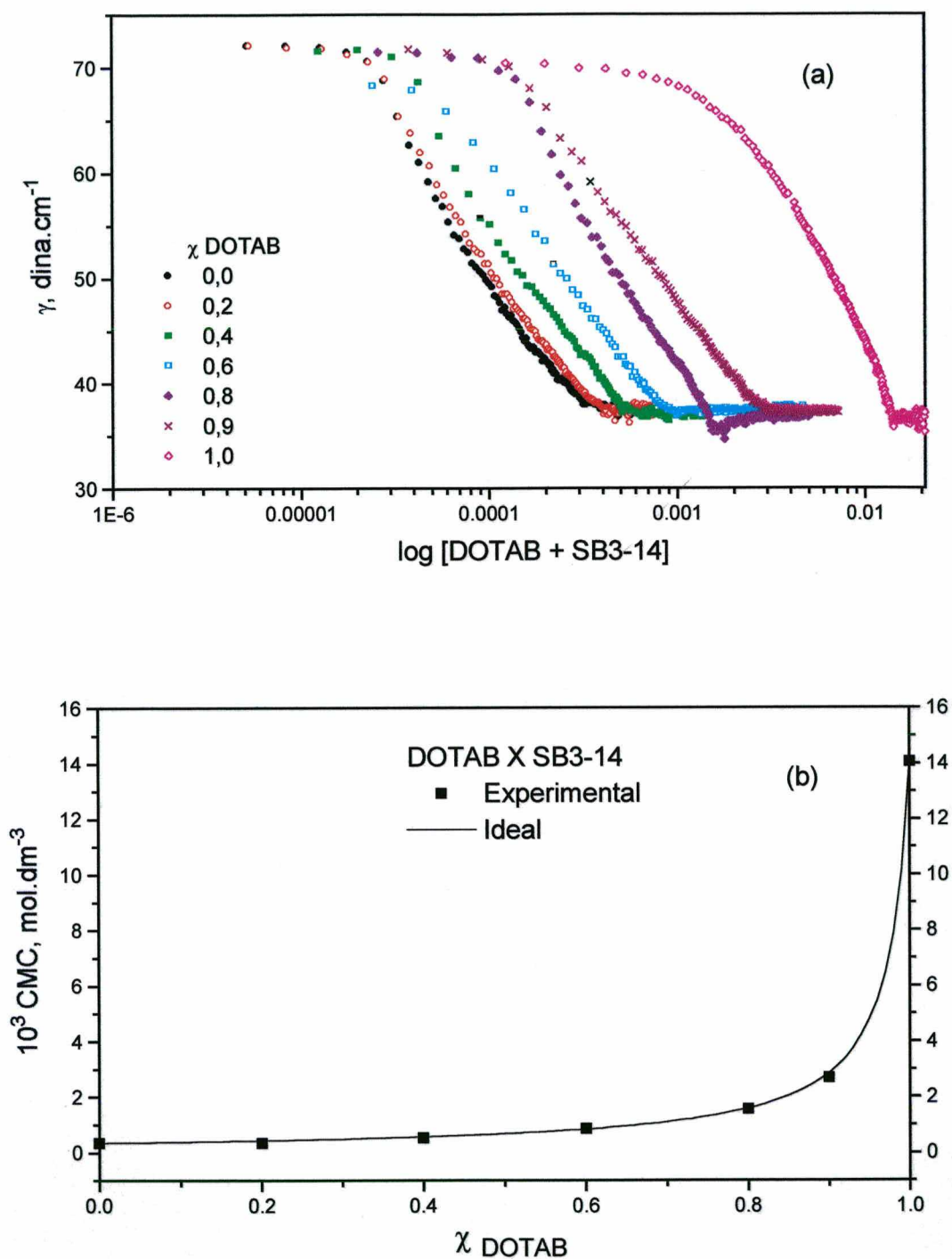


Figura 31 - (a) Tensão superficial em função da concentração total da mistura [DOTAB + SB3-14], a 25°C . (b) Valores das CMCs experimentais da mistura (■) em função da fração molar do DOTAB. A linha contínua representa o comportamento de uma mistura ideal.

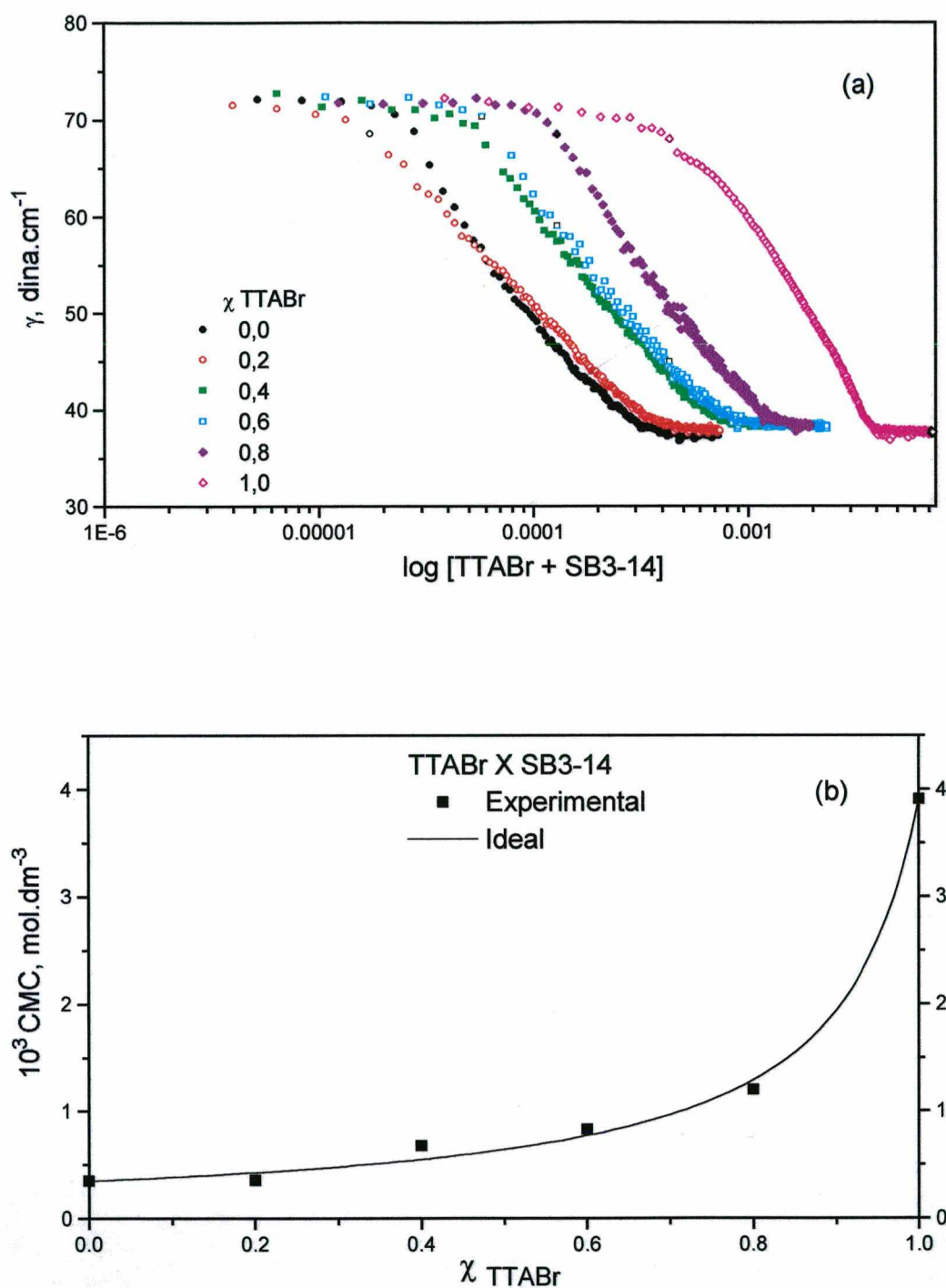


Figura 32 - (a) Tensão superficial em função da concentração total da mistura [TTABr + SB3-14], a 25°C. (b) Valores das CMCs experimentais da mistura (■) em função da fração molar do TTABr. A linha contínua representa o comportamento de uma mistura ideal.

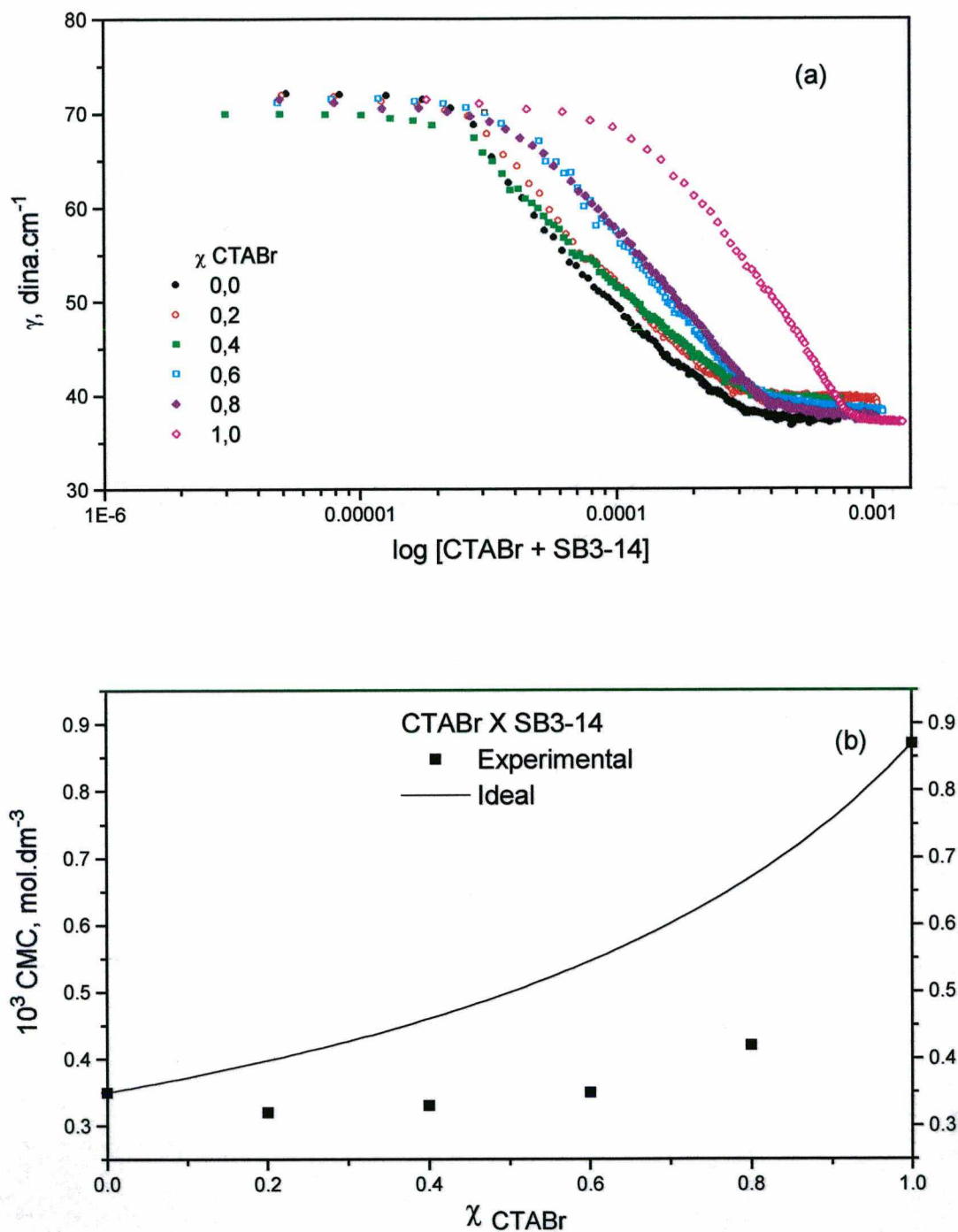


Figura 33 - (a) Tensão superficial em função da concentração total da mistura [CTABr + SB3-14], a 25°C. (b) Valores das CMCs experimentais da mistura (■) em função da fração molar do CTABr. A linha contínua representa o comportamento de uma mistura ideal.

4.2.3. Mistura de SDS e Sulfobetáínas

Os sistemas envolvendo as misturas das sulfobetáínas SB3-10, SB3-12 e SB3-14 com o surfactante SDS foram estudados. A Figura 34-a apresenta os valores da tensão superficial em função da concentração total de surfactante, para várias frações de SDS na mistura de SDS e SB3-10. A Figura 34-b apresenta os valores das CMCs experimentais da mistura em função da fração de SDS juntamente aos valores esperados para uma mistura cujo comportamento é o ideal. É de fácil observação que há uma forte interação sinérgica entre os dois constituintes da mistura, uma vez que todas as CMCs experimentais encontram-se bem abaixo dos valores calculados para uma mistura ideal. Esse desvio negativo é confirmado através do valor -6,2 para o parâmetro β . Outra importante observação é que os valores da tensão superficial após o ponto da CMC variam de forma significativa em função da composição da mistura.

Na Figura 35-a estão apresentados os valores de tensão superficial da mistura SDS+SB3-12 em função da concentração total de surfactante, para uma série de frações do SDS. A Figura 35-b apresenta os valores das CMCs experimentais e ideais para a mesma mistura. O valor -5,6 para o parâmetro β , igualmente ao caso anterior, indica haver um forte desvio negativo do comportamento ideal, que pode ser observado pelo gráfico da figura 35-b.

A Figura 36-a ilustra os valores da tensão superficial em função da concentração total da mistura SDS+SB3-14, para várias frações de SDS. Na Figura 36-b é mais uma vez evidenciado um valor negativo para β . Uma observação inicial do gráfico levaria à expectativa de haver um pequeno desvio em relação ao comportamento de uma mistura ideal, porém o valor calculado para β é de -7,1, a mais alta interação obtida neste trabalho.

Os resultados apresentados neste trabalho encontram-se na mesma faixa daqueles apresentados na literatura⁽³⁷⁻³⁹⁾ para sistemas químicos que envolvem surfactantes com estruturas próximas a estes utilizados.

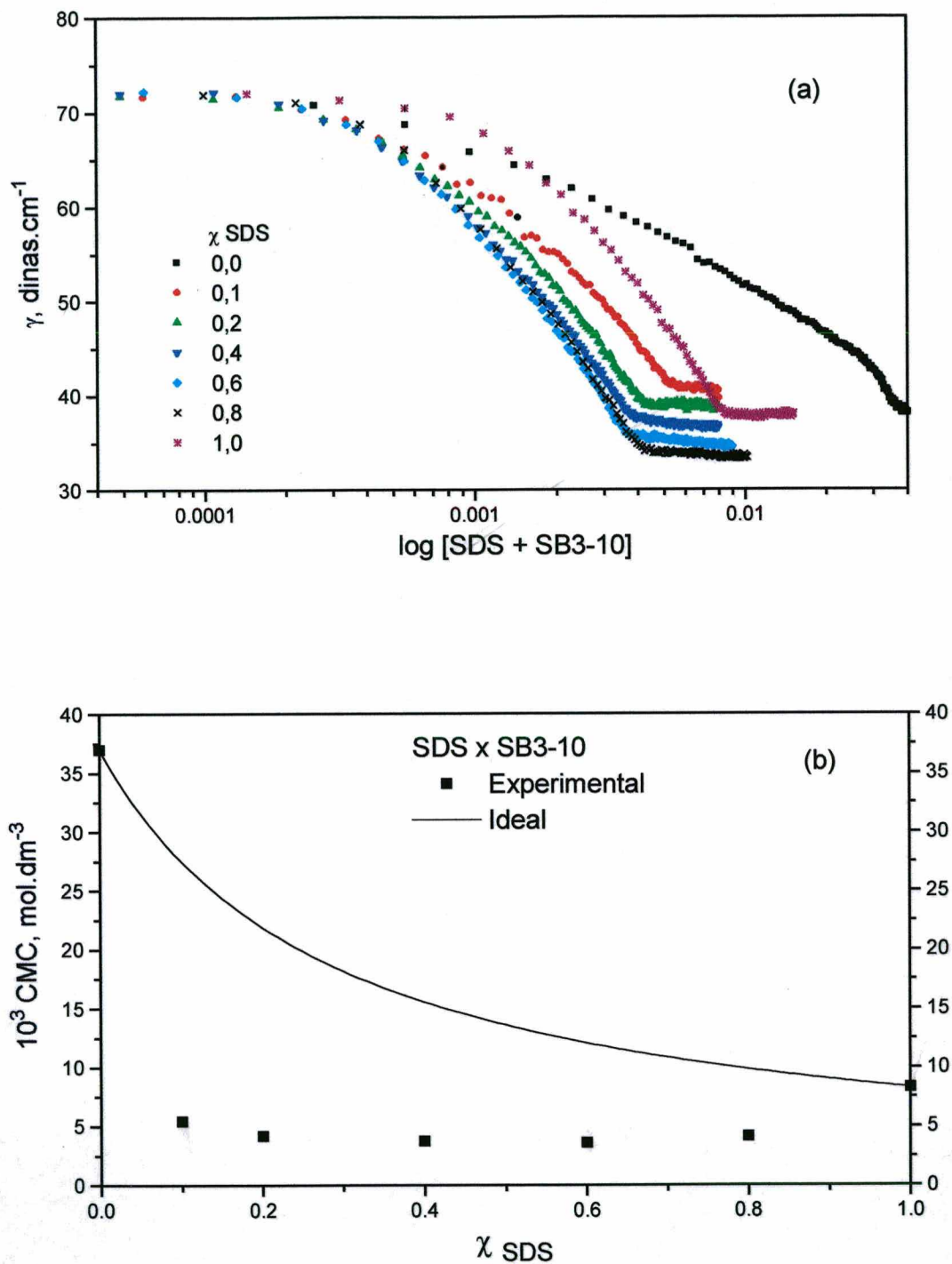


Figura 34 - (a) Tensão superficial em função da concentração total da mistura [SDS + SB3-10], a 25°C. (b) Valores das CMCs experimentais da mistura (■) em função da fração molar do SDS. A linha contínua representa o comportamento de uma mistura ideal.

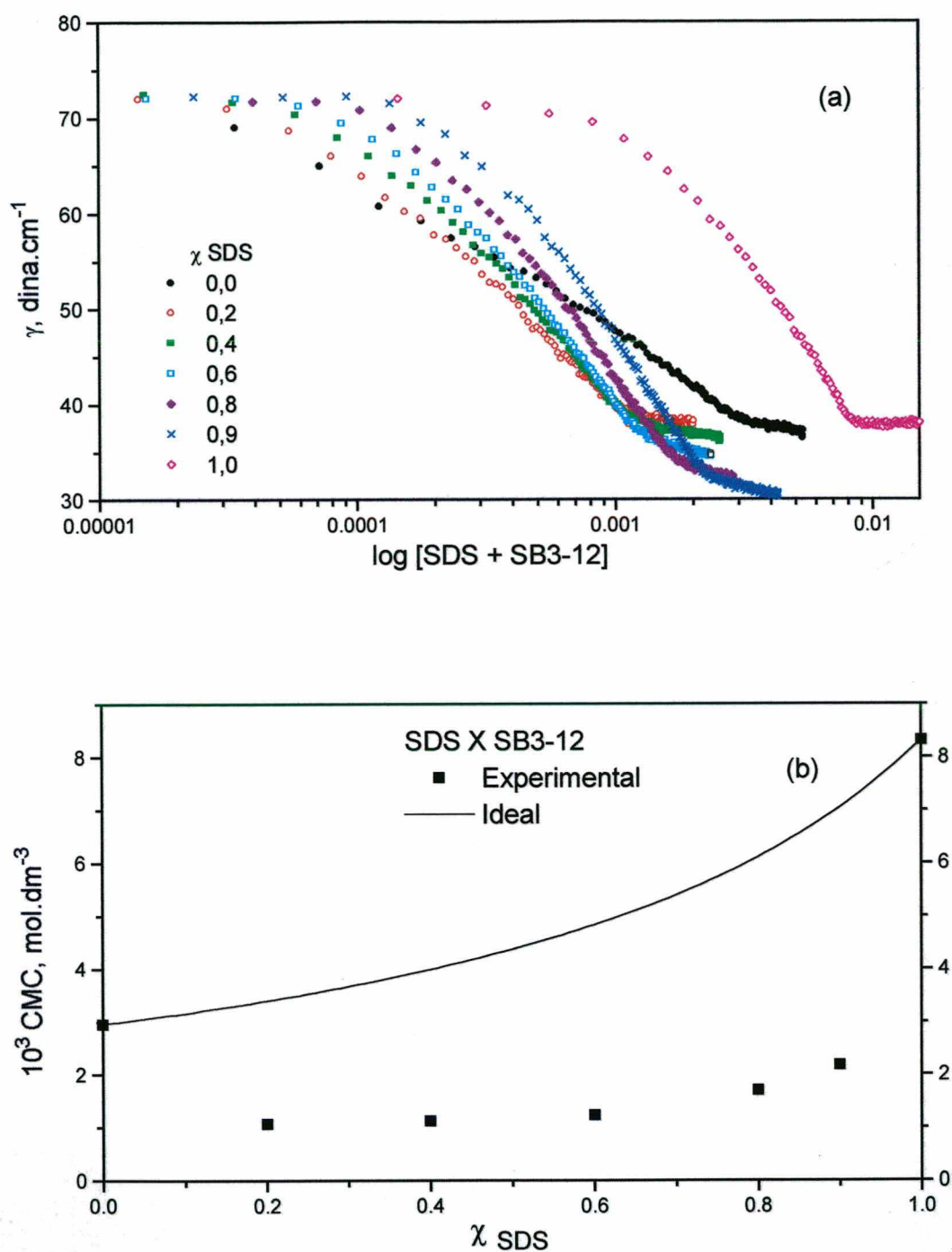


Figura 35 - (a) Tensão superficial em função da concentração total da mistura [SDS + SB3-12], a 25°C. (b) Valores das CMCs experimentais da mistura (■) em função da fração molar do SDS. A linha contínua representa o comportamento de uma mistura ideal.

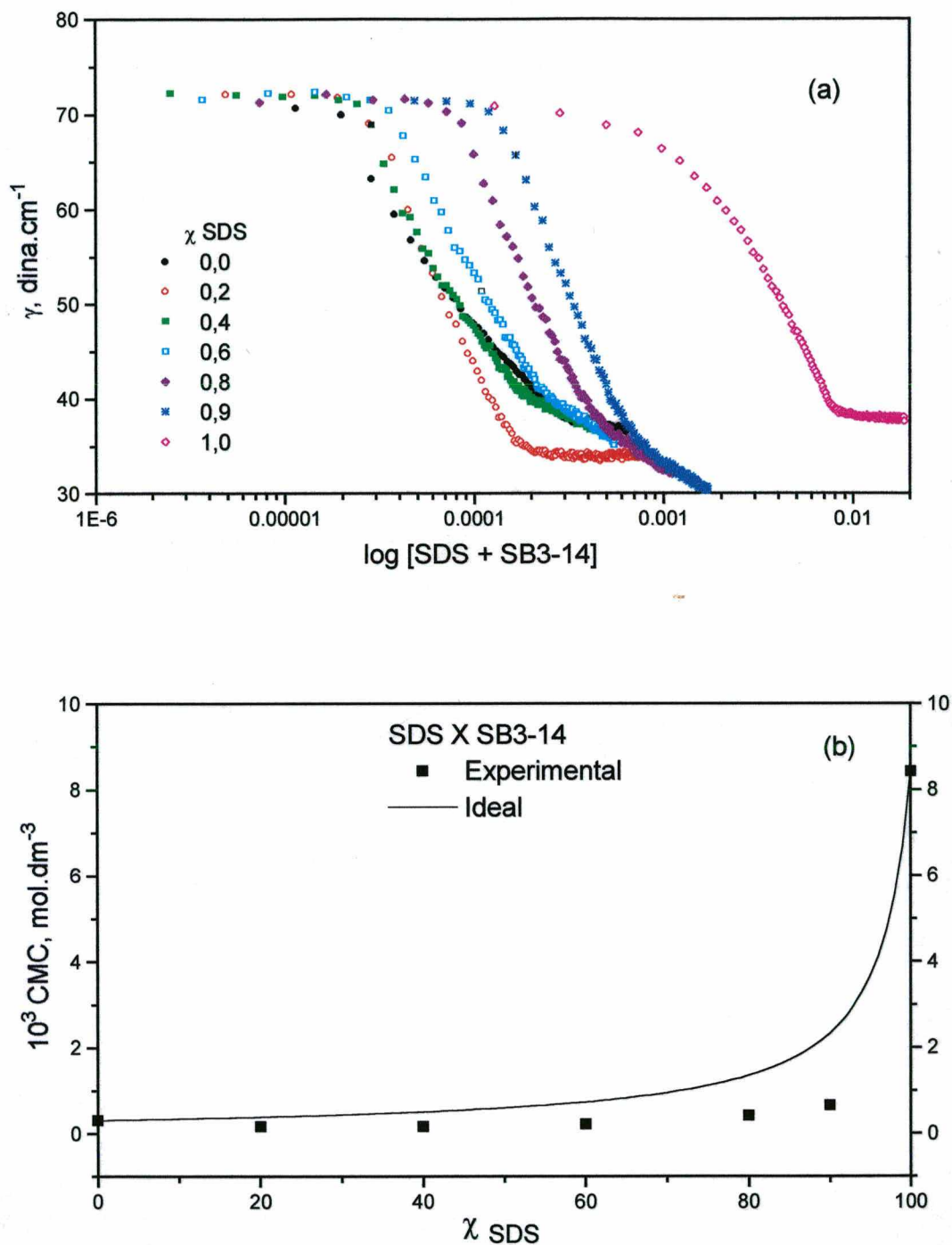


Figura 36 - (a) Tensão superficial em função da concentração total da mistura [SDS + SB3-14], a 25°C. (b) Valores das CMCs experimentais da mistura (■) em função da fração molar do SDS. A linha contínua representa o comportamento de uma mistura ideal.

4.2.4. Mistura de SDOS e Sulfobetainas

Os sistemas envolvendo as misturas das sulfobetainas SB3-10, SB3-12 e SB3-14 com o surfactante SDOS foram estudados. A Figura 37-a apresenta os resultados experimentais de tensão superficial para uma série de misturas de SDOS e SB3-10 em função da concentração total da mistura, em várias frações de SDOS. Na mesma figura, após a CMC, pode ser observado que os valores da tensão superficial variam em função da fração de SDOS presente na mistura. Um forte desvio negativo em relação ao comportamento ideal pode ser observado através do gráfico da Figura 37-b. É perceptível que todos os valores de CMC experimental são bastante inferiores aos calculados para uma mistura ideal. Nessa mistura, o valor calculado para o parâmetro β foi -3,9.

O sistema envolvendo os surfactantes SDOS e SB3-12 apresentou o comportamento ilustrado na Figura 38-a. Novamente, a Figura 38-b indica um valor negativo para o parâmetro β , cujo valor calculado é de -3,8.

As misturas dos surfactantes SDOS e SB3-14 apresentaram o comportamento descrito nas Figuras 39-a e 39-b. Na figura 39-a é possível verificar que as CMCs das misturas são intermediárias às CMCs dos constituintes puros. Na figura 39-b é pouco perceptível, mas em todo intervalo de fração do SDOS o valor da CMC experimental mostra-se inferior ao valor da CMC para uma mistura ideal e o parâmetro β calculado para essa mistura é igual a -4,6.

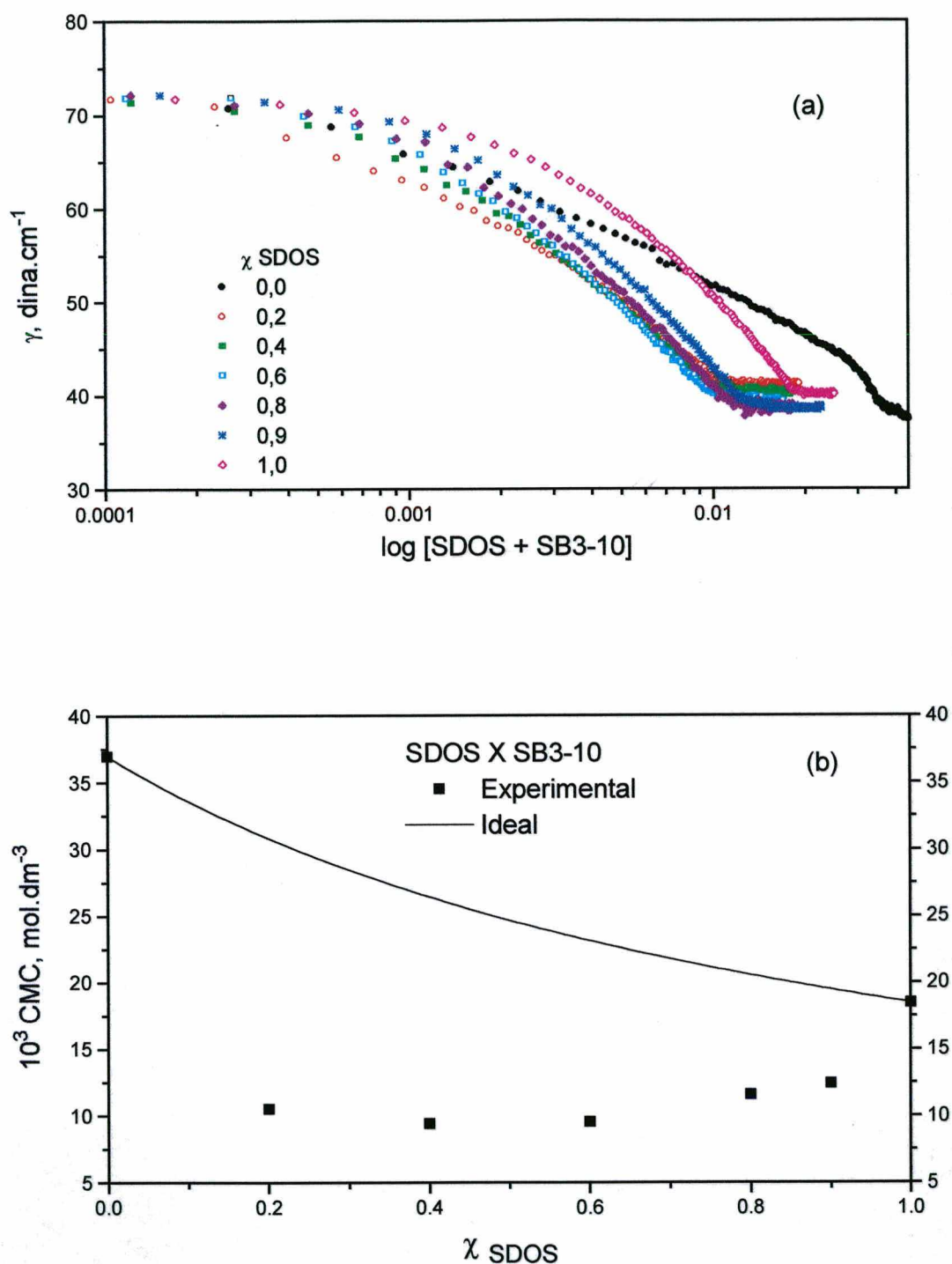


Figura 37 - (a) Tensão superficial em função da concentração total da mistura [SDOS + SB3-10], a 25°C. (b) Valores das CMCs experimentais da mistura (■) em função da fração molar do SDOS. A linha contínua representa o comportamento de uma mistura ideal.

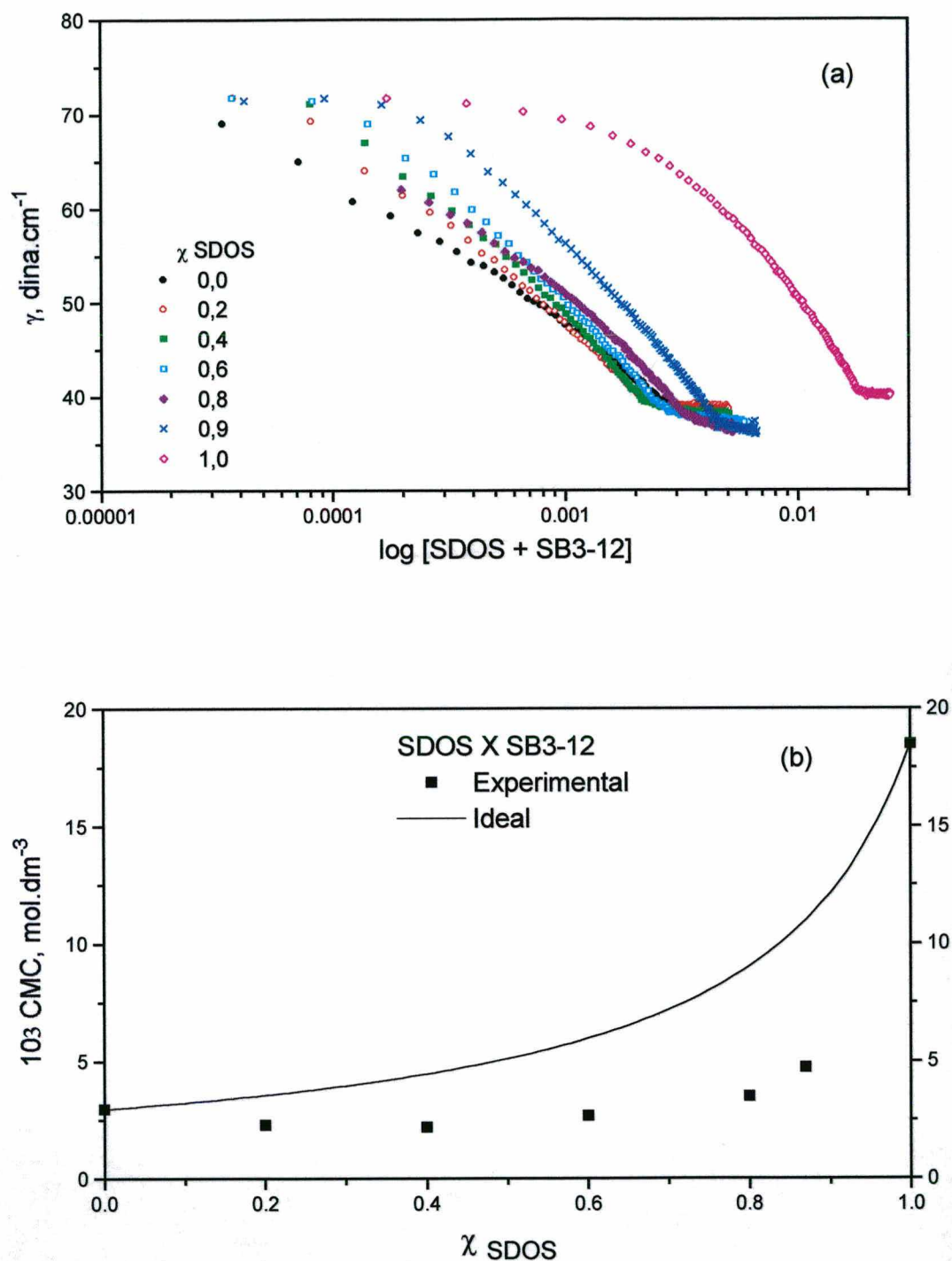


Figura 38 - (a) Tensão superficial em função da concentração total da mistura [SDOS + SB3-12], a 25°C. (b) Valores das CMCs experimentais da mistura (■) em função da fração molar do SDOS. A linha contínua representa o comportamento de uma mistura ideal.

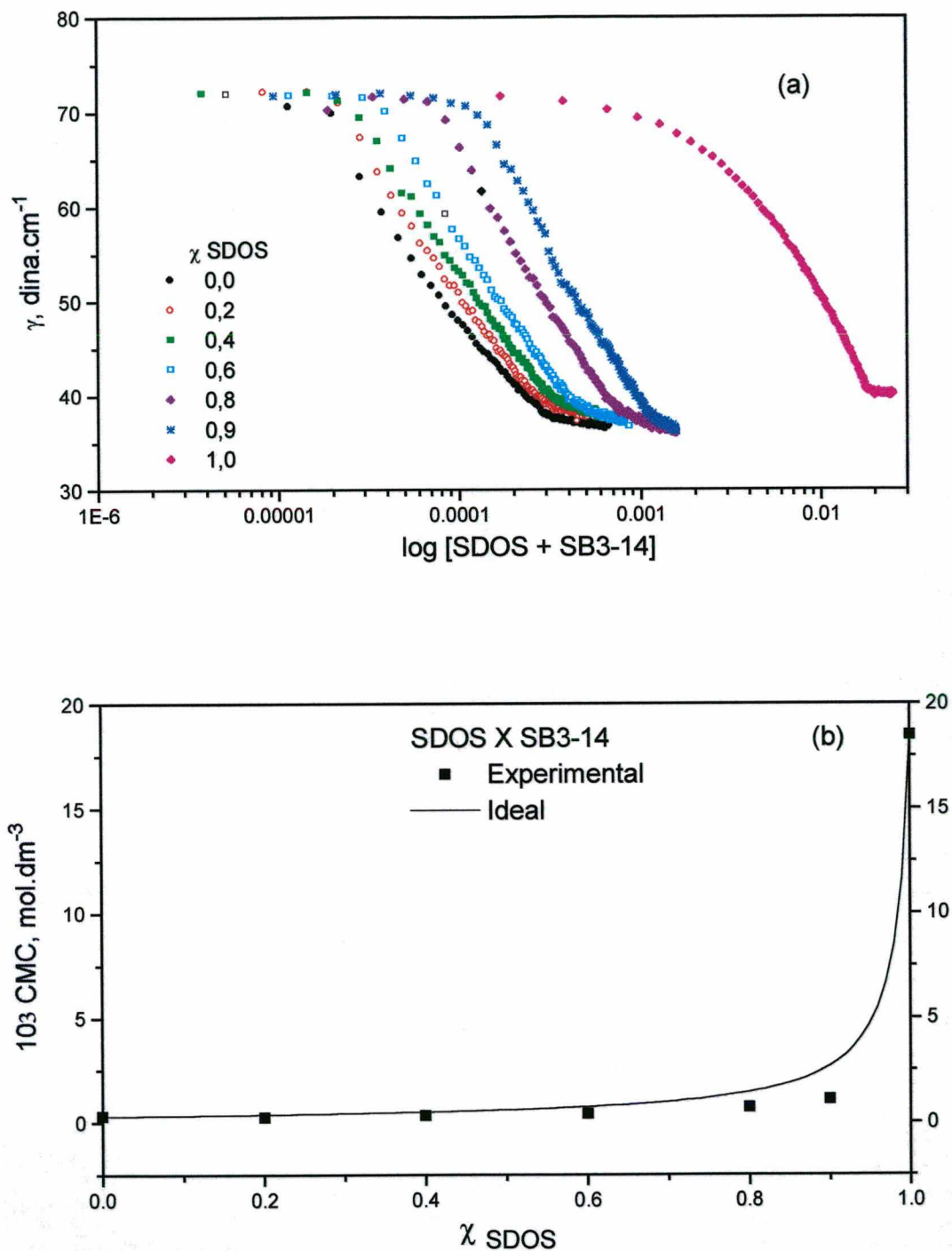


Figura 39 - (a) Tensão superficial em função da concentração total da mistura [SDOS + SB3-14], a 25°C. (b) Valores das CMCs experimentais da mistura (■) em função da fração molar do SDOS. A linha contínua representa o comportamento de uma mistura ideal.

4.2.5. Valores de β para as Misturas de Surfactante

Os valores do parâmetro β para os sistemas envolvendo a mistura dos surfactantes catiônicos DOTAB, TTABr e CTABr com a sulfobetaina SB3-14 encontram-se descritos na Figura 40. O maior valor calculado para β foi na mistura TTABr + SB3-14, onde a possibilidade de se encontrar os grupos polares catiônicos de ambos os surfactantes próximos um do outro é grande. Essa proximidade provocaria um aumento da repulsão entre eles e o comportamento da sua mistura seria muito próximo ao comportamento de uma mistura ideal. Entre as misturas desse grupo, o maior desvio em relação ao comportamento ideal foi observado para a mistura dos surfactantes CTABr e SB3-14. O provável motivo para isso é que, nesse caso, o grupo polar do CTABr encontra-se entre ambos os grupos polares do SB3-14 e a proximidade com o grupo de carga oposta aumentaria a interação entre ambos.

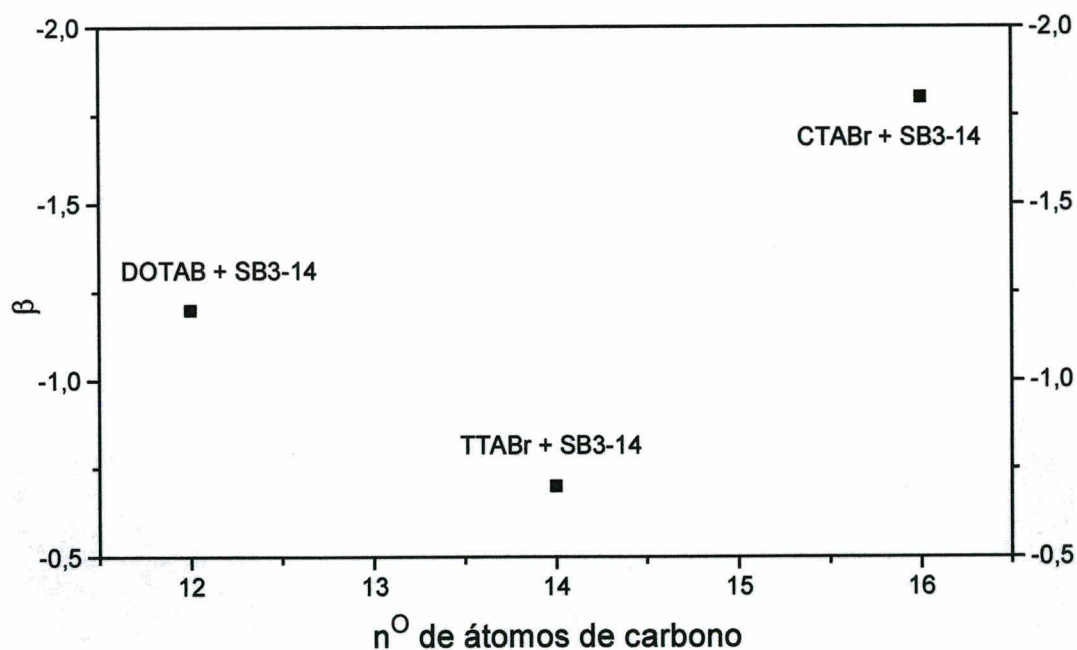


Figura 40 - Valores do parâmetro β para as misturas de DOTAB, TTABr e CTABr com o SB3-14 em função do número de átomos de carbono do surfactante catiônico.

O comportamento das misturas envolvendo as sulfobetainas e os surfactantes aniônicos encontra-se descrito na Figura 41. Neste gráfico é possível observar que as misturas envolvendo as sulfobetainas SB3-10, SB3-12 e SB3-14 com o surfactante SDOS apresentaram o menor desvio do comportamento ideal, quando comparadas às misturas envolvendo as mesmas sulfobetainas com o SDS. Outra importante observação é que as misturas envolvendo a sulfobetaina SB3-14 com os surfactantes aniônicos apresentaram um valor para β mais negativo quando comparadas às misturas das outras duas sulfobetainas com os mesmos surfactantes.

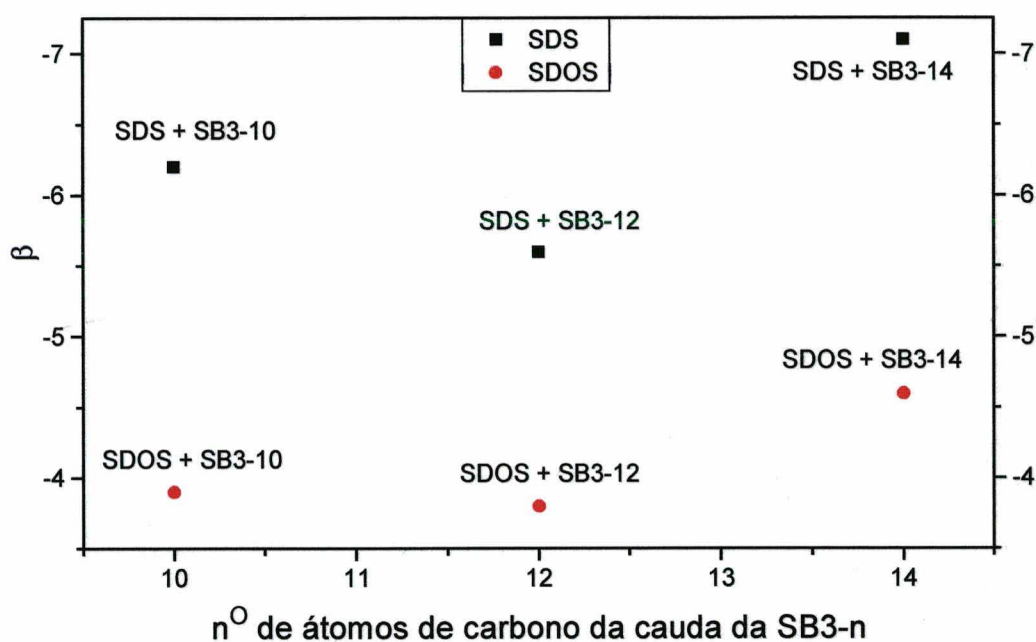


Figura 41 - Valores do parâmetro β para as misturas de SB3-10, SB3-12 e SB3-14 com o SDS e SDOS em função do número de átomos de carbono da sulfobetaina.

Capítulo V

Considerações Gerais

**"A maioria de nós prefere olhar para fora
e não para dentro de si mesmo"**
Albert Einstein

**"Muitas vezes o que se cala faz maior impacto
do que o que se diz"**
Píndaro

**"Os miseráveis não têm outro remédio
a não ser a esperança"**
William Shakespeare

5.1. Tensiômetro

A utilização de um computador simples, como um PC/compatível 386, torna o sistema de baixo custo. A utilização de um computador mais rápido como um Pentium (PC/compatível 586) não tornaria o tensiômetro mais eficiente em nenhum aspecto, pois:

- A resolução das medidas realizadas pelo sistema é igual à resolução dada pela balança analítica, e independe do computador;
- A velocidade de aquisição de dados é decorrente exclusivamente da tensão superficial da gota e da velocidade da bomba de seringa. Mesmo em situações extremas, na qual a vazão selecionada para a bomba de seringa seja excessivamente alta, o computador apenas movimenta a bomba de seringa e adquire os dados enviados pela balança e pelo sensor ótico de gotas. Essas tarefas são realizadas rapidamente e durante o restante do tempo nada é feito pelo computador. O único momento em que alguma tarefa adicional é realizada é quando ocorre a queda da gota, onde os cálculos de γ , de concentração e de densidade são exigidos. Como tarefa adicional, o dado é salvo em disco e representado na tela do computador na forma gráfica.

A utilização de um sistema de fluxo e de mistura controlado por computador, tornaria o tensiômetro bastante prático, uma vez que nenhuma operação manual precisaria ser realizada pelo analista e um grau maior de automação poderia ser alcançado. Porém, a dificuldade em conseguir precisão adequada na bomba de seringa obrigou-nos a uma modificação no projeto inicial. A aplicação de uma balança analítica tornou o equipamento muito mais exato e preciso, e ainda assim foi mantida a sua viabilidade. Em contrapartida, perdeu-se a possibilidade de realimentação dos parâmetros de operação.

Uma vantagem adicional, é que os equipamentos utilizados (a balança, o banho termocriostático, o computador e o agitador magnético) têm funcionamento dedicado somente enquanto o tensiômetro estiver operando. Em outras palavras, para utilizar o computador, como editor de textos, ou a balança, para medir a massa de um material qualquer, é necessário apenas ligá-los e proceder de forma normal, tal como seria feito se nenhum tensiômetro estivesse presente.

Dessa forma, o investimento necessário para adquirir os equipamentos exclusivos do tensiômetro é bastante reduzido.

A câmara de mistura mostrou-se eficiente, porém deve-se ter o cuidado de garantir a perfeita homogeneidade da solução ali contida. Uma baixa vazão deve ser utilizada para essa finalidade, e também, para que o processo de formação da gota ocorra em estado de equilíbrio.

As medidas de massa da gota de água pura (figura 23) apresentaram para o desvio padrão relativo o valor máximo de 0,21%, para uma análise com 5 replicatas. Deve-se salientar que melhores resultados poderiam ser obtidos, bastando, para isso, que o número de replicatas fosse aumentado, uma vez que o desvio devido aos erros aleatórios diminui na proporção da raiz quadrada do número de medidas.

O baixo valor para o desvio padrão relativo pode, ao menos em parte, ser atribuído à utilização de um amortecedor de vibrações (borracha) junto à base da balança. Também é possível concluir que:

- A gota em formação, abaixo do capilar, é bastante estável;
- O nível de ruídos sonoros na sala onde ocorreu o experimento é baixo;
- O nível de vibrações da estrutura física do prédio devido às perturbações externas causadas pôr caminhões trafegando nas proximidades, por bate-estacas utilizadas na construção civil, por aparelhos de ar condicionado e geladeiras, etc, é extremamente baixo ou o amortecedor de vibrações utilizado funcionou de forma eficiente.

O valor da concentração da gota calculada pelo programa (Tabela 1) apresentou-se muito próximo do valor atribuído à mesma por espectrofotometria. Esse resultado garante que o comportamento da câmara de mistura, quanto ao efeito de diluição, esteja corretamente equacionado (Equações 12-17), que essas equações estejam sendo corretamente aplicadas pelo programa e que o processo de diluição ocorra em equilíbrio.

O sistema de termostatização desenvolvido mostrou-se eficiente e pode ser utilizado com segurança entre 10 e 50 C.

A variação no valor da CMC do SDS em função da mudança da vazão pode ser atribuída à idade da superfície da gota ou à não-homogeneidade da câmara de mistura, sendo que os valores corretos são obtidos a baixas vazões.

Um outro fator a ser considerado é a evaporação de água, a partir da gota em formação, quando a vazão é baixa, devido ao tempo elevado de exposição da gota. Nessa situação, a concentração real de surfactante é ligeiramente superior à concentração calculada e a CMC seria atingida de forma prematura.

Os experimentos com CTABr (Figura 28), com a mistura TTABr + CTABr (Figura 30) e com boa parte dos surfactantes puros não apresentaram pontos discrepantes, comprovando mais uma vez a alta estabilidade do aparelho.

Em condições de equilíbrio, para uma única medida de γ é necessário aproximadamente 1 ou 2 minutos. Um gráfico completo, nos experimentos que envolvem variação de concentração, é obtido em um período de 2 a 6 horas, dependendo das condições experimentais. O processo completo de análise envolve o preparo do equipamento, a realização das medidas experimentais e a análise dos dados. A presença do analista é requerida apenas em 10% de todo o tempo necessário para realizar essas tarefas.

5.2. Misturas de surfactantes

O sistema TTABr + CTABr (Figura 30) apresentou um comportamento ideal. Esse resultado já era esperado, em virtude de esses surfactantes diferirem entre si por 2 átomos de carbono.

As misturas entre os surfactantes catiônicos DOTAB, TTABr, e CTABr com a sulfobetaína SB3-14 apresentaram um pequeno desvio do comportamento ideal, crescente nessa mesma sequência.

As misturas das sulfobetaínas SB3-10, SB3-12 e SB3-14 com o surfactante aniônico SDS apresentou maior interação que as misturas das mesmas sulfobetaínas com o surfactante SDOS.

As misturas envolvendo a sulfobetaína SB3-14 com os surfactantes aniônicos SDS e SDOS apresentaram maior interação quando comparadas às misturas das sulfobetaínas SB3-12 e SB3-10 com os mesmos surfactantes aniônicos.

5.3. Outros Sistemas Químicos

A utilização desse instrumento para medidas de propriedades de outros sistemas químicos também é possível, respeitando-se as limitações instrumentais e as do método. Dessa forma, estudos das ligações entre surfactantes e amiláceos,⁽⁸⁵⁾ entre surfactantes e proteínas⁽⁸⁶⁾ ou trabalhos envolvendo misturas de polímeros, como sulfato de dodecilpolioxietileno,⁽⁴⁰⁾ podem ser realizados, desde que o equilíbrio na superfície da gota seja propriamente estabelecido.

A aplicação prática desse instrumento é dependente da natureza da interface, da viscosidade do líquido, da faixa de tensão superficial, da faixa de temperatura e pressão, do envelhecimento da superfície, da facilidade da análise, da precisão, da exatidão, da velocidade de aquisição de dados e de considerações práticas a respeito da conveniência, do custo e das combinações com outros métodos.⁽⁸⁷⁾

Capítulo VI

Referências Bibliográficas

"Os mais ilustres entre os gregos sustentavam que a escravidão era justificável sempre que os amos fossem gregos e os escravos bárbaros, porém o caso oposto era contrário à natureza"

Russell

"Há vários motivos para não se amar uma pessoa e um só para amá-la"
Carlos Drumond de Andrade

"O que busca a verdade corre o risco de encontrá-la"
Isabel Allende

REFERÊNCIAS BIBLIOGRÁFICAS

- 1- HAYAMI, Y. The influence of asymmetry in drop shape on an interfacial tension measurement. **Colloid Polym. Sci.**, v.274, p.643-647, 1996.
- 2- McMILLAN, N.D.; FORTUNE, F.J.M.; FINLAYSON, O.E.; McMILLAN, D.D.G.; TOWNSEND, D.E.; DALY, D.M.; FINGLETON, M.J.; DALTON, M.G.; CRYAN, C.V.; A fiber drop analyzer: A new analytical instrument for the individual, sequential, or collective measurement of the physical and chemical properties of liquids. **Rev. Sci. Instrum.**, v.63, p.3431-3454, 1992.
- 3- FAOUR, G.; GRIMALDI, M.; RICHOU, J.; BOIS, A., Real-time pendant drop tensiometer using image processing with interfacial area and interfacial tension control capabilities. **J. Colloid Interface Sci.**, v.181, p.385-392, 1996.
- 4- LIN, S.-Y.; HWANG, H.-F., Measurement of low interfacial tension by pendant drop digitization. **Langmuir**, v.10, n.12, p.4703-4709, 1994.
- 5- ANASTASIADIS, S.H.; CHEN, J.-K.; KOBERSTEIN, J.T.; SIEGEL, A.F.; SOHN, J.E.; EMERSON, J.A. The determination of interfacial tension by video image processing of pendant fluid drops. **J. Colloid Interface Sci.**, v.119, p.55-66, 1987.
- 6- LIGGIERI, L.; RAVERA, F.; PASSERONE, A. Dynamic interfacial tension measurements by a capillary pressure method. **J. Colloid Interface Sci.**, v.169, p.226-237, 1995.
- 7- SOOS, J.M.; KOCZO, K.; ERDOS, E.; WASAN, D.T. An automatic apparatus for measuring interfacial and film tension under static and dynamic conditions. **Rev. Sci. Instrum.**, v.65, n.11, p.3555-3562, 1994.
- 8- HENDERSON, D.C.; MICALE, F.J. Dynamic surface tension measurement with the drop mass technique. **J. Colloid Interface Sci.**, v.158, p.289-294, 1993.

- 9- MATSUKI, H.; KANESHINA, S. Automatic surface tension measurements of aqueous surfactant solutions by the drop volume method. **Langmuir**, v.10, n.11, p.4394-4396, 1994.
- 10- FAINERMAN, V.B.; MILLER, R. Hydrodynamic effects in measurements with the drop volume technique at small drop times. 2. Drop time and drop volume bifurcations. **Colloids and surfaces A**, v.97, p.255-262, 1995.
- 11- NUNEZ-TOLIN, V.; HOEBREGS, H.; LÉONIS, J.; PAREDES, S. Surface tension measurements by the drop-weight method for continuously varying surfactant concentration. **J. Colloid Interface Sci.**, v.85, n.2, p.597-600, 1982.
- 12- TORNBERG, E. A surface tension apparatus according to the drop volume principle. **J. Colloid Interface Sci.**, v.60, n.1, p.50-53, 1977.
- 13- MOLLET, C.; TOUHAMI, Y.; HORNOF, V. A comparative study of the effect of ready-made and *in situ*-formed surfactants on interfacial tension measured by drop volume tensiometry. **J. Colloid Interface Sci.**, v.178, p.523-530, 1996.
- 14- MOTOMURA, K.; IWANAGA, S.-I.; HAYAMI, Y.; URYU, S.; MATUURA, R.; Thermodynamic studies on absorption at interfaces IV. Dodecylammonium chloride at water/air interface. **J. Colloid Interface Sci.**, v.80, n.1, p.32-38, 1981.
- 15- ADAMSON, A.W. **Physical chemistry of surfaces**. 5ed. New York: J. Willey & Sons, 1976.
- 16- ELWORTHY, P.H.; FLORENCE, A.T.; MACFARLANE, C.B. **Solubilization by surface active agents**. London: Chapman and Hall, 1968.
- 17- FENDLER, J.H. Interactions and reactions in reversed micellar systems. **Acc. Chem. Res.**, v.9, p.153-161, 1976.
- 18- FENDLER, J.H.; FENDLER, E.J. **Catalysis in micellar and macromolecular systems**. New York: Academic Press, 1975.
- 19- FENDLER, J.H. **Membrane mimetic chemistry**. New York: Willey-Interscience, 1982.

- 20- MARTINEK, K. et al.; The principles of enzyme stabilization. VI- Catalysis by water-soluble enzyme entrapped into reversed micelles of surfactants in organic solvents. **Biochim. Biophys. Acta**, v.657, p.277-294, 1981.
- 21- FRESCURA, V.L.A.; **Estudo da reação de hidrólise de ésteres arílicos e anidrido benzóico em presença de dodecanoato de sódio e sulfobetainas**. Fpolis, 1995. Tese de Doutorado em Físico-Química - Departamento de Química - UFSC.
- 22- SHINODA, K.; HUTCHISON, E. Pseudo-phase separation model for thermodynamic calculations on micellar solutions. **J. Phys. Chem.** v.66, p.577-582, 1962.
- 23- EMERSONM, M.F.; HOLTZER, A. On the ionic strength dependence of micelle number., **J. Phys. Chem.**, v.69, p.3718-3721, 1965.
- 24- SHAW, D.J. **Introdução à química dos coloides e de superfícies**. São Paulo: E. Blucher, 1975.
- 25- HARTLEY, G.S. **Aqueous solutions of paraffin chain salts**. Paris: Hermann et Cie., 1936.
- 26- BROWN, W.H. **Organic chemistry**. Orlando: Saunders College Publishing, 1995.
- 27- EVANS, H.C. Alkyl sulphates part I - Critical micelle concentration of the sodium salts. **J. Chem. Soc.**, v.1, p.579-586, 1956.
- 28- ROMSTED, L.S.; **Rate enhancements in micellar systems**. Indiana, 1975. Tese de doutorado em química - Department of Chemistry, Indiana University Bloomington.
- 29- MUKERJEE, P.; MYSELS, K.J. **Critical micelle concentrations of aqueous surfactants systems**. NSDRS-NBS 36, 1971.
- 30- IKEDA, N.; SANAFUJI, N.; LU, K.-K.; ARATONO, M.; MOTOMURA, K. Surface Absorption and Micelle Formation of the Mixture of N-methylated Dodecyl Ammonium Chloride and Dodecylammonium Chloride. **J. Colloid Interface Sci.**, v.164, n.2, p.439-443, 1994.

- 31- SCAMEHORN, J. **An overview of phenomena involving surfactant mixtures in phenomena in mixed surfactant systems.** Washington: American Chemical Society, 1986.
- 32- FUNASAKI, N; HADA, S. Demixing of micelles of fluorocarbon and hydrocarbon surfactants. Effects of hydrophobic and hydrophilic groups of hydrocarbon surfactants. **J. Phys. Chem.**, v. 87, n. 2, p. 342-347, 1983.
- 33- HOLLAND, P.M; RUBINGH, D.N.; **Mixed surfactant systems.** Washington: American Chemical Society, 1992.
- 34- HOLLAND, P.M.; RUBING, D.N., Nonideal multicomponent mixed micelle model. **J. Phys. Chem.**, v.87, p. 1984-1990, 1983.
- 35- Guo, L. N.; ARNAUD, I.; PETIT-RAMEL, M.; GAUTHIER, R.; MONNET, C.; PERCHEC, P. L.; CHEVALIER, Y. Solution behavior of dye-surfactant associations. **J. Colloid Interface Sci.**, v.163, n.2, p.334-346, 1994.
- 36- ABE, M.; KATO, K.; OGINO, K. Effects of inorganic eletrolytes and of pH on micelle formation of amphoteric-anionic mixed surfactant systems. **J. Colloid Interface Sci.**, v.127, p.328-335, 1988.
- 37- SHILOACH, A.; BLANKSCHTEIN, D. Prediction of critical micelle concentrations and synergism of binary surfactant mixtures containing zwitterionic surfactants. **Langmuir**, v.13, p.3968-3981, 1997.
- 38- ROSEN, M.J.; ZHU, B.Y. Synergism in binary mixtures of surfactants. iii. betaïne-containing systems. **J. Colloid Interface Sci.**, v.99, n.2, p.427-434, 1984.
- 39- ZHU, B.Y.; ROSEN, M.J. Synergism in binary mixtures of surfactants. iv. effectiveness of surface tension reduction. **J. Colloid Interface Sci.**, v.99, n.2, p.435-442, 1983.
- 40- ZHU, D-M; ZHAO, G-Z. Effect of polyoxyethylene chain length on the interactions between anionic and nonionic surfactants. I. Sodium dodecylpolyoxyethylenesulfate-decylmethylsulfoxide systems. **Colloids And Surfaces**, v.49, p.269-279, 1990.

- 41- TATE, T. **Phil. Mag.** v.27, p.176, 1864.
- 42- HARKINS, W.D.; BROWN, F.E. The determination of surface tension (free surface energy), and the weight of falling drops: the surface tension of water and benzene by the capillary height method. **J. Am. Chem. Soc.**, v.41, p.499-524, 1919.
- 43- LANDO, J.L.; OAKLEY, H.T. Tabulated correction factors for the drop-weight-volume determination of surface and interfacial tensions. **J. Colloid Interface Sci.**, v.25, p.526-530, 1967.
- 44- SOUZA, I.G.; QUEIRÓZ, R. R. U.; LEHMKUHL, A.; GIACOMOZZI, C.A.; SCHMIDT, J.C.; Desenvolvimento de um dispensador automatizado empregando sistemas de análise química por injeção em fluxo. in: REUNIÃO ANUAL DA SOCIEDADE BRASILEIRA DE QUÍMICA (18.: 1995:Caxambú). **Anais...** Caxambú, 1995.
- 45- SOUZA, I.G.; QUEIRÓZ, R.R.U.; LEHMKUHL, A.; GIACOMOZZI, C.A.; LIMA, C.F.; MINATTI, E.; SCHMIDT, J.C.; CESCINETTO, L.L.; Titulador potenciométrico automático empregando sistemas de injeção em fluxo. in: CONGRESSO BRASILEIRO DE QUÍMICA (34.: 1994: Porto Alegre). **Anais...** Porto Alegre, 1994.
- 46- MARIN, M.A.B.; LEHMKUHL, A; GANZAROLLI, E.M.; QUEIRÓZ, R.R.U; SOUZA, I.G.; Sistema automático para determinação sequencial de cianeto livre e total empregando detector potenciométrico em sistemas FIA. em preparação.
- 47- MARIN, M.A.B.; LEHMKUHL, A; GANZAROLLI, E.M.; QUEIRÓZ, R.R.U; SOUZA, I.G.; Sistema automático para determinação sequencial de cianeto livre e total empregando detector potenciométrico em sistemas FIA. in ENCONTRO NACIONAL DE QUÍMICA ANALÍTICA (9.: 1997: São Carlos). **Anais...** São Carlos:1997. pg. 137.
- 48- GANZAROLLI, E.M.; LEHMKUHL, A; QUEIRÓZ, R.R.U.; SOUZA, I.G. Titulador potenciométrico automatizado baseado em sistema de fluxo monossegmentado. **Química Nova.** (submetido).

- 49- GANZAROLLI, E.M.; LEHMKUHL, A.; QUEIRÓZ, R.R.U.; SOUZA, I.G. Titulador potenciométrico automatizado baseado em sistema de fluxo monossegmentado. in: REUNIÃO ANUAL DA SOCIEDADE BRASILEIRA DE QUÍMICA (19.: 1996: Poços de Calda). **Anais...** Poços de Calda:1996.
- 50- SILVA, J.B.B.;GIACOMELLI, M.B.; LEHMKUHL, A.; CURTIUS A. J.; QUEIRÓZ, R.R.U.; SOUZA, I.G. Determinação de chumbo e estanho em ligas metálicas por espectrometria de absorção atômica após eletrodissolução anódica em linha. **Química Nova.** (submetido).
- 51- SILVA, J.B.B.; GIACOMOZZI, C.A.; LEHMKUHL, A.; GIACOMELLI, M.B.; SOUZA, I. G., QUEIRÓZ, R.R.U. Determinação de estanho em ligas de latão por eletrodissolução anódica de corrente contínua em sistemas FIA e GFAAS. in: ENCONTRO NACIONAL DE QUÍMICA ANALÍTICA (9.: 1997: São Carlos). **Anais...** São Carlos: 1997. p.140.
- 52- REIS, B.F.; MARTELLI, P.B.; MENEGÁRIO, A.A.; GINÉ, M^A F. Sistema de análise química por injeção em fluxo controlado por microcomputador para diluição automática em espectrometria de absorção atômica. **Química Nova**, v.16, n.2, p.109-112, 1993.
- 53- REIS, B.F. Análise química por injeção em fluxo: vinte anos de desenvolvimento. **Química Nova.**, v.19, n.1, p.51-58, 1996.
- 54- RUZICKA, J.; HANSEN, E. Flow injection anasusis. Part I. A new concept of fast constinuous flow analysis. **Anal. Chim. Acta.**, v.78, p.145-157, 1975.
- 55- REIS, B.F.; BERGAMIN F^O, H. Evolução dos injetores empregados em sistemas de análise química por injeção em fluxo. **Química Nova**, v.16, n.6, p.570-573, 1993.
- 56- FARIA, L.C.; PASQUINI, C. Um módulo automático de introdução de amostras para sistemas de análise por injeção em fluxo. **Química Nova**, v.14, n.3, p.216-218, 1991.

- 57- REIS, B.F.; GINE, M^A. F. Um microcomputador de baixo custo para controle e aquisição de dados em análise química por injeção em fluxo. **Química Nova**, v.15, n.3, p.231-236, 1992.
- 58- BRIEN, S.O. **Turbo Pascal 6.0. Completo e total.**, São Paulo: McGraw - Hill, 1979.
- 59- **Sigma Stepping Motort Handbook.** Sigma Instruments Inc., 1972.
- 60- **Sistema de aquisição de dados e controle MQI12/8PCC.** Microquímica Ind. e Com., Florianópolis, 1994.
- 61- BRAGA, N. C. **Manutenção de impressoras jato de tinta.** Saber eletrônica, ano 33, n.298, pg.57-60, nov/97.
- 62- BROPHY, J.J. **Basic electronics for scientists.** 2ed. New York: McGraw-Hill, 1972.
- 63- LENK, J.D. **Manual for integrated circuits users.** Reston: Reston Publishing Company Inc., 1973.
- 64- WEIGERT, C.; **Síntese e caracterização do n-decil-1-oxietilenosulfonato de sódio e estudo de suas propriedades em sistemas liomesoméricos.** Fpolis, 1996. Dissertação de Mestrado em Química Orgânica - Departamento de Química - UFSC.
- 65- CUCCOVIA, I.M.; FEITOSA, E.; CHAIMOVICH, H.; SEPÚLVEDA, L.; REED, W., Size, eletrophoretic mobility, and ion dissociation of vesicles prepared with synthetic amphiphiles. **J. Phys. Chem.**, v.94, n.9, p.3722-3725, 1990.
- 66- **Instruction manual for JL Series.**, Chyo YMC Co. Ltd, Kyoto.
- 67- FEICHAS, M.B. **Prática de ajustamento estatístico de curvas matemáticas.** Rio de Janeiro: Eletrobrás, 1980.
- 68- TRIBOUT, M., PAREDES, S. AND LÉONIS, J. Investigation of ribonuclease-catalysed kinetics by a micro-calorimetric method. **Biochem. J.**, v.153, p.89-91, 1976.

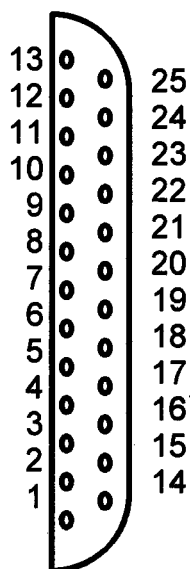
- 69- MOUNTCASTLE, D.B., FREIRE, E. AND BILTONEN, R.L. Generation of continuous ligand-macromolecule binding isotherms. Use of exponential concentration gradients. **Biopolymers.**, v.15, p.355-371, 1976.
- 70- WILKINSON, M.C. AND KIDWELL, R.L. A mathematical description of the Harkins and Brown correction curve for the determination of surface and interfacial tensions. **J. Colloid Interface Sci.**, v.35, p.114-119, 1971.
- 71- WILKINSON, M.C., Extended use of, and comments on, the drop-weight (drop-volume) technique for the determination of surface and interfacial tensions. **J. Colloid Interface Sci.**, v.40, p.14-26, 1972.
- 72- LIDE, R.D. (ed).; **Handbook of chemistry and physics.** 76ed. New York: CRC Press., 1995.
- 73- STANDARD METHODS FOR EXAMINATION OF WATER AND WASTE WATER. 18ed. Washington: Apha, 1992.
- 74- LEHMKUHL, A.; GANZAROLLI, E.M.; SOUZA, I.G DE; NOME, F. Sistema de termostatização para aplicação em análise por injeção em fluxo. **Química Nova.** aceito para publicação.
- 75- PIERSON, F.W.; WHITAKER, S. Studies of the drop-weight for surfactant solutions. ii. experimental results for water and surfactant solutions. **J. Colloid Interface Sci.**, v.54, n.2, p.219-230, 1975.
- 76- JHO, C.; BURKE, R. Drop weight technique for the measurement of dynamic surface tension, **J. Colloid Interface Sci.**, v.95, n.1, p.61-71, 1983.
- 77- JHO, C.; CARRERAS, M. The effect of the drop weight technique for the measurement of dynamic surface tension. **J. Colloid Interface Sci.**, v.99, n.2, p.543-548, 1984.
- 78- HUNSEL, J.V.; BLEYS, G.; JOOS, P. Adsorption kinetics at the oil/water interface. **J. Colloid Interface Sci.**, v.114, n.2, p.432-441, 1986.
- 79- TAJIMA, K.; Radiotracer studies on adsorption of surface active substance at aqueous surface. ii. the effect of excess salt on the adsorption of sodium dodecylsulfate. **Bulletin of the Chemical Society of Japan**, v.43, p.3063-3066, 1970.

- 80- TAJIMA, K.; MURAMATSU, M.; SASAKI, T. Radiotracer studies on adsorption of surface active substance at aqueous surface. i. accurate measurement of adsorption of tritiated sodium dodecylsulfate. **Bulletin of the Chemical Society of Japan**, v.43, p.1991-1998, 1970.
- 81- JAYSON, G.G.; THOMPSON, G. Adsorption studies of tritiated surfactants at the gas-liquid interface for a nonionic-anionic mixed surfactant system. **J. Colloid Interface Sci.**, v.111, n.1, p.65-74, 1986.
- 82- LU, J.R.; MARROCCO, A.; SU, T.J.; THOMAS R.K.; PENFOLD, J. Adsorption of dodecyl sulfate surfactants with monovalent metal counterions at the air-water interface studied by neutron reflection and surface tension. **J. Colloid Interface Sci.**, v.158, p.303-316, 1993.
- 83- ZANA, R.; YIV, S.; STRAZIELLE, C.; LIANOS, P. **J. Colloid Interface Sci.**, v.80, p.208, 1981.
- 84- OGINO, K.; ABE, M. Mixed surfactant systems in: Schick, M.J.; Fowkes, F.M. **Surfactant science series**, New York: Marcel Dekker Inc, v.46, p.271, 1993.
- 85- SVENSSON, E; GUDMUNDSSON, M.; ELIASSON, A-C. Binding of sodium dodecylsulphate to starch polysaccharides quantified by surface tension measurements. **Colloids and Surfaces B**, v.6, p.227-233, 1996.
- 86- TORNBERG, E. The application of the drop volume technique to measurements of the adsorption of proteins at interfaces. **J. Colloid Interface Sci.** v.64, n.3, p.391-402, 1978.
- 87- FRANSES, E.I.; BASARAN, O.A.; CHANG, C.-H.; Techniques to measure dynamic surface tension. **Current Opinion in Colloid & Interface Science**, v.1, p.296-303, 1996.

Apêndices

APÊNDICE I

CONECTOR DB-25



conector
DB-25

PINO DB-25	INTERFACE PARALELA MQI 12/8PCC - CN1	INTERFACE PARALELA DA IMPRESSORA	INTERFACE SERIAL DB-25	INTERFACE SERIAL 9 PINOS
1	bit 0 (R) porta C	strobe	-	-
2	bit 1 (R) porta C	bit 0	TxD	3
3	bit 2 (R) porta C	bit 1	RxD	2
4	+5V	bit 2	RTS	7
5	bit 7 (R) porta B	bit 3	CTS	8
6	bit 6 (R) porta B	bit 4	DSR	6
7	bit 5 (R) porta B	bit 5	Terra	5
8	bit 4 (R) porta B	bit 6	DCD	1
9	bit 3 (R) porta B	bit 7	-	-
10	bit 2 (R) porta B	ack	-	-
11	bit 1 (R) porta B	busy	-	-
12	bit 0 (R) porta B	paper out	-	-
13	terra (retorno)	selected	-	-
14	bit 5 (R) porta C	gnd	-	-
15	bit 4 (R) porta C	error	-	-
16	bit 3 (R) porta C	reset	-	-
17	+5V	-	-	-
18	bit 7 (T) porta B	-	-	-
19	bit 6 (T) porta B	gnd	-	-
20	bit 5 (T) porta B	gnd	DTR	4
21	bit 4 (T) porta B	gnd	-	-
22	bit 3 (T) porta B	gnd	RI	9
23	bit 2 (T) porta B	gnd	-	-
24	bit 1 (T) porta B	gnd	-	-
25	bit 0 (T) porta B	gnd	-	-

Anexo 1 - Conector DB-25. A lista à direita indica a função de cada pino do conector DB-25 utilizado em uma placa paralela MQI12/8PCC da Microquímica Ind. e Com., no conector CN1, em uma placa paralela de impressora e em uma placa serial. A coluna da direita indica a equivalência do conector DB-25 em um adaptador de 9 pinos utilizado para comunicação serial.

APÊNDICE II

PROGRAMA DO TENSÍOMETRO

```

{$M 40000, 0, 65535}
program tensiometro;

uses unmtm, untela, utiarq, utitens, unirqmqi, crt, dos, utiser, ungrafic,
    utisttc, graph, UNRS232, uncrt, utitexto, utigrtns, unregres,
    unstring, untecl;
const maxeixox=500;

procedure cadastra_raio;
var volume:real;
begin
    ch:=#0;
    repeat
        prepara_tela(1,4,1,3,");
        move_bomba;
        if (ch = #13) then adquiere_passo('CAL-
RAIO.TCR',oldct,gota,volume);
        if (ch = #13) then begin
            if ((gota>0) and (volume>0)) then begin
                calcula_raio(volume/gota);
                erro('REPETIR CALIBRACAO ? [S/N]',ch,true);
            end
            else erro('NENHUMA GOTA DETECTADA.',ch,false);
        end
        else erro('PROCEDIMENTO ABORTADO.',ch,false);
    until (ch in ['N',#27]);
end;

procedure move_bomba_ensaio;
interrupt;
begin
    inc(countera);
    fim:=port[$302];
    if (countera>=ctaa) then begin
        countera:=0;
        programa_porta;
        port[porta]:=bobina[ct1 and 3];
        inc(ct1);
    end;
    if (not pingou) then begin
        if ((fim and $1)=$1) then begin
            oldadostr:=dadostr;
            ctaa:=maxvalor;
            pingou:=true;
            numct:=ct1-oldct;
            oldct:=ct1;
        end;
    end
    else if (pingou) then begin
        inc(aguarde);
        if ((aguarde>=taxa) and ((fim and $1)=0)) then begin
            plotar:=true;
            pingou:=false;
        end;
    end;
    port[$20]:=$20;
end;

procedure ensaio_concentracao_linear_simulada;
var
tensaop,raiz,fator,concp,concm,vpg,mpg,massa,oldmassa,newvtg:re
al;
    oldvtg,vtg,oldconc,vpgres,olddens,dens,cmtg,cstg:real;
    Cser,cmis,vmis:real;
    lendo,xplot,yplot,code,oldcta,tamdec:word;
    f:text;
    aux,arq:string;
begin
    aux:='ENSAIO VARIÁVEL LINEAR SIMULADA';
    gota:=0;
    vtg:=0;

    oldvtg:=0;
    oldconc:=0;
    concm:=0;
    oldcta:=ctaa;
    Cser:=par[1];
    Cmis:=par[2];
    Vmis:=par[3];
    vpgres:=0;
    tamdec:=7;
    dens:=par[5];
    lendo:=0;
    aguarde:=0;
    pingou:=false;
    plotar:=false;
    parametros;
    if (ch=#27) then exit;
    prepara_tela(1,2,1,2,");
    texto_crt(10,10,40,'NOME DO COMPOSTO = ',aux,ch,1,7);
    if (ch = #27) then exit;
    prepara_tela(1,0,1,3,aux);
    arq:=get_dir_name;
    abre_arquivo(f,arq,'ns','NOME DO ARQUIVO : ',ch);
    if (ch = #27) then exit;
    if (ch<>'A') then begin
        writeln(f,' CONC. TENSAO VOL. ');
        writeln(f,aux);
        writeln(f,par[10]:5:1);
        writeln(f,round(par[11]));
    end;
    fecha_arquivo(f);
    prepara_tela(1,0,1,3,aux);
    move_bomba;
    if (ch<>#13) then begin erase(f);exit;end;
    ch:=#0;
    if (par[2]<par[1]) then begin
        if (par[2]=0) then cmtg:=int(log(par[1] + (par[2]-par[1]) *
EXP(-0.04/par[3]),10))-1
        else if (frac(log(1/par[2],10)) = 0) then cmtg:=log(par[2],10)
        else cmtg:=int(log(par[2],10)-1);
        if (frac(log(1/par[1],10)) = 0) then cstg:=log(par[1],10)
        else cstg:=int(log(par[1],10) + 1);
    end
    else begin
        if (par[1]=0) then cmtg:=int(log(par[2]+(par[1]-
par[2])*EXP(-0.04/par[3]),10))-1
        else if (frac(log(1/par[1],10)) = 0) then cmtg:=log(par[1],10)
        else cmtg:=int(log(par[1],10)-1);
        if (frac(log(1/par[2],10)) = 0) then cstg:=log(par[2],10)
        else cstg:=int(log(par[2],10) + 1);
    end;
    if (not liga_balanca) then exit;
    telagrfgahi(cmtg,0,cstg,80,5,8,10,5,'TENSÍOMETR
O',aux
        ,LOG [CONC(Molar)],'TENSAO
(Dinas/cm),'[ESC] ABORTA');
    settxtstyle(0,0,0);
    porta_bobina;
    porta:=$301;
    falhou:=false;
    {$I-}
    inicia_comunicacao(com2,1200,'n',1,8,T);
    inicia_aquisicao(@move_bomba_ensaio);
    repeat
        if (pr<>1) then lendo:=0 else inc(lendo);
        case ch of
            '+' : if (ctaa>taxa div 250) then ctaa:=ctaa div 2;
            '-' : if (ctaa<taxa/0.1) then ctaa:=ctaa*2;
            #32 : ctaa:=oldcta;
        end;
        ch:=#0;
        if (fim and fimall <> fimall) then falha;

```

```

if keypressed then ch:=upcase(readkey);
if (plotar) then begin
  inc(gota);
  aguarde:=0;
  ch:=#0;
  plotar:=false;
  if (dadostr[length(dadostr)]<>'G') then begin
    repeat
      countera:=0;
      until (dadostr[length(dadostr)]='G');
      aux:=dadostr;
      massa:=trans_dado_massa(aux);
      oldmassa:=trans_dado_massa(olddadostr);
      mpg:=massa-oldmassa;
    end else mpg:=0;
    if (mpg>0) then raiz:=par[7]/exp((ln(mpg/dens))/3) else
raiz:=0;
    if (raiz>0.3) and (raiz<1.2) then begin
      porta:=porta xor 1;
      repeat
        olddens:=dens;
        vpg:=mpg/dens;
        vtg:=oldvtg+vpg;
        raiz:=par[7]/exp((ln(vpg))/3);
        if (raiz>0.65) and (raiz<0.95) then
          Fator:=0.9267-
0.7694*raiz+0.4512*raiz*raiz
          else Fator:=0.9054-
0.7249*raiz+0.4293*raiz*raiz;
        Tensaop:=(mpg*par[12])/(2*pi*par[7]*Fator);
        if (tensaop>80) then tensaop:=80;
        if (odd(gota)) then begin
          concp:=par[1] + Vmis*(par[1]-Concm)*(exp(-
vpg/Vmis)-1)/vpg;
          concm:=par[1] + (Cmis-par[1])*(exp(-vpg/Vmis));
        end
        else concp:=concm;
          newvtg:=(vpg/fator)-vpgres;
          newvtg:=newvtg*concp;
          concp:=newvtg+(oldconc*vpgres);
          concp:=concp*fator/vpg;
          dens:=par[13] + par[14]*concp +
par[15]*concp*concp;
        until ((olddens-dens)<0.00001) and ((olddens-dens)>-
0.00001);
        oldconc:=concp;
        if (odd(gota)) then Cmis:=concm else Vmis:=Vmis-vpg;
        vpgres:=vpg*(1-fator)/fator;
        oldvtg:=vtg;
        xplot:=round(X2-((X2-X1)*(cstg-
log(CONCP,10))/(cstg-cmtg)));
        yplot:=round(Y1 + (TENSAOP - 80) * (Y2 - Y1) / (0 -
80));
        putpixel(xplot,yplot,15);
        setfillstyle(1,0);
        bar(250,470,500,478);
        setfillstyle(1,15);
        outtextxy(250,470,strings(gota,3,0)+strings(vpg,8,4)+
strings(tensaop,7,2)+strings(concp,tamdec+3,tamdec));
        append(f);
        writein(f,concp:tamdec+2:tamdec,'',tensaop:5:2,
',vpg:6:4);
      close(f);
    end
    else dec(gota);
      ctaa:=oldcta;
    end;
  until ((ch=#27) or (falhou) or (lendo>3000) or (par[3]/Vmis
>= 5));
  termina_aquisicao_bombas;
  termina_comunicacao;
  if (gota<=0) then erase(f);
  {$i+}
  ctaa:=oldcta;
  msggrf(getmaxx*2 div 5,getmaxy-10,0,
);
  setpntlognrmtns(cmtg,0,cstg,80,par[10],round(par[11]));

```

```

closegraph;
end;

procedure ensaio_concentracao_exponencial;
var tensaop,raiz,fator,concp,vpg,mpg,masa,oldmassa,newvtg:real;
    vtg,oldvtg,oldconc,vpgres,olddens,dens,cmtg,cstg:real;
    lendo,xplot,yplot,code,oldcta,tamdec:word;
    f:text;
    aux,arq:string;
begin
  aux:='ENSAIO VARIÁVEL EXPONENCIAL';
  gota:=0;
  vtg:=0;
  oldvtg:=0;
  oldconc:=0;
  oldcta:=ctaa;
  vpgres:=0;
  tamdec:=7;
  dens:=par[5];
  lendo:=0;
  aguarde:=0;
  pingou:=false;
  plotar:=false;
  parametros;
  if (ch=#27) then exit;
  prepara_tela(1,2,1,2,);
  texto_cri(10,10,40,'NOME DO COMPOSTO = ',aux,ch,1,7);
  if (ch = #27) then exit;
  prepara_tela(1,0,1,3,aux);
  arq:=get_dir_name;
  abre_arquivo(f,arq,'ns','NOME DO ARQUIVO : ','?',ch);
  if (ch = #27) then exit;
  if (ch<>'A') then begin
    writeln(f,' CONCENTRAÇÃO TENSÃO VOL. ');
    writeln(f,aux);
    writeln(f,par[10]:5:1);
    writeln(f,round(par[11]));
  end;
  fecha_arquivo(f);
  prepara_tela(1,0,1,3,aux);
  move_bomba;
  if (ch<>#13) then begin erase(f);exit;end;
  ch:=#0;
  if (par[2]<par[1]) then begin
    if (par[2]=0) then cmtg:=int(log(par[1] + (par[2]-par[1]) *
EXP(-0.04/par[3]),10))-1
    else if (frac(log(1/par[2],10)) = 0) then cmtg:=log(par[2],10)
    else cmtg:=int(log(par[2],10))-1;
    if (frac(log(1/par[1],10)) = 0) then cstg:=log(par[1],10)
    else cstg:=int(log(par[1],10) + 1);
  end
  else begin
    if (par[1]=0) then cmtg:=int(log(par[2]+(par[1]-
par[2])*EXP(-0.04/par[3]),10))-1
    else if (frac(log(1/par[1],10)) = 0) then cmtg:=log(par[1],10)
    else cmtg:=int(log(par[1],10))-1;
    if (frac(log(1/par[2],10)) = 0) then cstg:=log(par[2],10)
    else cstg:=int(log(par[2],10) + 1);
  end;
  if (not liga_balanca) then exit;
  telagrfgahi(cmtg,0,cstg,80,5,8,10,5,'TENSÍOMETRO',aux
,LOG [CONC(Molar)],TENSAO
(Dinas/cm),'[ESC] ABORTA');
  settxtstyle(0,0,0);
  porta_bobina;
  falhou:=false;
  {$i-}
  inicia_comunicacao(com2,1200,'n',1,8,T);
  inicia_aquisicao(@move_bomba_ensaio);
  repeat
    if (pr<>1) then lendo:=0 else inc(lendo);
    case ch of
      '+' : if (ctaa>taxa div 250) then ctaa:=ctaa div 2;
      '.' : if (ctaa<taxa*5) then ctaa:=ctaa*2;
      #32 : ctaa:=oldcta;
    end;

```

```

ch:=#0;
if (fim and fimall <> fimall) then falha;
if keypressed then ch:=upcase(readkey);
if (plotar) then begin
  inc(gota);
  aguarde:=0;
  ch:=#0;
  plotar:=false;
  if (dadostr[length(dadostr)]<>'G') then begin
    repeat
      countera:=0;
      until (dadostr[length(dadostr)]='G');
      aux:=dadostr;
      massa:=trans_dado_massa(aux);
      oldmassa:=trans_dado_massa(olddadostr);
      mpg:=massa-oldmassa;
    end else mpg:=0;
    if (mpg>0) then raiz:=par[7]/exp((ln(mpg/dens))/3) else
raiz:=0;
    if (raiz>0.3) and (raiz<1.2) then begin
      repeat
        olddens:=dens;
        vpg:=mpg/dens;
        raiz:=par[7]/exp((ln(vpg))/3);
        if (raiz>0.65) and (raiz<0.95) then Fator:=0.9267-
0.7694*raiz+0.4512*raiz*raiz
        else Fator:=0.9054-0.7249*raiz+0.4293*raiz*raiz;
        Tensaop:=(mpg*par[12])/(2*pi*par[7]*Fator);
        if (tensaop>80) then tensaop:=80;
        vtg:=oldvtg+vpg;
        concp:=par[1]+par[3]*(par[1]-par[2])*(exp(-
vtg/par[3])-exp(-oldvtg/par[3]))/vpg;
        newvtg:=(vpg/fator)-vpgres;
        newvtg:=newvtg*concp;
        concp:=newvtg+(oldconcp*vpgres);
        concp:=concp*fator/vpg;
        dens:=par[13]+par[14]*concp+
par[15]*concp*concp;
        until ((olddens-dens)<0.00001) and ((olddens-dens)>
0.00001);
        oldvtg:=vtg;
        oldconcp:=concp;
        vpgres:=vpg*(1-fator)/fator;
        xplot:=round(X2-((X2-X1)*(cstg-
log(CONCP,10))/(cstg-cmtg)));
        yplot:=round(Y1+(TENSAOP-80)*(Y2-Y1)/(0-
80));
        putpixel(xplot,yplot,15);
        setfillstyle(1,0);
        bar(250,470,500,478);
        setfillstyle(1,15);
        outtextxy(250,470,strings(gota,3,0)+strings(vpg,8,4)+
strings(tensaop,7,2)+strings(concp,tamdec+3,tamdec));
        append(f);
        writeln(f,concp,tamdec+2:tamdec,',tensaop:5:2,'
',vpg:6:4);
        close(f);
        end
      else dec(gota);
        ctaa:=oldcta;
        end;
      until ((ch=#27) or (falhou) or (lendo>3000));
      termina_aquisicao_bombas;
      termina_comunicacao;
      if (gota<=0) then erase(f);
      {$i+}
      ctaa:=oldcta;
      msggrf(getmaxx*2 div 5,getmaxy-10,0,
      setptlognrmtns(cmtg,0,cstg,80,par[10],round(par[11]));
      closegraph;
    end;

procedure ensaio_simples;
var
desvio,raiz,fator,vpg,mpg,massa,oldmassa,tensaom,vtg,oldvtg,soma:
real;

```

```

lendo,i,xplot,yplot,code,oldcta,X1g,X2g:word;
tensaop:array [1..100] of real;
f:text;
aux,arq:string;
begin
  parametros;
  if (ch=#27) then exit;
  prepara_tela(1,8,1,1,');
  aux:=ENSAIO SIMPLES';
  texto_crt(10,10,40,NOME DO COMPOSTO =',aux,ch,1,7);
  if (ch=#27) then exit;
  arq:=get_dir_name;
  prepara_tela(1,0,1,1,aux);
  abre_arquivo(f,arq,'TSP',NOME DO ARQUIVO :',?',ch);
  if (ch=#27) then exit;
  writeln(f,'TENSAO');
  writeln(f,aux);
  writeln(f,par[10]:5:1);
  fecha_arquivo(f);
  prepara_tela(1,0,1,3,aux);
  move_bomba;
  if (ch<>#13) then begin erase(f);exit;end;
  ch:=#0;
  deltact:=maxvalor;
  ct1:=0;
  oldct:=0;
  tensaom:=0;
  vtg:=0;
  lendo:=0;
  oldvtg:=0;
  gota:=0;
  oldcta:=round(taxa/par[8]);
  ctaa:=round(taxa/par[16]);
  aguarde:=0;
  pingou:=false;
  plotar:=false;
  if (not liga_balanca) then exit;
  telagrfgahi(0,0,100,80,5,8,10,5,'TENSÍOMETRO',aux,
  'No. PONTOS',TENSAO (Dinas/cm)',[ESC];
  ABORTA);
  settxtstyle(0,0,0);
  X1g:=(getmaxx+2*X2) div 3;
  X2g:=(2*getmaxx+X2) div 3;
  porta_bobina;
  inicia_comunicacao(com2,1200,'n',1,8,T);
  inicia_aquisicao(@move_bomba_ensaio);
  repeat
    if (pr<>1) then lendo:=0 else inc(lendo);
    case ch of
      #13 : deltact:=ct1-oldct;
      '*': begin
        deltact:=deltact+(deltact div 10);
        ctaa:=round(taxa/par[16]);
        end;
      '/' : deltact:=deltact-(deltact div 10);
      #32 : ctaa:=oldcta;
      '+' : if (ctaa>taxa div 250) then ctaa:=ctaa div 2;
      '.' : if (ctaa<taxa/0.1) then ctaa:=ctaa*2;
    end;
    ch:=#0;
    if keypressed then ch:=upcase(readkey);
    if (fim and fimall <> fimall) then falha;
    setfillstyle(1,0);
    bar(450,470,600,478);
    setfillstyle(1,15);
    outtextxy(450,470,strings(ct1-
oldct,6,0)+strings(deltact,6,0)+strings(taxa/ctaa,6,2));
    if ((ct1-oldct) > deltact) then ctaa:=oldcta;
    if (plotar) then begin
      oldct:=ct1;
      inc(gota);
      aguarde:=0;
      plotar:=false;
      if (dadostr[length(dadostr)]<>'G') then begin
        repeat
          until (dadostr[length(dadostr)]='G');
          aux:=dadostr;

```

```

ctaa:=round(taxa/par[16]);
massa:=trans_dado_massa(aux);
oldmassa:=trans_dado_massa(olddostr);
mpg:=massa-oldmassa;
end else mpg:=0;
if (mpg>0) then raiz:=par[7]/exp((ln(mpg/par[4]))/3) else
raiz:=0;
if (raiz>0.3) and (raiz<1.2) then begin
vpg:=mpg/par[4];
vtg:=vtg+vpg;
raiz:=par[7]/exp((ln(vpg))/3);
if (raiz>0.65) and (raiz<0.95) then Fator:=0.9267-
0.7694*raiz+0.4512*raiz*raiz
else Fator:=0.9054-0.7249*raiz+0.4293*raiz*raiz;
Tensaop[gota]:=(mpg*par[12])/(2*pi*par[7]*Fator);
if (tensaop[gota]>80) then tensaop[gota]:=80;
xplot:=round(X2-((X2-X1)*(100-gota)/(100-0)));
yplot:=round(Y1+(TENSAOP[gota]-80)*(Y2-Y1)
/(0-80));
calcula_media(tensaop,gota,tensaom,desvio,soma);
bar(xplot+1,Y2,xplot-1,yplot);
setfillstyle(1,0);
bar(200,470,500,478);
bar(X1g,Y2+22,getmaxx-1,Y1);
setfillstyle(1,15);

outtextxy(200,470,strings(gota,3,0)+strings(vpg,8,4)+strings(te
nsaop[gota],6,2));
outtextxy(X1g+1,Y2+8,strings(tensaom,5,2));
outtextxy(X1g+1,Y2+16,chr(241)+strings(desvio,4,2));
append(f);
writeln(f,tensaop[gota]:5:2);
close(f);
bar(X1g,Y2,X2g,round(Y1+(TENSAOM-80)*(Y2-
Y1)/(0-80)));
end
else dec(gota);
end;
until ((ch=#27) or (fim and $06 <> $06) or (gota=par[9]) or
(lendo>3000));
termina_aquisicao_bombas;
termina_comunicacao;
if (gota > 0) then begin
calcula_media(tensaop,gota,tensaom,desvio,soma);
teste_discordancia(tensaop,gota,1);
append(f);
writeln(f,tensaom:7:4,desvio:8:4);
close(f);
end
else erase(f);
ctaa:=oldcta;
ch:=leteclado(true);
closegraph;
end;

procedure ensaio_fluxo;
const ntotvel=15;
var
tensaom,desvio,raiz,fator,vpg,mpg,massa,oldmassa,vtg,oldvtg,soma:
real;
qtddpto,gotatot,lendo,i,xplot,yplot:word;
ptoatual,code,oldcta,X1g,X2g:word;
velocidade:array [1..ntotvel] of real;
tensaop:array [1..maxeixox div 5] of real;
f:text;
aux,aux2,arq:string;
velrapido:boolean;
begin
qtddpto:=0;
prepara_tela(1,8,1,1,['F3] PROSSEGUE');
repeat
inc(qtddpto);
repeat
aux:=';
str(qtddpto,aux2);
aux2:='VELOCIDADE DA BOMBA NA CORRIDA
'+aux2+'=';

```

```

texto_crt(10,6+qtddpto,6,aux2,aux,ch,1,7);
val(aux,velocidade[qtddpto],code);
until ((velocidade[qtddpto]>0) and (code=0)) or (ch in
[#27,#61]);
until (ch in [#27,#61]) or (qtddpto=ntotvel);
if (ch=#27) then exit;
prepara_tela(1,8,1,1,');
aux:='ENSAIO SIMPLES';
texto_crt(10,10,40,'NOME DO COMPOSTO = ',aux,ch,1,7);
if (ch=#27) then exit;
arq:=get_dir_name;
prepara_tela(1,0,1,1,aux);
abre_arquivo(f,arq,'TSF','NOME DO ARQUIVO : ','?',ch);
if (ch=#27) then exit;
writeln(f,'TENSAO');
writeln(f,aux);
writeln(f,par[10]:5:1);
fecha_arquivo(f);
prepara_tela(1,0,1,3,aux);
move_bomba;
if (ch<=#13) then begin erase(f);exit;end;
if (not liga_balanca) then exit;
telagrfrvahi(0,0,maxeixox,80,5,8,10,5,'TENSIO METR
O',aux,
'No. PONTOS','TENSAO (Dinas/cm)',[ESC]
ABORTA);
settextstyle(0,0,0);
X1g:=(getmaxx+2*X2) div 3;
X2g:=(2*getmaxx+X2) div 3;
ch:=#0;
deltact:=50;
oldcta:=round(taxa/par[8]);
if (taxa/par[16])<(taxa/velocidade[1]) then
ctaa:=round(taxa/par[16])
else ctaa:=round(taxa/velocidade[1]);
ct1:=0;
oldct:=0;
vtg:=0;
numct:=0;
lendo:=0;
oldvtg:=0;
gota:=0;
velrapido:=true;
gotatot:=0;
aguarde:=0;
ptoatual:=1;
pingou:=false;
plotar:=false;
porta_bobina;
{$i}
inicia_comunicacao(com2,1200,'n',1,8,'T');
inicia_aquisicao(@move_bomba_ensaio);
repeat
if keypressed then ch:=upcase(readkey);
if (ch=#13) then begin
deltact:=ct1-oldct;
ch:=#0;
end;
if (fim and fimall <> fimall) then falha;
if (velrapido and ((ct1-oldct) > deltact)) then begin
velrapido:=false;
ctaa:=round(taxa/velocidade[ptoatual]);
end;
setfillstyle(1,0);
bar(370,470,600,478);
setfillstyle(1,15);
outtextxy(370,470,strings(gotatot,6,0)+strings(ct1-
oldct,6,0)+strings(deltact,6,0)+strings(taxa/ctaa,6,2));
if (plotar) then begin
aguarde:=0;
plotar:=false;
if (dadostr[length(dadostr)]<>'G') then begin
repeat
countera:=0;
until (dadostr[length(dadostr)]='G');
aux:=dadostr;
massa:=trans_dado_massa(aux);

```

```

    oldmassa:=trans_dado_massa(olddadostr);
    mpg:=massa-oldmassa;
end else mpg:=0;
if (mpg>0) then raiz:=par[7]/exp((ln(mpg/par[4]))/3) else
raiz:=0;
if (raiz>0.3) and (raiz<1.2) then begin
    oldct:=ct1;
    inc(gotatot);
    inc(gota);
    velrapido:=true;
    vpg:=mpg/par[4];
    vtg:=vig+vpg;
    raiz:=par[7]/exp((ln(vpg))/3);
    if (raiz>0.65) and (raiz<0.95) then Fator:=0.9267-
0.7694*raiz+0.4512*raiz*raiz
    else Fator:=0.9054-0.7249*raiz+0.4293*raiz*raiz;
    Tensaop[gota]:=(mpg*par[12])/(2*pi*par[7]*Fator);
    if (tensaop[gota]>80) then tensaop[gota]:=80;
    xplot:=round(X2 - (((X2-X1)/(maxeixox-
0))*(maxeixox-gotatot)));
    yplot:=round(Y1 + (TENSAOP[gota] - 80) * (Y2 - Y1)
/(0 - 80));
    line(xplot,Y2,xplot,yplot);
    setfillstyle(1,0);
    bar(200,470,500,478);
    setfillstyle(1,15);
    outtextxy(200,470,strings(gota,3,0)+strings(vpg,8,4)+strings(te
nsaop[gota],6,2)+ strings(numct,4,0));
    append(f);
    writeln(f,tensaop[gota]:5:2,',vpg:6:4',
'velocidade[ptoatual]:5:2,',numct);
    close(f);
    if (gota=par[9]) then begin
        calcula_media(tensaop,gota,tensaom,desvio,soma);
        append(f);
        writeln(f,tensaom:5:2,',desvio:5:2);
        close(f);
        gota:=0;
        inc(ptoatual);
    end;
end;
if (ptoatual<<qtddpto) then begin
    if (taxa/par[16])<(taxa/velocidade[ptoatual]) then
ctaa:=round(taxa/par[16])
    else ctaa:=round(taxa/velocidade[ptoatual]);
    end;
end;
until ((ch=#27) or (falhou) or (ptoatual=qtddpto));
termina_aquisicao_bombas;
termina_comunicacao;
if (gotatot <= 0) then erase(f);
{$i+}
ctaa:=oldcta;
avisa_fim(200,500,5000);
closegraph;
end;

procedure move_bomba_ensaio_duas_bombas;
interrupt;
begin
    inc(countera);
    inc(counterb);
    fim:=port[$302];
    if (countera>=ctaa) then begin
        countera:=0;
        programa_porta;
        port[$301]:=bobina[ct1 and 3];
        inc(ct1);
    end;
    if (counterb>=ctab) then begin
        counterb:=0;
        programa_porta;
        port[$300]:=bobina[ct2 and 3];
        inc(ct2);
    end;
    if (not pingou) then begin
        if ((fim and $1)=$1) then begin

```

```

        olddadostr:=dadostr;
        ctaa:=maxvalor;
        ctab:=maxvalor;
        pingou:=true;
        numct:=ct1-oldct;
        oldct:=ct1;
    end;
end
else if (pingou) then begin
    inc(aguarde);
    if ((aguarde>=taxa) and ((fim and $1)=0)) then begin
        plotar:=true;
        pingou:=false;
    end;
end;
port[$20]:=$20;
end;

procedure ensaio_variavel_duas_bombas;
var
    vtg,oldconc,vpgres,dens,cmtg,cstg,masa,oldmassa:real;
    mpg,raiz,fator,olddens,vpg,tensaop,concm,newvtg,concp:real;
    tamdec,lendo,xplot,yplot:word;
    f:text;
    aux,arq:string;

procedure calc_counter(pto:byte);
begin
    ctaa:=round(taxa/(par[8]*pto/100));
    ctab:=round(taxa/(par[8]*(100-pto)/100));
    concm:=((par[1]*ctab)+(par[2]*ctaa))/(ctaa+ctab);
end;

begin
    aux:='ENSAIO VARIAVEL DUAS BOMBAS';
    gota:=0;
    vtg:=0;
    oldconc:=0;
    vpgres:=0;
    tamdec:=7;
    dens:=par[5];
    lendo:=0;
    aguarde:=0;
    countera:=0;
    counterb:=0;
    pingou:=false;
    plotar:=false;
    parametros;
    if (ch=#27) then exit;
    calc_counter(1);
    prepara_tela(1,2,1,2,');
    texto_crt(10,10,40,'NOME DO COMPOSTO = ',aux,ch,1,7);
    if (ch = #27) then exit;
    prepara_tela(1,0,1,3,aux);
    arq:=get_dir_name;
    abre_arquivo(f,arq,'.tms','NOME DO ARQUIVO : ',?',ch);
    if (ch = #27) then exit;
    if (ch<>'A') then begin
        writeln(f,' CONC. TENSAO VOL. ');
        writeln(f,aux);
        writeln(f,par[10]:5:1);
        writeln(f,round(par[11]));
    end;
    fecha_arquivo(f);
    prepara_tela(1,0,1,3,aux);
    move_bomba;
    if (ch<>#13) then begin erase(f);exit;end;
    ch:=#0;
    if (par[2]<par[1]) then begin
        if (par[2]=0) then cmtg:=int(log(par[1] + (par[2]-par[1]) *
EXP(-0.04/par[3]),10))-1
        else if (frac(log(1/par[2],10)) = 0) then cmtg:=log(par[2],10)
        else cmtg:=int(log(par[2],10)-1);
        if (frac(log(1/par[1],10)) = 0) then cstg:=log(par[1],10)
        else cstg:=int(log(par[1],10) + 1);
    end
    else begin

```



```

if (par[1]=0) then cmtg:=int(log(par[2]+(par[1]-
par[2])*EXP(-0.04/par[3]),10))-1
else if (frac(log(1/par[1],10)) = 0) then cmtg:=log(par[1],10)
else cmtg:=int(log(par[1],10)-1);
if (frac(log(1/par[2],10)) = 0) then cstg:=log(par[2],10)
else cstg:=int(log(par[2],10) +1);
end;
if (not liga_balanca) then exit;
telagrfgahi(cmtg,0,cstg,80,5,8,10,5,T E N S I O M E T R
O',aux,'LOG [CONC(Molar)],'TENSAO (Dinas/cm)',[ESC]
ABORTA');
settextstyle(0,0,0);
porta_bobina;
falhou:=false;
{$i-}
inicia_comunicacao(com2,1200,'n',1,8,T);
inicia_aquisicao(@move_bomba_ensaio_duas_bombas);
repeat
if (pr<>1) then lendo:=0 else inc(lendo);
if (fim and fimall <> fimall) then falha;
if keypressed then ch:=upcase(readkey);
if (plotar) then begin
aguarde:=0;
ch:=#0;
plotar:=false;
if (dadostr[length(dadostr)]<>'G') then begin
repeat
countera:=0;
until (dadostr[length(dadostr)]='G');
aux:=dadostr;
massa:=trans_dado_massa(aux);
oldmassa:=trans_dado_massa(olddadostr);
mpg:=massa-oldmassa;
end else mpg:=0;
if (mpg>0) then raiz:=par[7]/exp((ln(mpg/dens))/3) else
raiz:=0;
if (raiz>0.3) and (raiz<1.2) then begin
inc(gota);
repeat
olddens:=dens;
vpg:=mpg/dens;
raiz:=par[7]/exp((ln(vpg))/3);
if (raiz>0.65) and (raiz<0.95) then Fator:=0.9267-
0.7694*raiz+0.4512*raiz*raiz
else Fator:=0.9054-0.7249*raiz+0.4293*raiz*raiz;
Tensaop:=(mpg*par[12])/(2*pi*par[7]*Fator);
if (tensaop>80) then tensaop:=80;
newvtg:=(vpg/fator)-vpgres;
newwtg:=newvtg*concm;
concp:=newwtg+(oldconcp*vpgres);
concp:=concp*fator/vpg;
dens:=par[13] + par[14]*concp +
par[15]*concp*concp;
until ((olddens-dens)<0.00001) and ((olddens-dens)>-
0.00001);
oldconcp:=concp;
vpgres:=vpg*(1-fator)/fator;
xplot:=round(X2-((X2-X1)*(cstg-
log(CONCP,10))/(cstg-cmtg)));
yplot:=round(Y1+(TENSAOP-80)*(Y2-Y1)/(0-
80));
putpixel(xplot,yplot,15);
setfillstyle(1,0);
bar(250,470,600,478);
setfillstyle(1,15);
outtextxy(250,470,strings(gota,3,0))+strings(vpg,8,4)+
strings(tensaop,7,2)+strings(concp,tamdec+3,tamdec)+
strings(concm,tamdec+3,tamdec));
append(f);
writeln(f,concp:tamdec+2:tamdec,'',tensaop:5:2,'
',vpg:6:4);
close(f);
end;
calc_counter(gota+1);
end;
until ((ch=#27) or (falhou) or (lendo>3000));
termina_aquisicao_bombas;

```

```

termina_comunicacao;
if (gota<=0) then erase(f);
{$i+}
msggrf(getmaxx*2 div 5,getmaxy-10,0,');
setpntlognrmtns(cmtg,0,cstg,80,par[10],round(par[11]));
closegraph;
end;

```

```

procedure analise_concentracao;
var concaux,tensaooaux,excesaux,corr,desvio,maxconc,minconc,
maxexc,minexc,temper,nmol:real;
concx,tensaoy,excesso:array [1..1000] of real;
coefic:array [0..9] of real;
xplot,yplot:longint;
i,j,qtdd,qtddini,code,ordem:word;
f,fxc:text;
arq,aux,aux1,aux2:string;
isfirst,calculok,learquivo,trocanome:boolean;
option:char;

```

```

procedure make_conc_max_min(min,max:word);
var i:word;
begin
maxconc:=1.7E38;
minconc:=1.7E38;
for i:=min to max do begin
if (concx[i]>maxconc) then maxconc:=concx[i];
if (concx[i]<minconc) then minconc:=concx[i];
end;
if (frac(log(1/minconc,10)) = 0) then
minconc:=log(minconc,10)
else minconc:=int(log(minconc,10)-1);
if (frac(log(1/maxconc,10)) = 0) then
maxconc:=log(maxconc,10)
else maxconc:=int(log(maxconc,10));
end;

```

```

procedure make_exc_max_min(min,max:word);
var i:word;
begin
maxexc:=1.7E38;
minexc:=1.7E38;
for i:=min to max do begin
if (excesso[i]>maxexc) then maxexc:=excesso[i];
if (excesso[i]<minexc) then minexc:=excesso[i];
end;
if (frac(maxexc)<>0) then maxexc:=int(maxexc)+1;
if (frac(minexc)<>0) then minexc:=int(minexc)-1;
end;

```

```

procedure make_tela_grafica_plote(ptoinicio,ptofinal:word);
var i:word;
begin
telagrfgahi(minconc,0,maxconc,80,5,8,10,5,T E N S I O M E
T R O',aux1,
'LOG [CONC(Molar)],'TENSAO (Dinas/cm)',[ESC]
RETORNA');
msg_tela(568,64,0,','[F6]ELIMINA|PONTO|[F7]NOVO|
ARQ,|+ '|[F8]CORRIGE|CONC.|[F9]TENSAO|
[F10]REGRES.|POLINOM.|+
'|[ENTER]ISOTERMA|ADSORCAO|');
for i:=ptoinicio to ptofinal do begin
xplot:=round(X2-((X2-X1)*(maxconc-
log(CONCX[i],10))/(maxconc-minconc)));
yplot:=round(Y1+(TENSAOY[i]-80)*(Y2-Y1)/(0-
80));
putpixel(xplot,yplot,15);
end;
end;

```

```

procedure make_excesso;
var i:word;
dgamadlnc,incrm,newvalor:real;
begin
prepara_tela(1,5,1,1,');
arq:=';

```

```

abre_arquivo(fexc,arq,'exc',NOME DO ARQUIVO DE
EXCESSO:','?',option);
if (option<>#27) then begin
  writeln(fexc,'CONC.  EXC. ');
  writeln(fexc,aux1);
  writeln(fexc,temper:5:1);
  writeln(fexc,nmol:3:1);
  for i:=1 to ordem do coefic[i]:=i*coefic[i];
  increm:=concx[round(j*1.4)];
  increm:=increm/200;
  for i:=1 to 200 do begin
    newvalor:=increm*i;
    if (newvalor>=concx[1]) then begin
      dgamadlnc:=polinomio(log(newvalor,10),ordem-
1,coefic);
      dgamadlnc:=dgamadlnc/ln(10);
      dgamadlnc:=-dgamadlnc*1000/(8.314*temper*nmol);
      writeln(fexc,newvalor:9:7,',',dgamadlnc:6:2);
    end;
  end;
  close(fexc);
end;
ch:=F9;
end;

procedure make_fiting_excesso;
var i:word;
begin
  j:=0;
  repeat
    inc(j);
  until ((concx[j]>cdisp) or (j=qtdd));
  dec(j);
  for i:=1 to j do excesso[i]:=log(concx[i],10);
  regressao_polinomial(excesso,tensaoy,coefic,ordem,j,corr,desvi
o);
  if not calculook then ch:=F10
  else begin
    make_tela_grafica_plote(1,qtdd);
    isfirst:=true;
    cdisp:=cdisp/100;
    for i:=1 to 140 do begin
      concaux:=log(cdisp*i,10);
      if (concaux>=minconc) then begin
        tensaoaux:=polinomio(concaux,ordem,coefic);
        xplot:=round(X2- ((X2-X1)*(maxconc-
concaux))/(maxconc-minconc));
        yplot:=round(Y1 + (tensaoaux - 80) * (Y2 - Y1) / (0 -
80));
        if (not isfirst) then lineto(xplot,yplot)
        else begin
          moveto(xplot,yplot);
          isfirst:=false;
        end;
      end;
    end;
    msg_tela(576,320,0,['CORR.']+strings(corr,6,4)+
' ||DESVIO|'+strings(desvio,6,4));
    setpntlognrmtns(minconc,0,maxconc,80,temper,nmol);
    closegraph;
    if (ch=#13) then make_excesso;
  end;
end;

begin
  aux:="";
  ch:=F9;
  trocanome:=true;
  learquivo:=true;
  repeat
    if (learquivo) then begin
      learquivo:=false;
      prepara_tela(1,5,1,1,"");
      if (trocanome) then begin
        arq:=get_dir_name;
        abre_arquivo(f,arq,'tns',NOME DO ARQUIVO :
',R',ch);
        trocanome:=false;
      end;
      if (ch=#27) then exit;
      reset(f);
      readln(f,aux2);
      readln(f,aux1);
      readln(f,temper);
      readln(f,nmol);
      qtdd:=0;
      while not eof(f) do begin
        inc(qtdd);
        readln(f,concx[qtdd],tensaoy[qtdd]);
      end;
      fecha_arquivo(f);
      ch:=F9;
    end;
    if (ch=F6) then begin
      make_tela_grafica_plote(1,qtdd);
      closegraph;
      ch:=F7; {??? AQUI VAI A ROTINA DE ELIMINAR
PONTO???)
    end;
    if (ch=F7) then begin
      learquivo:=true;
      trocanome:=true;
    end;
    if (ch=F9) then begin
      make_conc_max_min(1,qtdd);
      make_tela_grafica_plote(1,qtdd);
      setpntlognrmtns(minconc,0,maxconc,80,temper,nmol);
      closegraph;
    end;
    if (ch=F8) then begin
      prepara_tela(1,5,1,1,"");
      troca_pto(arq,ch,qtdd);
      assign(f,arq);
      learquivo:=true;
      ch:=F9;
    end;
    if (ch=F10) then begin
      prepara_tela(1,5,1,1,"");
      repeat
        texto_crt(10,10,1,ORDEM DA REGRESSAO =
',aux,ch,1,7);
        val(aux,ordem,code);
        until (ch=#27) or ((code=0) and (ordem>0));
        if (ch=#27) then ch:=F9 else make_fiting_excesso;
      end;
    until (ch=#27);
  end;

procedure analise_simples;
var f
      : text;
      temp,dens,fluxo,desvio,media,soma
      : real;
      aux1,aux2,arq
      : string;
      xplot,yplot
      : longint;
      X1g,X2g
      : word;
      tensao:array [1..100] of real;
begin
  prepara_tela(1,10,1,1,"");
  arq:=get_dir_name;
  abre_arquivo(f,arq,'TSP',NOME DO ARQUIVO :',R',ch);
  if (ch=#27) then exit;
  readln(f,aux1);
  readln(f,aux1);
  readln(f,temp);
  telagrfgahi(0,0,100,80,5,8,10,5,TENSÍMETR
O',aux1,
      'No. PONTOS','TENSAO (Dinas/cm)',[ESC]
RETORA');
  xplot:=X1;
  X1g:=(getmaxx+2*X2) div 3;
  X2g:=(2*getmaxx+X2) div 3;
  {$I-}
  gota:=0;
  repeat
    inc(gota);

```

```

readln(f,tensoap[gotas]);
if (not eof(f)) then begin
  xplot:=round(X2-((X2-X1)*(100-gotas)/(100-0)));
  yplot:=round(Y1+(TENSAOP[gotas]-80)*(Y2-Y1)/
(0-80));
  bar(xplot-1,Y2,xplot+1,yplot);
  end;
until eof(f) or (gotas=100);
{$i+}
bar(X1g,Y2,X2g,yplot);
outtextxy(X1g+1,Y2+8,strings(tensoap[gotas],5,2));
calcula_media(tensoap,gotas-1,media,desvio,soma);
outtextxy(X1g+1,Y2+16,chr(241)+strings(desvio,4,2));
ch:=leteclado(true);
closegraph;
end;

```

```

procedure analisa_fluxo;
var f : text;
    temp,dens,fluxo,desvio,media,soma : real;
    aux1,aux2,arq : string;
    xplot,yplot,X1g,X2g : word;
    tensoap:array [1..500] of real;
begin
  prepara_tela(1,10,1,1,"");
  arq:=get_dir_name;
  abre_arquivo(f,arq,'TSF',NOME DO ARQUIVO : ',R',ch);
  if (ch=#27) then exit;
  readln(f,aux1);
  readln(f,aux2);
  readln(f,temp);
  telagrfgahi(0,0,maxeixox,80,5,8,10,5,TENSÍOMETR
O',aux1,'No. PONTOS',TENSAO (Dinas/cm)',[ESC] RETORA');
  xplot:=X1;
  X1g:=(getmaxx+2*X2) div 3;
  X2g:=(2*getmaxx+X2) div 3;
  {$i-}
  gotas:=0;
  repeat
    inc(gotas);
    readln(f,tensoap[gotas]);
    if (not eof(f)) then begin
      xplot:=round(X2-(((X2-X1)/(maxeixox-0))*(maxeixox-
gotas)));
      yplot:=round(Y1+(TENSAOP[gotas]-80)*(Y2-Y1)/
(0-80));
      line(xplot,Y2,xplot,yplot);
      end;
    until eof(f);
    {$i+}
    bar(X1g,Y2,X2g,yplot);
    outtextxy(X1g+1,Y2+8,strings(tensoap[gotas],5,2));
    calcula_media(tensoap,gotas-1,media,desvio,soma);
    outtextxy(X1g+1,Y2+16,chr(241)+strings(desvio,4,2));
    ch:=leteclado(true);
  closegraph;
end;

```

```

procedure menu_cadastro;
var volume:real;
begin
  repeat
    tela3;
    ch:=leteclado(true);
    case ch of
      F1 : begin
        prepara_tela(1,3,1,3,"");
        move_bomba;
        if (ch=#13) then adquiere_passo('CAL-
SER.TCS',oldct,gotas,volume);
        if (ch=#13) then begin
          if (gotas>0) and (oldct<>0) then
            calcula_fluxo(volume/oldct)
          else erro('NENHUMA GOTA
DETECTADA.',ch,false);
        end
      end;

```

```

    else erro('PROCEDIMENTO ABORTADO.',ch,false);
    ch:=#0;
  end;
F2 : begin
  prepara_tela(1,3,1,4,['DEL] APAGA');
  expoe_seringa(false);
  seta_seringa;
  end;
F3 : begin
  prepara_tela(1,3,1,4,"");
  expoe_seringa(true);
  end;
end;
until (ch=#27);
if gravar then grava_ser;
end;

```

```

procedure menu_ensaio;
var option:char;
begin
  repeat
    tela_ensaio;
    option:=leteclado(true);
    case option of
      F1 : ensaio_concentracao_exponencial;
      F2 : ensaio_concentracao_linear_simulada;
      F3 : ensaio_fluxo;
      F4 : ensaio_simples;
      F5 : ensaio_variavel_duas_bombas;
    end;
  until (option=#27);
end;

```

```

procedure menu_analise;
var option:char;
begin
  repeat
    tela_analise;
    option:=leteclado(true);
    case option of
      F1 : analise_concentracao;
      F2 : analise_fluxo;
      F3 : analise_simples;
    end;
  until (option=#27);
end;

```

```

procedure menu;
var option:char;
begin
  repeat
    tela2;
    inicia_varia;
    option:=leteclado(true);
    case option of
      F1 : menu_ensaio;
      F2 : menu_cadastro;
      F3 : cadastra_raio;
      F4 : menu_analise;
      F5 : retornar_bomba;
      F6 : passo_da_bomba;
      F7 : confere_detector;
      F8 : parametros;
    end;
  until (option=#27);
end;

```

```

begin
  tela2;
  le_tens_dados;
  le_seringa;
  repeat
    menu;
    erro('TERMINAR PROGRAMA ? [S/N]',ch,true);
  until (ch='S');
end.

```

APÊNDICE III

PROGRAMA PARA CÁLCULO DE BETA

```
program make_beta;
uses utitexto,crt,unregres,utiarq;
var cmca,cmcb,cmcexp,cmcideal,fracao1,
    fracao2,media,desvio,soma:real;
    G1,G2,X1,X2,XM1,XM2,F1,F2,F,ln10:real;
    beta:array [1..255] of real;
    i,qtdd:word;
    ch:char;
    arq:text;
    name:string;
    okayvalor:boolean;

function get_valores(msg:string):real;
var aux:string;
    code:word;
    cmcaux:real;
begin
    repeat
        aux:= ' ';
        texto_crt(5,10,7,msg,aux,ch,1,7);
        val(aux,cmcaux,code);
        until ((code=0) and (cmcaux>0)) or (ch=#27);
        if (ch=#27) then okayvalor:=false else okayvalor:=true;
        get_valores:=cmcaux;
    end;

procedure calcula_valores;
begin
    repeat
        inc(qtdd);
        fracao1:=get_valores('Fracao molar de A : ');
        if (not okayvalor) then exit;
        cmcexp:=get_valores('CMCexp da mistura : ');
        if (not okayvalor) then exit;
        fracao2:= 1- fracao1;
        cmcideal:=fracao1*cmcb + fracao2*cmca;
        cmcideal:=cmcideal/(cmca*cmcb);
        cmcideal:=1/cmcideal;
        G1:= CMCexp * fracao1/CMCa;
        G2:= CMCexp * fracao2/CMCb;
        X1:=0;
        X2:=1;
        i:=0;
        repeat
            inc(i);
            XM1:= 0.5 * (X1 + X2);
            XM2:= 1 - XM1;
            if (XM1<>0) THEN begin
                F1:= Ln(G1/XM1);
                F1:=F1*XM1*XM1/ln10;
            end;
            if (XM2<>0) THEN begin
                F2:= Ln(G2/XM2);
                F2:=F2*XM2*XM2/ln10;
            end;
            F:=F1- F2;
            IF (F > 0) THEN X1:= XM1 ELSE IF (F < 0)
            THEN X2:= XM1;
            until ((F<1E-10) and (F>-1E-10)) or (XM1 = 0) or
            (XM2 = 0);
            if (XM1<>0) and (XM2<>0) then begin
                beta[qtdd]:=ln(G1/XM1);
                beta[qtdd]:=beta[qtdd] / (XM2 * XM2);
            end else beta[qtdd]:=0;
            calcula_media(beta,qtdd,media,desvio,soma);
            gotoxy(10,15);write(' BETA = ',beta[qtdd]:10:4);
            gotoxy(10,18);write('BETA MEDIO = ',media:10:4);
            gotoxy(10,19);write('D. PADRAO = ',desvio:10:4);
            gotoxy(10,20);write('N. AMOSTRAS= ',qtdd:4);
```

```
        gotoxy(40,1+qtdd);write(fracao1:5:2,CMCexp:10:6,CMC
        ideal:10:6,beta[qtdd]:10:4);
        append(arq);

        writeln(arq,fracao1:5:2,CMCexp:10:6,CMCideal:10:6,be
        ta[qtdd]:10:4);
        close(arq);
        until (ch=#27);
    end;

begin
    ch:=#0;
    clrscr;
    name:= '';
    abre_arquivo(arq,name,'bet','NOME DO ARQ. DADOS
    ','?',ch);
    if (ch=#27) then halt;
    clrscr;
    cmca:=get_valores('CMCa puro (100%) : ');
    if (not okayvalor) then halt;
    cmcb:=get_valores('CMCb puro ( 0%) : ');
    if (not okayvalor) then halt;
    qtdd:=0;
    gotoxy(40,1);write('FRACAO CMCexp. CMCideal
    BETA');
    writeln(arq,' 0.00',cmcb:10:6,' - ' - ');
    close(arq);
    ln10:=ln(10);
    calcula_valores;
    append(arq);
    writeln(arq,' 1.00',cmca:10:6,
    ');
    close(arq);
end.
```

APÊNDICE IV

BIBLIOTECA DE ROTINAS

```
unit utigrtns;
interface
uses crt,dos,ungrafic,graph,unmtm,unstring,uncrt;
var cdisp:real;
procedure
setpntlognrmtns(vx1,vy1,vx2,vy2,temper,Nioniz:real);
implementation

procedure
setpntlognrmtns(vx1,vy1,vx2,vy2,temper,Nioniz:real);
var f_line,automatico,escreve:boolean;
    xx,yy,leitura:word;
    rt,vel,icx,icy:byte;
    l:integer;
    xh,ylh,xle,yle: array [1..5] of integer;
    tdisp:real;
    stt,ste:string[20];

procedure intercepto;
var ca1,ca2,cl1,cl2:real;
    xint,yint:integer;
    excesso,area,dgama,conc1,conc2,dlnc,livre:real;
    nr1,nr2:byte;

procedure escreve_interc;
var p : pointer;
    size,locx,locy : word;
begin
locx:=getmaxx div 4;locy:=Y2-81;
repeat
ch:=#0;
Size := ImageSize(locx,locy,locx+320,locy+80);
GetMem(P, Size);
GetImage(locx,locy,locx+320,locy+80,P^);
bar(locx,locy,locx+320,locy+80);
rectangle(locx,locy,locx+320,locy+80);
Cdisp:=vx2-((x2-xint)*(vx2-vx1)/(x2-x1));
Tdisp:=vy2+((yint-y1)*(vy1-vy2)/(y2-y1));
Cdisp:=expbase(Cdisp,10);
dgama:=-(yle[nr1]-ylh[nr1])*80/(y2-y1);
conc1:=expbase(((vx2-((x2-xlh[nr1])*(vx2-vx1)/(x2-
x1))))),10);
conc2:=expbase(((vx2-((x2-xle[nr1])*(vx2-vx1)/(x2-
x1))))),10);
if ((conc2<>0) and (conc1<>0)) then
dlnc:=ln(conc2/conc1)
else dlnc:=0;
excesso:=dgama*1000/(dlnc*8.314*temper*nloniz);
if (excesso<>0) then
area:=1E16*1E10/(excesso*6.022E23)
else area:=0;
if (cdisp<>0) then
livre:=8.314*temper*ln(cdisp*18/1000)
else livre:=0;
msggrf(locx+4,locy+0*16+4,7,'TNSAO =
'+strings(tdisp,7,2)+' Dinac/cm. ');
msggrf(locx+4,locy+1*16+4,7,'CMC =
'+strings(cdisp*1E3,7,2)+' mMolar. ');
msggrf(locx+4,locy+2*16+4,7,'EXC.(x10E10) =
'+strings(excesso,7,2)+' Mol/cm*cm. ');
msggrf(locx+4,locy+3*16+4,7,'AREA SUP. =
'+strings(area,7,2)+' A*A/Molec. ');
msggrf(locx+4,locy+4*16+4,7,'DELTA G =
'+strings(livre/1000,7,2)+' KJ/Mol. ');
repeat
if keypressed then ch:=upcase(readkey);
until (ch in [#72,#80,#75,#77,#27]);
PutImage(locx,locy, P^, NormalPut);
freemem(P,Size);
case ch of
#72 : if (locy>=y1+8) then dec(locy,8) else
dec(locy,locy-y1);
#80 : if (locy+80<=y2-8) then inc(locy,8) else
inc(locy,y2-locy-80);
#75 : if (locx>=x1+8) then dec(locx,8) else
dec(locx,locx-x1);
#77 : if (locx+320<=x2-8) then inc(locx,8) else
inc(locx,x2-locx-320);
end;
until (ch=#27);
end;

begin
repeat
ch:=#0;
msggrf(getmaxx*2 div 5,getmaxy-10,0,'ESCOLHA RETA
1. [F1 .. F5]');
repeat
if keypressed then ch:=upcase(readkey);
until (ch in [#59..#63]);
nr1:=ord(ch)-58;
ch:=#0;
msggrf(getmaxx*2 div 5,getmaxy-10,0,'ESCOLHA
RETA 2. [F1 .. F5]');
repeat
if keypressed then ch:=upcase(readkey);
until (ch in [#59..#63]);
nr2:=ord(ch)-58;
if (nr1=nr2) then msggrf(getmaxx*2 div 5,getmaxy-
10,0,'ESCOLHA RETAS DIFER. ');
until (nr1<nr2);
msggrf(getmaxx*2 div 5,getmaxy-10,0,
');
ch:=#0;
ca1:=(yle[nr1]-ylh[nr1])/(xle[nr1]-xh[nr1]);
ca2:=(yle[nr2]-ylh[nr2])/(xle[nr2]-xh[nr2]);
cl1:=yle[nr1]-ca1*xle[nr1];
cl2:=yle[nr2]-ca2*xle[nr2];
if (ca1-ca2 <>0) then begin
xint:=round((cl2-cl1)/(ca1-ca2));
yint:=round((cl1*ca2 - cl2*ca1)/(ca2-ca1));
escreve_interc;
end
else begin
msggrf(getmaxx*2 div 5,getmaxy-10,0,'IGUAL COEF.
ANGULAR. ');
delay(1000);
msggrf(getmaxx*2 div 5,getmaxy-10,0,
');
end;
ch:=#0;
end;

procedure trata_autom;
begin
automatico:= not automatico;
if (not automatico) then setcolor(0);
outtextxy(472,469,'AUTOM. ');
setcolor(15);
ch:=#0;
end;
procedure esc_lin;
begin
line(xh[RT],ylh[RT],xle[RT],yle[RT]);
DELAY(2);
end;

procedure estica;
begin if (xh[RT]-vel>X1+1) and (xle[RT]+vel<X2-1) then
begin
```

```

esc_lin;dec(xlh[RT],vel);inc(xle[RT],vel);esc_lin;end;end;

procedure encolhe;
begin if (xh[RT]+vel<xle[RT]-vel) then begin
    esc_lin;inc(xh[RT],vel);dec(xle[RT],vel);esc_lin;end;end;

procedure inc_ang;
begin if (yle[RT]-vel>Y1+1) and (yjh[RT]+vel<Y2-1) then begin
    esc_lin;dec(yle[RT],vel);inc(yjh[RT],vel);esc_lin;end;end;

procedure dec_ang;
begin if (yjh[RT]-vel>Y1+1) and (yle[RT]+vel<Y2-1) then begin
    esc_lin;inc(yle[RT],vel);dec(yjh[RT],vel);esc_lin;end;end;

procedure incx;
begin esc_lin;
    if (xle[RT]+vel<X2-1) then begin
        inc(xle[RT],vel);inc(xlh[RT],vel);end
    else begin inc(xle[RT],x2-xle[RT]);inc(xlh[RT],x2-
        xle[RT]);end;esc_lin;end;

procedure decx;
begin esc_lin;
    if (xh[RT]-vel>X1+1) then begin
        dec(xle[RT],vel);dec(xlh[RT],vel);end
    else begin dec(xle[RT],xh[RT]-X1);dec(xlh[RT],xh[RT]-
        X1);end;esc_lin;end;

procedure decy;
begin if (yjh[RT]-vel>Y1+1) and (yle[RT]-vel>Y1+1) then begin
    esc_lin;dec(yle[RT],vel);dec(yjh[RT],vel);esc_lin;end;end;

procedure incy;
begin if (yjh[RT]+vel<Y2-1) and (yle[RT]+vel<Y2-1) then begin
    esc_lin;inc(yle[RT],vel);inc(yjh[RT],vel);esc_lin;end;end;

procedure escreve_cursor;
begin
    line(xx,yy-6,xx,yy-3);
    line(xx,yy+6,xx,yy+3);
    line(xx-6,yy,xx-3,yy);
    line(xx+6,yy,xx+3,yy);
    rectangle(xx-6,yy-6,xx+6,yy+6);
end;

procedure trata_plote;
begin
    Cdisp:=vx2-((x2-xx)*(vx2-vx1)/(x2-x1));
    Tdisp:=vy2+((yy-y1)*(vy1-vy2)/(y2-y1));
    Cdisp:=exp(Cdisp*ln(10));
    msggrf(300,469,15,strings(cdisp,icx+1,icx)+' '+
    strings(tdisp,icy+1,icy));
end;

begin
    settxtstyle(0,0,0);
    getcoordtela;
    automatico:=true;
    RT:=1;
    f_line:=false;
    xc=(x1+x2) div 3;
    for i:=1 to 5 do begin
        yle[i]:=Y2-1;yjh[i]:=Y2-1;
        xle[i]:=xc*2;xh[i]:=xc;
    end;

```

```

    xc=(x1+x2) div 2;
    yy:=(y1+y2) div 2;
    ch:=#0;
    vel:=1;
    icx:=2+get_dec(vx1,vx2);
    icy:=2+get_dec(vy1,vy2);
    icx:=2+trunc(abs(vx1));
    setwritemode(xorput);
    setcolor(15);
    outtextxy(472,469,'AUTOM. ');
    setfillstyle(0,0);
    escreve:=true;
    repeat
        if (not automatico) then ch:=#0;
        if escreve then begin
            escreve_cursor;
            trata_plote;
        end;
        repeat
            if keypressed then ch:=(readkey);
            until (ch<>#0);
            if (ch in [#72,#80,#75,#77]) then escreve:=true else
                escreve:=false;
            if escreve then escreve_cursor;
            case ch of
                #72 : if (yy>(y1+vel)) then dec(yy,vel) else
                #80 : if (yy<(y2-vel)) then inc(yy,vel) else
                #75 : if (xx>(x1+vel)) then dec(xx,vel) else
                #77 : if (xx<(x2-vel)) then inc(xx,vel) else
                #42 : inc_ang;
                #47 : dec_ang;
                #52 : decx;
                #54 : incx;
                #56 : decy;
                #50 : incy;
                #82 : estica;
                #83 : encolhe;
                #09 : intercepto;
                #59..#63 : RT :=ord(ch)-58;
                '+' : if (vel<15) then inc(vel);
                '-' : if (vel>1) then dec(vel);
                #32 : trata_autom;
            end;
            until (ch in [#27,#13,#64..#68]);
            trata_plote;
        end;
    end.
end.

```

UNIT UNGRAFIC;

```

INTERFACE
var x1,y1,x2,y2:word;
Procedure inicializacao;
function get_dec(v1,v2:real):integer;
procedure msggrf(posx,posy,back:integer;msg:string);
procedure msggrfint(posx,posy,back:integer;valor:longint);
procedure
msggrfreal(posx,posy,back:integer;valor:real;tam,dec:byte);
procedure getcoordtela;
procedure msg_tela(X1,Y1,cor:word;separ:char;msg:string);
procedure monta_tela_grf(titulo,frase1,msg:string);
procedure
monta_abc_nrm(XX1,XX2:word;vx1,vx2:real;divx,divy:byte;ab
c:string);
procedure
monta_ord_nrm(YY1,YY2:word;vy1,vy2:real;divy,divx:byte;ord:
string);

```

```
procedure
telagrfgahi(xini,yini,xfin,yfin:real;divx,divy,divix,diviy:byte;
titulo,frase1,titx,tity,msg:string);
```

IMPLEMENTATION

uses graph;

Procedure inicializacao;

```
var graphdriver,graphmode,errorcode : integer;
begin
  detectgraph(graphdriver,graphmode);
  initgraph(graphdriver,graphmode,"");
  Errorcode :=graphresult;
  if (errorcode = grOk) then getcoordtela
  else begin
    WriteLn('Graphics error: ',GraphErrorMsg(ErrorCode));
    halt;
  end;
end;
```

function get_dec(v1,v2:real):integer;

```
var vmaior,vmenor:real;
    cascas:byte;
begin
  if (v1>v2) then begin
    vmaior:=v1;
    vmenor:=v2;
  end
  else begin
    vmenor:=v1;
    vmaior:=v2;
  end;
  if ((vmaior-vmenor) >=100) then cascas:=0
  else if ((vmaior-vmenor) >=10) then cascas:=1
  else if ((vmaior-vmenor) >=1) then cascas:=2
  else if ((vmaior-vmenor) >=0.1) then cascas:=3
  else if ((vmaior-vmenor) >=1E-2) then cascas:=4
  else if ((vmaior-vmenor) >=1E-3) then cascas:=5
  else if ((vmaior-vmenor) >=1E-4) then cascas:=6
  else cascas:=7;
  get_dec:=cascas;
end;
```

procedure msggrf(posx,posy,back:integer;msg:string);

```
var old:fillsettingstype;
begin
  getfillsettings(old);
  setfillstyle(old.pattern,back);
  bar(posx-1,posy,posx+1+length(msg)*8,posy+8);
  setfillstyle(old.pattern,old.color);
  outtextxy(posx,posy,msg);
end;
```

procedure msggrfint;

```
var aux:string;
begin
  str(valor,aux);
  msggrf(posx,posy,back,aux);
end;
```

procedure msggrfreal;

```
var aux:string;
begin
  str(valor:tam:dec,aux);
  msggrf(posx,posy,back,aux);
end;
```

```
{
HercMonoHi ³ 0 ³ 720 x 348 ATT400Hi ³ 5 ³ 640 x 400
IBM8514Hi ³ 1 ³ 1024 x
PC3270Hi ³ 0 ³ 720 x 350
}
```

```
procedure getcoordtela;
var graphdriver,graphmode:integer;
begin
```

```
  detectgraph(graphdriver,graphmode);
  case graphmode of
    vgahi,
    mcgahi,
    ibm8514lo : begin
      x1:=64;y1:=38;x2:=564;y2:=438;
    end;
    cgac0,cgac1,cgac2,cgac3,
    mcgac0,mcgac1,mcgac2,
    mcgac3,att400c0,att400c1,
    att400c2,att400c3 : {320 x 200} begin end;
    cgahi,mcgamed,vgalo,egal0,
    ega64lo,att400med : {640 x 200} begin end;
    egahi,ega64hi,egamonohi,
    vgamed : {640 x 350} begin end;
  end;
end;
```

procedure msg_tela(X1,Y1,cor:word;separ:char;msg:string);

```
var i,posic:byte;
    aux:string;
begin
  settxtstyle(0,0,0);
  i:=0;
  repeat
    inc(i);
    posic:=pos(separ,msg);
    if (posic<>0) then aux:=copy(msg,1,posic-1)
    else aux:=msg;
    delete(msg,1,posic);
    msggrf(X1,Y1+i*8,cor,aux);
  until (posic=0);
end;
```

procedure monta_tela_grf(titulo,frase1,msg:string);

```
begin
  rectangle(0,0,getmaxx,getmaxy);
  line(0,19,getmaxx,19);
  line(0,getmaxy-14,getmaxx,getmaxy-14);
  line(48,getmaxy,48,getmaxy-14);
  rectangle(x1,y1,x2,y2);
  settxtjustify(1,2);
  settxtstyle(0,0,2);
  outtextxy((x1+x2) div 2,3,titulo);
  outtextxy((x1+x2) div 2,22,frase1);
  settxtstyle(0,0,0);
  settxtjustify(0,1);
  outtextxy(12,getmaxy-6,'MSG: '+msg);
end;
```

procedure

monta_abc_nrm(XX1,XX2:word;vx1,vx2:real;divx,divic:byte;abc:string);

```
var a,i,c:integer;
    ee,ex,incre:real;
    st:string[10];
begin
  settxtjustify(0,2);
  settxtstyle(0,0,0);
  ic:=get_dec(vx1,vx2);
  ee:=vx1;
  a:=XX1;
  ex:=(vx2-vx1)/divx;
  incre:=(xx2-xx1)/divx;
  for i:=0 to divx do begin
    a:=round((i*incre)+xx1);
    line(a,y2,a,y2+4);
    str(ee:ic+1:ic,st);
    outtextxy(a-textwidth(st) div 2,y2+textheight(st),st);
    ee:=ee+ex;
  end;
  incre:=(xx2-xx1)/(divx*divic);
```

```

a:=xx1;
for i:=1 to divx*divy do begin
  a:=round((1*inre)+xx1);
  line(a,y2,a,y2+2);
end;
settextstyle(0,0,0);
outtextxy(x2-textwidth(abc),y2+18,abc);
end;

procedure
monta_ord_nrm(YY1,YY2:word;vy1,vy2:real;divy,diviy:byte;ord:
string);
var a,i,ic:integer;
    ee,ex,inre:real;
    st:string;
begin
  a:=yy2;
  ee:=vy1;
  ex:=(vy2-vy1)/divy;
  ic:=get_dec(vy1,vy2);
  inre:=(yy2-yy1)/divy;
  settextjustify(2,2);
  settextstyle(0,0,0);
  for i:=divy downto 0 do begin
    a:=round((1*inre)+yy1);
    line(x1,a,x1-4,a);
    str(ee:ic+1:ic,st);
    if (length(st)>=(x1 div 8)) then str(ee:(x1 div 8)-1:(x1 div
8)-1-pos(' ',st),st);
    outtextxy(x1-5,a+5-textheight(st),st);
    ee:=ee+ex;
  end;
  inre:=(yy2-yy1)/(divy*diviy);
  a:=yy2;
  for i:=divy*diviy downto 0 do begin
    a:=round((1*inre)+yy1);
    line(x1,a,x1-2,a);
  end;
  settextstyle(0,1,0);
  settextjustify(0,2);
  outtextxy(x1-(ic+4)*8,yy1+textheight(ord),ord);
end;

procedure telagrfgahi;
begin
  inicializacao;
  monta_tela_grf(titulo,frase1,msg);
  monta_ord_nrm(y1,y2,yini,yfin,divy,diviy,tity);
  monta_abc_nrm(x1,x2,xini,xfin,divx,divix,tix);
end;
end.

unit utiser;

interface

uses crt,utitexto,uncrt,untela,utitens,unarquiv,unstring;

type
  Tipo_seringa = record
    nome:string[8];
    ml_cm:real;
  end;

var
  n_seringa:byte;
  ser_atu:integer;
  seringa:array [0..99] of tipo_seringa;
  procedure grava_ser;
  procedure expoe_seringa(pausa:boolean);
  procedure le_seringa;
  procedure calcula_fluxo(volp:real);
  procedure seta_seringa;

implementation

```

```

var l:byte;

Procedure
escreve_seringa(jx,jy,nnum:word;coloracao:boolean);
begin
  if (coloracao) then cor(1,7);
  gotoxy(jx,jy);write(branco(22));
  gotoxy(jx,jy);write(seringa[nnum].nome:8,'=
',seringa[nnum].ml_cm:4:2,' ml/cm');
  cor(7,1);
end;

Procedure Grava_ser;
var f:file;
    errowrite:boolean;
    arq:string;
begin
  repeat
    errowrite:=false;
    arq:=get_dir_name+'TENSERIG.CAD';
    {$i-}
    assign(f,arq);
    rewrite(f,1);
    blockwrite(f,n_seringa,sizeof(n_seringa));
    if (ioresult<>0) then errowrite:=true;
    if (n_seringa>0) then for i:=1 to n_seringa do begin
      blockwrite(f,seringa[i],sizeof(seringa[i]));
      if (ioresult<>0) then errowrite:=true;
    end;
    close(f);
    if (ioresult<>0) then errowrite:=true;
    {$i+}
    if (errowrite) then erro('ERRO DE GRAVACAO.
TENTAR NOVAMENTE ? [S/N] ',ch,true);
    until ((not errowrite) or (ch='N'));
    gravar:=false;
  end;

Procedure Le_seringa;
var f:file;
    arq:string;
procedure msg_erro(mensagem:string);
begin
  erro(arq+mensagem+' . CRIAR, REPETIR, ABORTAR ?
[C/R/A] ',ch,true);
end;

begin
  arq:=get_dir_name+'TENSERIG.CAD';
  gotoxy(10,24);write('LENDO '+arq);
  repeat
    ch:='s';
    if (existe_arquivo(arq)) then begin
      {$i-}
      assign(f,arq);
      reset(f,1);
      blockread(f,n_seringa,sizeof(n_seringa));
      if (n_seringa>0) then for i:=1 to n_seringa do
        blockread(f,seringa[i],sizeof(seringa[i]));
      close(f);
      {$i+}
    end
    else msg_erro(' NAO ENCONTRADO');
    if (ioresult <> 0) then msg_erro(' COM ERRO DE I/O');
    until (ch in ['C','A','s']);
    case ch of
      'C' : begin
        n_seringa:=0;
        grava_ser;
      end;
      'A' : begin
        erro('PROGRAMA ABORTADO.',ch,false);
        halt;
      end;
    end;
  end;
end;

```



```

        end;
    end;
    ser_atu:=n_seringa;
    if (n_seringa=0) then begin
        seringa[0].nome:='VAZIO';
        seringa[0].ml_cm:=0.0;
    end;
end;

procedure expoe_seringa(pausa:boolean);
var xx,yy:byte;
    ch:char;
begin
    if (n_seringa>0) then begin
        ch:=#0;
        xx:=3;
        yy:=0;
        for i:=1 to n_seringa do begin
            if (5+i-yy=23) then begin
                inc(yy,17);
                inc(xx,25);
            end;
            gotoxy(xx,5+i-yy);
            cor(7,1);
            escreve_seringa(xx,5+i-yy,i,false);
        end;
        if (pausa) then
            repeat
                if keypressed then ch:=upcase(readkey);
            until (ch<>#0);
        end
        else erro('CADASTRO VAZIO.',ch,false);
    end;
end;

procedure apaga_seringa;
begin
    prepara_tela(1,3,1,1,"");
    for i:=ser_atu to n_seringa-1 do seringa[i]:=seringa[i+1];
    dec(n_seringa);
    expoe_seringa(false);
    gravar:=true;
end;

procedure seta_seringa;
var xx,yy,coluna :byte;
    ch:char;
begin
    if (n_seringa <=0) then exit;
    xx:=3;
    yy:=0;
    coluna:=0;
    repeat
        ch:=#0;
        if (ser_atu>n_seringa) then ser_atu:=n_seringa;
        if (ser_atu<1) then ser_atu:=1;
        coluna:=ser_atu div 17;
        if ((ser_atu div 17) = (ser_atu/17)) then dec(coluna);
        escreve_seringa(coluna*25+3,ser_atu+5-
coluna*17,ser_atu,true);
        repeat
            if keypressed then ch:=upcase(readkey);
        until (ch in [DEL,up,dn,LE,RI,#27]);
        if (ch in [UP,DN,LE,RI]) then
            escreve_seringa(coluna*25+3,ser_atu+5-
coluna*17,ser_atu,false);
        case ch of
            DEL : apaga_seringa;
            UP : dec(ser_atu);
            Dn : inc(ser_atu);
            LE : dec(ser_atu,17);
            RI : inc(ser_atu,17);
        end;
    until (ch=#27);
end;
end;

```

```

procedure calcula_fluxo(volp:real);
var    fluxo : real;
        nome  : string;
begin
    nome:='SERINGA1';
    Prepara_tela(1,3,1,1,"");
    if (volp >0) then begin
        fluxo:=volp*par[6];
        repeat
            erro('FLUXO DA SERINGA = '+strings(fluxo,5,2)+'
ml/cm. GRAVAR ? [S/N]',ch,true);
            until (ch in [#13,#27,'S','N']);
            if (ch in [#13,'S']) then begin
                repeat
                    texto_crt(10,10,8,'DIGITE O NOME DA SERINGA:
',nome,ch,1,7);
                    until (ch=#27) or ((ch=#13) and (nome<>""));
                    if (ch=#27) then exit;
                    inc(n_seringa);
                    seringa[n_seringa].nome:=nome;
                    seringa[n_seringa].ml_cm:=fluxo;
                    grava_ser;
                end;
            end;
        end;
    end;
end;

unit utitens;

interface

uses dos,untela,utitexto,unarquiv,crt,uncrt,utiarq,
    unirqmqj,unrs232,unstring,unregres,untecl,utisttc;

const
    bobina : array [0..3] of byte = ($11,$12,$14,$18);
    velmax=500;
    qtddpar=16;
    maxvalor=32767;

var
    countera,counterb,ct1,ct2,numct,oldct,deltact:integer;
    ctaa,ctab:longint;
    oldadostr:string;
    par : array [1..qtddpar] of real;
    fim1,fim2,fim1t,fim2t,fim1f,fim2f,fimall,fim,anda:byte;
    gota,porta,aguarde:word;
    bombaok,gravar,plotar,bomba1,bomba2,pingou,falhou,f_c
hkplaca:boolean;
    function sensor_gota_ok:boolean;
    procedure confere_detector;
    procedure falha;
    procedure inicia_varia;
    procedure falhando;
    procedure programa_porta;
    procedure porta_bobina;
    procedure grava_tens_dados;
    procedure le_tens_dados;
    procedure passo_da_bomba;
    procedure retomar_bomba;
    procedure calcula_raio(volg:real);
    procedure calculo_tensao_conc(vpb:real;var
tensp:real;var concp:real);
    procedure move_bomba;
    function trans_dado_massa(var aux:string):real;
    procedure adquire_passo(nome_arq:string;var
oldct:integer;
                                var gota:word;var
volume:real);
    procedure parametros;
    procedure troca_pto(var arq:string;var ch:char;qtdd:word);
    function liga_balanca:boolean;
    function get_dir_name:string;

```

```

implementation
function sensor_gota_ok:boolean;
var count,i:word;
begin
  count:=0;
  for i:=1 to 500 do begin
    porta_bobina;
    if ((fim and 1) = 0) then inc(count);
  end;
  sensor_gota_ok:=boolean(count and i);
end;

procedure confere_detector;
var oncount,offcount:word;
  detecresult:real;
procedure next_procedure;
begin
  gotoxy(10,20);write('TECLE [ENTER] PARA PROXIMO
  TESTE. ');
end;
procedure mensagem_3segundos;
begin
  gotoxy(10,14);write('AGUARDE 3 SEGUNDOS. ');
end;
procedure zera_contadores;
begin
  oncount:=0;
  offcount:=0;
end;
procedure incrementa_leitura(qtdd:word);
var i:word;
begin
  for i:=1 to qtdd do begin
    porta_bobina;
    if ((fim and 1) = 0) then inc (oncount) else inc(offcount);
  end;
end;
begin
  prepara_tela(1,23,1,4,");
  gotoxy(10,12);write('MANTENHA O DETECTOR LIVRE
  E TECLE [ENTER]');
  if (leteclado(true)<>#13) then exit;
  mensagem_3segundos;
  zera_contadores;
  incrementa_leitura(3000);
  detecresult:=oncount;
  detecresult:=(detecresult*100)/3000;
  gotoxy(10,16);write('RESULTADO : DETECTOR LIVRE
  EM ',detecresult:5:2,' % ');
  gotoxy(10,18);write('PROBLEMA EM ',offcount,'
  LEITURAS. (TOTAL = 3000)');
  next_procedure;
  if (leteclado(true)<>#13) then exit;
  prepara_tela(1,23,1,4,");
  gotoxy(10,12);write('OBSTRUA O DETECTOR E TECLE
  [ENTER]. ');
  if (leteclado(true)<>#13) then exit;
  mensagem_3segundos;
  zera_contadores;
  incrementa_leitura(3000);
  detecresult:=offcount;
  detecresult:=(detecresult*100)/3000;
  gotoxy(10,16);write('RESULTADO : DETECTOR
  OBSTRUÍDO EM ',detecresult:5:2,' % ');
  gotoxy(10,18);write('PROBLEMA EM ',oncount,'
  LEITURAS. (TOTAL = 3000) ');
  next_procedure;
  if (leteclado(true)<>#13) then exit;
  prepara_tela(1,23,1,4,");
  zera_contadores;
  repeat
    incrementa_leitura(1);
    gotoxy(10,10);write('CLARO = ',oncount,' ESCURO =
    ',offcount);
    until (leteclado(off)=#27);
  end;

  procedure falha;
  var i:word;
  begin
    falhou:=true;
    for i:=1 to 50 do begin
      porta_bobina;
      if (fim and fimali = fimali) then falhou:=false;
    end;
  end;

  procedure inicia_varia;
  begin
    ch:=#0;
    gravar:=false;
    ctaa:=round(taxa/par[8]);
    bomba1:=true;fim1:=$C0;fim1t:=$40;fim1f:=$80;porta:=$3
    00;
    bomba2:=true;fim2:=$06;fim2t:=$02;fim2f:=$04;porta:=$3
    01;
    fimail:=fim1 or fim2;
  end;

  procedure programa_porta;
  begin
    port[$303]:=$89;port[$300]:=$10;port[$301]:=$10;fim:=por
    t[$302];
  end;

  procedure porta_bobina;
  begin
    programa_porta;
    delay(1);
    fim:=port[$302];
  end;

  procedure movimento;
  interrupt;
  begin
    inc(countera);
    if (countera>=ctaa) then begin
      countera:=0;

      port[$303]:=$89;port[$300]:=$10;port[$301]:=$10;fim:=por
      t[$302];
      if ((anda and $20) = $20) then begin
        if ((anda and $10)=$10) then dec(ct2) else inc(ct2);
        port[$301]:=bobina[ct2 and 3];
      end;
      if ((anda and $02) = $02) then begin
        if ((anda and $01)=$01) then dec(ct1) else inc(ct1);
        port[$300]:=bobina[ct1 and 3];
      end;
    end;
    port[$20]:=$20;
  end;

  procedure move_bomba;
  const maxtempo=15;
    espera : array [0..maxtempo] of word =
    (1,2,3,5,7,10,15,20,25,50,75,100,200,250,400,500);
  var i,oldcta:word;
    tempo:byte;
  begin
    oldcta:=ctaa;
    tempo:=5;
    countera:=0;
    ctaa:=taxa div espera[tempo];
    gotoxy(25,8);write('POSICIONE A BOMBA E TECLE
    [ENTER]. ');
    gotoxy(10,11);
    write(' [F1] [LE] [UP] [RI] [F2]');

```

```

gotoxy(10,12);
write('ANDA/PARA INVERTE  RETORNA  INVERTE
ANDA/PARA');
gotoxy(10,13);
write(' BOMBA1  BOMBA1  BOMBAS  BOMBA2
BOMBA2');
gotoxy(10,16);
write('[ESPACO] [+] [DN] [-] [ENTER]');
gotoxy(10,17);
write(' PARA AUMENTA AVANCA DIMINUI
[PROSSEGUE]');
gotoxy(10,18);
write(' BOMBAS VELOC. BOMBAS VELOC.
[ROTINA]');
gotoxy(10,21);writeln('VELOC. =
',(taxa/ctaa)*60/par[6]:4:2,' cm/seg. ');
gotoxy(10,22);writeln(' = ',taxa/ctaa:5:2,' passos/seg. ');
if (not f_chkplaca) then begin
  chkck(4);
  f_chkpiaca:=true;
end;
repeat
  falhando;
  if (sensor_gota_ok=false) then begin
    erro('SENSOR OTICO DE GOTAS COM
PROBLEMA.',ch,false);
    ch:=#27;
    exit;
  end;
  anda:=anda and $DD;
  if not bombaok then begin
    ch:=#27;
    exit;
  end;
  ch:=#0;
  inicia_aquisicao(@movimento);
  repeat
    if keypressed then begin
      ch:=upcase(readkey);
      case ch of
        DN : anda:=anda and $EE;
        UP : anda:=anda or $11;
        ri : anda:=anda xor $01;
        le : anda:=anda xor $10;
        space: anda:=anda and $DD;
        F1 : anda:=anda xor $02;
        F2 : anda:=anda xor $20;
        '+' : if (tempo<maxtempo) then inc(tempo);
        '-' : if (tempo>0) then dec(tempo);
      end;
      ctaa:=taxa div espera[tempo and maxtempo];
      gotoxy(10,21);writeln('VELOC. =
',(taxa/ctaa)*60/par[6]:4:2,' cm/seg. ');
      gotoxy(10,22);writeln(' = ',taxa/ctaa:5:2,'
passos/seg. ');
      end;
      until ((ch in [#13,#27]) or (fim and fimall <> fimall));
      termina_aquisicao_bombas;
      until ((ch in [#13,#27]));
      porta_bobina;
      ctaa:=oldcta;
    end;
  end;

procedure falhando;
var oldcta,oldanda,retorno: word;
  ch : char;
const msg : string = ' DESLIGADA. VERIFIQUE E TECLE
[ENTER].';
begin
  bombaok:=false;
  ch:=#0;
  oldcta:=ctaa;
  oldanda:=anda;
  retorno:=round(par[6]/10);
  ctaa:=taxa div velmax;

```

```

porta_bobina;
while (fim and fimall<>fimall) do begin;
  if bomba2 then begin
    repeat
      repeat
        porta_bobina;
        if ((fim and fim2)=0) then begin
          erro('BOMBA 2'+msg,ch,true);
          if (ch =#27) then exit;
        end;
        until ((fim and fim2) <> 0);
        if ((fim and fim2) <> fim2) then begin
          ct2:=0;
          if ((fim and fim2) <> fim2t) then anda:=$20;
          if ((fim and fim2) <> fim2f) then anda:=$30;
          porta:=$301;
          erro('FIM DE CURSO PARA BOMBA 2.
',ch,false);
          inicia_aquisicao(@movimento);
          repeat
            until (ct2>=retorno) or (ct2<=-retorno);
            termina_aquisicao_bombas;
          end;
          porta_bobina;
          until ((fim and fim2) = fim2);
        end;
        if bomba1 then begin
          repeat
            repeat
              porta_bobina;
              if ((fim and fim1)=0) then begin
                erro('BOMBA 1'+msg,ch,true);
                if (ch =#27) then exit;
              end;
              until ((fim and fim1) <> 0);
              if ((fim and fim1) <> fim1) then begin
                ct1:=0;
                if ((fim and fim1) <> fim1t) then anda:=$02;
                if ((fim and fim1) <> fim1f) then anda:=$03;
                porta:=$300;
                erro('FIM DE CURSO PARA BOMBA
1.',ch,false);
                inicia_aquisicao(@movimento);
                repeat
                  until (ct1>=retorno) or (ct1<=-retorno);
                  termina_aquisicao_bombas;
                end;
                porta_bobina;
                until ((fim and fim1) = fim1);
              end;
            end;
            porta_bobina;
            end;
            ctaa:=oldcta;
            bombaok:=true;
            anda:=oldanda;
          end;
        end;

procedure calculo_passo_bomba(passos:longint);
var auxword,code:word;
  espaco:real;
  aux : string;
begin
  auxc="";
  prepara_tela(1,4,1,2,"");
  repeat
    texto_crt(10,14,5,'ESPACO PERCORRIDO (cm) =
',aux,ch,1,7);
    val(aux,espaco,code);
    until ((espaco > 0) and (code=0)) or (ch=#27);
    if (ch=#27) then exit;
    auxword:=round(passos/espaco);
    erro('PASSO DA BOMBA = '+strings(auxword,4,0)+'
PASSOS/CM. GRAVAR ? [S/N]',ch,true);
    if (ch in [#13,'S']) then begin
      par[6]:=auxword;

```

```

    grava_tens_dados;
end;

function arq_tens_dado:string;
begin
    arq_tens_dado:=get_dir_name+'TENSADADO.INI';
end;

Procedure grava_tens_dados;
begin
    grava_dados(arq_tens_dado,@par,sizeof(par));
end;

Procedure le_tens_dados;
const  condini:array[1..qtddpar] of real =
(0.1,0,23.8,1,1,2679,0.3,1,20,298,1,981,1,0,0,8);
var i:byte;
begin
    if (not le_dados(arq_tens_dado,@par,sizeof(par))) then
        for i:=1 to qtddpar do par[i]:=condini[i];
end;

procedure movimenta_passo;
interrupt;
begin
    inc(countera);
    if (countera>=ctaa) then begin
        countera:=0;
        programa_porta;
        port[porta]:=bobina[ct1 and 3];
        inc(ct1);
    end;
    port[$20]:=$20;
end;

procedure passo_da_bomba;
begin
    ch:=#0;
    ct1:=0;
    countera:=0;
    prepara_tela(1,7,1,3,"");
    move_bomba;
    if (ch=#27) then exit;
    prepara_tela(1,7,1,0,['ESC] ABORTA');
    gotoxy(10,10);writeln('DIGITE [ENTER] PARA TERMINAR
CALIBRACAO. ');
    ch:=#0;
    inicia_aquisicao(@movimenta_passo);
    repeat
        if keypressed then ch:=upcase(readkey);
    until ((ch in [#13,#27]) or (fim and fimall <> fimall));
    termina_aquisicao_bombas;
    if (ch<>#27) then calculo_passo_bomba(ct1);
    if ((ch in [#27,'N']) or (fim and fimall <> fimall))
    then erro('PROCEDIMENTO ABORTADO.',ch,false);
end;

procedure retorno;
interrupt;
begin
    inc(countera);
    if (countera>=ctaa) then begin
        countera:=0;
        programa_porta;
        port[porta]:=bobina[ct1 and 3];
        dec(ct1);
    end;
    port[$20]:=$20;
end;

procedure retornar_bomba;
var oldcta:word;
begin
    oldcta:=ctaa;

```

```

    ctaa:=taxa div velmax;
    countera:=0;
    ch:=#0;
    prepara_tela(1,6,1,3,"");
    repeat
        erro('DESEJA RETORNAR BOMBA PARA INICIO DE
CURSO? [S/N]',ch,true);
    until (ch in [#13,#27,'S','N']);
    if (ch in [#13,'S']) then begin
        gotoxy(10,10);writeln('RETORNANDO BOMBA PARA
INICIO DE CURSO. ');
        if (not f_chkplaca) then begin
            chkclk(4);
            f_chkplaca:=true;
        end;
        repeat
            falhando;
            inicia_aquisicao(@retorno);
            repeat
                if keypressed then ch:=upcase(readkey);
            until ((ch=#27) or (fim and fimall <> fimall));
            termina_aquisicao_bombas;
        until (((fim and fimall) + fim1t) xor fim1) =0) or (ch=#27);
        falhando;
    end;
    ctaa:=oldcta;
end;

procedure calcula_raio(volg:real);
var vol13,c0,c1,c2,c3,zero,raio,tensao,sup,inf:real;
    aux:string;
    code:word;
begin
    repeat
        str(97.7 - 0.016335*par[10] - 2.35197E-
4*par[10]*par[10]:6:3,aux);
        ch:=#0;
        prepara_tela(1,4,1,3,"");
        repeat
            texto_crt(10,24,6,'DIGITE TENSÃO SUPERFICIAL DA
SOLUCAO : ',aux,ch,1,7);
            val(aux,tensao,code);
            until ((tensao>0) and (code=0)) or (ch=#27);
            if (ch=#27) then exit;
            vol13:=exp((ln(volg))/3);
            c0:=volg*par[4]*par[12]/(2*pi*tensao);
            c1:=0.9267;
            c2:=0.7694/vol13;
            c3:=0.4512/(vol13*vol13);
            cor(1,7);
            gotoxy(10,24);write('CALCULANDO. AGUARDE... ');
            sup:=3;
            inf:=0;
            repeat
                raio:=(sup+inf)/2;
                if (keypressed) then ch:=upcase(readkey);
                zero:=raio*raio*raio*c3 - raio*raio*c2 + raio*c1 - c0;
                if (zero>0) then sup:=raio else if (zero<0) then inf:=raio;
            until ((ch=#27) or ((zero < 1E-8) and (zero > -1E-8)));
            if (((raio/vol13)<0.65) or ((raio/vol13)>0.95)) then
                erro('EQUACAO NAO APROPRIADA PARA RAO
EFETIVO.',ch,false);
            erro('RAIO EFETIVO = '+strings(raio,6,4)+' cm. GRAVAR
? [S/N]',ch,true);
            if (ch in [#13,'S']) then begin
                erro('GRAVANDO RAO EFETIVO.',ch,false);
                par[7]:=raio;
                Grava_tens_dados;
            end;
            erro('REPETIR CALCULOS ? [S/N]',ch,true);
        until (ch in ['N',#27]);
    end;

procedure calculo_tensao_conc(vpb:real;var tensp:real;var
conc:real);

```

```

var fator,raiz:real;
begin
  raiz:=par[7]/exp((ln(vpb))/3);
  Fator:=0.9267-0.7694*raiz+0.4512*raiz*raiz;
  Tensp:=(vpb*981)/(2*pi*par[7]*Fator);
  if (tensp>87.5) then tensp:=87.5;
end;

procedure calibra_raio;
interrupt;
begin
  inc(countera);
  fim:=port[$302];
  if (countera>=ctaa) then begin
    countera:=0;
    programa_porta;
    port[porta]:=bobina[ct1 and 3];
    inc(ct1);
  end;
  if (((fim and $1)=$1) and (not pingou)) then begin
    oidadostr:=dadostr;
    pingou:=true;
    oldct:=ct1;
  end;
  if ((pingou) and ((fim and $1) = $0)) then begin
    inc(aguarde);
    if (aguarde>=taxa) then begin
      plotar:=true;
      pingou:=false;
    end;
  end;
  port[$20]=$20;
end;

function trans_dado_massa(var aux:string):real;
var massa:real;code:word;
begin
  aux[0]:=chr(length(aux)-2);
  while (pos(' ',aux)>0) do aux[pos(' ',aux)]:='0';
  val(aux,massa,code);
  trans_dado_massa:=massa;
end;

procedure adquire_passo;
var code,oldcta,i,lendo:word;
    vagmedio,desvio,massa,oldmassa:real;
    vag:array[1..100] of real;
    f:text;
    aux:string;
begin
  {-}$}
  assign(f,get_dir_name+nome_arq);
  rewrite(f);
  writeln(f,'CALIBRACAO DE RAO EFETIVO');
  close(f);
  prepara_tela(1,4,1,3,"");
  gotoxy(10,10);write("TECLE [F3] PARA TERMINAR
CALIBRACAO.");
  ch:=#0;
  countera:=0;
  lendo:=0;
  pingou:=false;
  oldcta:=ctaa;
  ctaa:=round(taxa/par[16]);
  volume:=0;
  gota:=0;
  aguarde:=0;
  plotar:=false;
  deltact:=maxvalor;
  ct1:=0;
  oldct:=0;
  if (not liga_balanca) then exit;
  inicia_comunicacao(com2,1200,'n',1,8,'l');
  inicia_aquisicao(@calibra_raio);
  repeat

```

```

  if (pr<>1) then lendo:=0 else inc(lendo);
  case ch of
    #13 : deltact:=ct1-oldct;
    "*" : begin
      deltact:=deltact+(deltact div 10);
      ctaa:=round(taxa/par[16]);
    end;
    '/' : deltact:=deltact-(deltact div 10);
    #32 : ctaa:=oldcta;
  end;
  ch:=#0;
  if keypressed then ch:=upcase(readkey);
  if (fim and fimall <> fimall) then falha;
  gotoxy(10,22);write(ct1-oldct:6,deltact:6,taxa/ctaa:6:2);
  if (ct1-oldct) > deltact then ctaa:=oldcta;
  if (plotar) then begin
    aguarde:=0;
    plotar:=false;
    inc(gota);
    repeat
      until (dadostr[length(dadostr)='G']);
      aux:=dadostr;
      massa:=trans_dado_massa(aux);
      oldmassa:=trans_dado_massa(oldadostr);
      vag[gota]:=(massa-oldmassa)/par[4];
      if (vag[gota]>0) then begin
        volume:=volume+vag[gota];
        vagmedio:=volume/gota;
        desvio:=0;
        for i:=1 to gota do desvio:=desvio+sqr(vagmedio-
vag[i]);
        if (gota>1) then desvio:=sqrt(desvio/(gota-1));
        gotoxy(10,20);write('GOTA = ',gota,' VOLUME =
',vag[gota]:7:4,
        VOL. MEDIO = ',vagmedio:7:4,
        ',chr(241),desvio:7:4);
        append(f);
        writeln(f,gota:3,vag[gota]:7:4,volume:8:4);
        close(f);
        ctaa:=round(taxa/par[16]);
      end
    else dec(gota);
  end;
  until ((ch in [esc,F3]) or (gota=par[9]) or (fim and fimall <>
fimall)
  or (lendo>3000));
  termina_aquisicao_bombas;
  termina_comunicacao;
  if (gota>0) then begin
    teste_discordancia(vag,gota,1);
    rewrite(f);
    volume:=0;
    for i:=1 to gota do begin
      volume:=volume+vag[i];
      writeln(f,i:3,vag[i]:7:4,volume:8:4);
    end;
    writeln(f,i+1:3,vagmedio:7:4,desvio:8:4);
    close(f);
  end
  else erase(f);
  {+$}
  ctaa:=oldcta;
  if ((ch <> #27) and (lendo<=3000)) then ch:=#13;
  if (fim and fimall <> fimall) then falhando;
  if (lendo>3000) then erro('FALHA COMUNICAO
BALANCA.',ch,false);
end;

procedure parametros;
var i,code:word;
    aux:string;
    partmp:array [1..qtddpar] of real;
    fora:boolean;
const
  inic=6;

```

```

tam:byte=8;
decimo:array [1..qtddpar] of
byte=(4,4,3,4,4,1,4,2,0,1,0,1,5,5,5,2);
condmax:array [1..qtddpar] of real=(20,20,100,5,5,
,30000,2,500,500,500,30,5000,5,0,05,0,01,500);
condmin:array [1..qtddpar] of real=(0,0,0,1,0,1,0,1,0,1,30
,0,1,1e-2,1,100,1,100,0,1,-0,05,-0,01,1e-2);

procedure gravacao_dados;
var i:byte;
begin
for i:=1 to qtddpar do par[i]:=partmp[i];
if (ch=#S1) then grava_tens_dados;
ctaa:=round(taxa/par[8]);
end;

procedure make_tela;
var j:byte;
begin
tela4;
for j:=1 to qtddpar do begin
partmp[j]:=par[j];
gotoxy(56-tam,inic+j);write(partmp[j]:tam.decimo[j]);
end;
end;

procedure makechangdens;
var i,xi,yi:word;
dens,conc:array [1..65] of real;
numdado:array[0..1] of byte;
dadoatual:array[0..1] of byte;
coef:array [0..2] of real;
desvio,valoraux,corr:real;
col,tam:byte;

procedure escreve_dados;
var i:byte;
procedure posiciona(coluna:byte);
begin
xi:=1 + ((i-1) div 13);
yi:=i - (xi-1)*13;
gotoxy((xi-1)*16+2+coluna*8,7+yi);
end;
begin
for i:=1 to numdado[0] do begin
posiciona(0);
write(dens[i]:7:4);
end;
for i:=1 to numdado[1] do begin
posiciona(1);
write(conc[i]:7:4);
end;
end;

procedure escreve_equacao;
var qtdd:byte;
msg,aux:string;
begin
if (numdado[0]>numdado[1]) then qtdd:=numdado[0] else
qtdd:=numdado[1];
gotoxy(5,22);
regressao_polinomial(conc,dens,coef,2,qtdd,corr,desvio);
if (calculook) then begin
str(coef[2]:5:5,aux);
msg:='DENS. = '+aux+' [ ]*[ ]';
str(coef[1]:5:5,aux);
aux:=aux+' [ ]';
if (coef[1]>=0) then msg:=msg+' '+aux else
msg:=msg+' '+aux;
str(coef[0]:5:5,aux);
if (coef[0]>=0) then msg:=msg+' '+aux else
msg:=msg+' '+aux;
str(desvio:5:5,aux);
write(msg,' ',chr(241),aux,' ');
str(corr:5:5,aux);
write(msg,' CORR. = ',aux,' ');
end
else write(branco(70));
end;

procedure deletadado;
var i:word;
begin
if (dadoatual[col]<=numdado[col]) then begin
if (col=0) then for i:=dadoatual[col] to numdado[col]-1 do
dens[i]:=dens[i+1];
if (col=1) then for i:=dadoatual[col] to numdado[col]-1 do
conc[i]:=conc[i+1];
dec(numdado[col]);
if (dadoatual[col]-1>numdado[col]) then
dadoatual[col]:=numdado[col]+1;
tela6;
escreve_dados;
end;
end;

procedure incrementadado;
var i:word;
begin
if (numdado[col]<65) and (dadoatual[col]<=numdado[col])
then begin
if (col=0) then begin
for i:=numdado[col] downto dadoatual[col] do
dens[i+1]:=dens[i];
dens[dadoatual[col]]:=0;
end;
if (col=1) then begin
for i:=numdado[col] downto dadoatual[col] do
conc[i+1]:=conc[i];
conc[dadoatual[col]]:=0;
end;
inc(numdado[col]);
escreve_dados;
end;
end;

begin
tela6;
for i:=0 to 1 do begin
numdado[i]:=0;
dadoatual[i]:=1;
end;
ch:=#0;
col:=0;
for i:=1 to 65 do begin
conc[i]:=0;
dens[i]:=0;
end;
repeat
if (ch=up) then dec(dadoatual[col]);
if (ch=dn) then inc(dadoatual[col]);
if (ch=F10) then col:=col xor 1;
if (ch=pgup) then incrementadado;
if (ch=pgdn) then deletadado;
if (dadoatual[col]<1) then dadoatual[col]:=1;
if (dadoatual[col]-1> numdado[col]) then
dadoatual[col]:=numdado[col]+1;
escreve_equacao;
xi:=1 + ((dadoatual[col]-1) div 13);
yi:=dadoatual[col] - (xi-1)*13;
if (col=0) then valoraux:=dens[dadoatual[col]]
else valoraux:=conc[dadoatual[col]];
if (dadoatual[col]>numdado[col]) then aux:=""
else str(valoraux:7:4,aux);
repeat
texto_crt((xi-1)*16+2+col*8,7+yi,7,"",aux,ch,1,7);
if (ch=#13) then val(aux,valoraux,code);
until ((ch=#13) and (code=0) and (valoraux >=0)) or
(ch in [#27,up,dn,pgup,pgdn,F10,#61]);
if (col=0) then dens[dadoatual[col]]:=valoraux

```

```

else conc[dadoatual[col]]:=valoraux;
if ((code=0) and (valoraux >=0) and (ch=#13)) then
begin
  if (dadoatual[col]<65) then inc(dadoatual[col]);
  if (dadoatual[col]-1>numdado[col]) then
inc(numdado[col]);
  escreve_dados;
  end;
  if (ch in [up,dn,#61,F10,#27]) then escreve_dados;
  until (ch in [#27,#61]);
  if (ch=#61) then begin
    par[15]:=coef[2];
    par[14]:=coef[1];
    par[13]:=coef[0];
  end;
  make_tela;
  gravacao_dados;
  ch:=#0;
end;

begin
  fora:=false;
  make_tela;
  i:=1;
  repeat
    if (not fora) then begin
      if (ch=up) then dec(i);
      if (ch in [Dn,#13]) then inc(i);
      if (i=0) then i:=qtddpar;
      if (i=qtddpar+1) then i:=1;
    end;
    fora:=false;
    str(partmp[i]:tam.decimo[i],aux);
    if (ch<>#61) then begin
      repeat
        texto_crt(56-tam,inic+i,tam,"aux,ch,1,7);
        val(aux,partmp[i],code);
        until (code=0) or (ch=#27);
      end;
      if (ch=#27) then exit;
      if (partmp[i] > condmax[i]) or (partmp[i] < condmin[i])
then begin
        fora:=true;
        gotoxy(56-tam,inic+i);write('FAIXA =
',condmin[i]:tam.decimo[i]
', - ',condmax[i]:tam.decimo[i]);
        delay(500);
        gotoxy(56-tam,inic+i);write(branco(30));
        if (partmp[i] > condmax[i]) then partmp[i]:=condmax[i];
        if (partmp[i] < condmin[i]) then partmp[i]:=condmin[i];
      end;
      gotoxy(56-tam,inic+i);write(partmp[i]:tam.decimo[i]);
      if (ch=#68) then makechangdens;
      if (ch=#61) then begin
        i:=1;
        repeat
          inc(i);
          if (partmp[i]>condmax[i])or(partmp[i]<condmin[i])
then fora:=true;
          until (i=qtddpar) or fora;
        end;
        until (ch=#61);
        gravacao_dados;
      end;
    end;
  procedure troca_pto(var arq:string;var ch:char;qtdd:word);
  var f,g:text;
  i,code,qtddcorr:word;
  tensaop:real;
  dado,masa:array [1..10] of real;
  aux,aux2:string;
begin
  aux="";
  repeat

```

```

  texto_crt(10,10,1,'QTDDE PTOS PARA CORRIGIR =
',aux,ch,1,7);
  val(aux,qtddcorr,code);
  until ((ch=#27) or ((code=0) and (qtddcorr>=1) and
(qtddcorr<qtdd)));
  if (ch=#27) then exit;
  assign(f,arq);
  assign(g,'$$$$$$$.$$$');
  reset(f);
  rewrite(g);
  for i:=1 to 4 do begin
    readln(f,aux2);
    if (i=2) then aux2:=aux2+'g';
    writeln(g,aux2);
  end;
  for i:=1 to qtddcorr do
  readln(f,dado[i],tensaop,masa[i]);
  while (not eof(f)) do begin
    readln(f,dado[qtddcorr+1],tensaop,masa[qtddcorr+1]);
    writeln(g,dado[1]:9:7,'',tensaop:5:2,'',masa[1]:6:4);
    for i:=1 to qtddcorr do begin
      dado[i]:=dado[i+1];
      masa[i]:=masa[i+1];
    end;
  end;
  close(f);
  close(g);
  aux:=get_dir_name;
  abre_arquivo(f,aux,'tns','NOVO NOME DO ARQUIVO :
','?',ch);
  if (ch=#27) then exit;
  close(f);
  erase(f);
  rename(g,aux);
  arq:=aux;
end;

function liga_balanca:boolean;
var ch:char;
begin
  repeat
    ch:=#0;
    inicia_comunicacao(com2,1200,'n',1,8,'i');
    if (balancaoff) then begin
      repeat
        erro('ERRO COMUN. BALANCA. VERIFIQUE E
TECLE [ENTER]',ch,true);
        until (ch in [#13,#27]);
      end;
      termina_aquisicao_bombas;
      until (not balancaoff) or (ch=#27);
      liga_balanca:=not balancaoff;
    end;
  end;

function get_dir_name:string;
var dirname:string;
begin
  getdir(0,dirname);
  dirname:=dirname+'\';
  if (paramcount=1) then dirname:=dirname+paramstr(1)+'\';
  get_dir_name:=dirname;
end;

begin
  f_chkplaca:=false;
END.

unit utisttc;
interface
uses unmtm,unregres;
procedure teste_q(var dado:array of real;var
qtdd:word;confianca:word);

```

```

procedure teste_discordancia(var dado:array of real; var
qtdd:word;discordancia:word);
implementation
const
  tabelaR10:array [1..6,3..30] of word = (
(886,679,557,482,434,399,370,349,332,318,305,294,285
,277,
269,263,258,252,247,242,238,234,230,227,224,220,218,215)
,
(941,765,642,560,507,468,437,412,392,376,361,349,338
,329,
320,313,306,300,295,290,285,281,277,273,269,266,263,260)
,
(970,829,710,625,568,526,493,466,444,426,410,396,384
,374,
365,356,349,342,337,331,326,321,317,312,308,305,301,298)
,
(976,846,729,644,586,543,510,483,460,441,425,411,399
,388,
379,370,363,356,350,344,338,333,329,324,320,316,312,309)
,
(988,889,780,698,637,590,555,527,502,482,465,450,438
,426,
416,407,398,391,384,378,372,367,362,357,353,349,345,341)
,
(994,926,821,740,680,634,598,568,542,522,503,488,475
,463,
452,442,433,425,418,411,404,399,393,388,384,380,376,372)
);
function get_T(qtdd,nivel:word):real;
const coefic:array [1..2,0..4] of real
=((-0.68988,2.28704,-0.67297,0.11333,-0.00762),
(-0.59991,1.87112,-0.26827,0.00664,0.00138));
var statisticalT:real;
ind:word;
begin
if (nivel=5) then ind:=1 else if (nivel=1) then ind:=2 else
exit;
statisticalT:=polinomio(ln(qtdd),4,coefic[ind]);
get_T:=statisticalT;
end;

function get_discordancia(discordancia:word):byte;
var valor:byte;
begin
if (discordancia=5) then valor:=1 else if (discordancia=1)
then valor:=2
else valor:=0;
get_discordancia:=valor;
end;

function get_confianza(confianza:word):byte;
var valor:byte;
begin
if (confianza=80) then valor:=1
else if (confianza=90) then valor:=2
else if (confianza=95) then valor:=3
else if (confianza=96) then valor:=4
else if (confianza=98) then valor:=5
else if (confianza=99) then valor:=6
else valor:=0;
get_confianza:=valor;
end;

procedure get_max_min(var dado:array of real;var
min,vizmin,max,vizmax:real;
var indicemax,indicemin:word;qtdd:word);
var i:word;

```

```

begin
max:=-1.7E38;
vizmax:=-1.7E38;
min:=1.7E38;
vizmin:=1.7E38;
for i:=0 to qtdd-1 do begin
if (dado[i] > vizmax) then begin
vizmax:=dado[i];
if (vizmax > max) then begin
vizmax:=max;
max:=dado[i];
indicemax:=i;
end;
end;
if (dado[i] < vizmin) then begin
vizmin:=dado[i];
if (vizmin < min) then begin
vizmin:=min;
min:=dado[i];
indicemin:=i;
end;
end;
end;
end;

procedure erase_dado(var dado:array of real;var
qtdd:word;indice:word);
var i:word;
begin
for i:=indice to qtdd-2 do dado[i]:=dado[i+1];
dec(qtdd);
end;

procedure teste_q;
var i,indice,indicemax,indicemin:word;
max,min,vizmax,vizmin:real;
testeqminok,testeqmaxok:boolean;
begin
indice:=get_confianza(confianza);
get_max_min(dado,min,vizmin,max,vizmax,indicemax,indi
cemin,qtdd);
repeat
if (((vizmin-min) / (max-min)) >=
tabelaR10[indice,qtdd]/1000) then begin
erase_dado(dado,qtdd,indicemin);

get_max_min(dado,min,vizmin,max,vizmax,indicemax,indi
cemin,qtdd);
testeqminok:=false;
end else testeqminok:=true;
if (((max-vizmax) / (max-min)) >=
tabelaR10[indice,qtdd]/1000) then begin
erase_dado(dado,qtdd,indicemax);

get_max_min(dado,min,vizmin,max,vizmax,indicemax,indi
cemin,qtdd);
testeqmaxok:=false;
end else testeqmaxok:=true;
until (testeqmaxok and testeqminok);
end;

procedure teste_discordancia;
var
experT,teorT,media,desvio,max,min,vizmax,vizmin,soma:r
eal;
indicemax,indicemin:word;
testeTminok,testeTmaxok:boolean;
procedure recalculo;
begin
teorT:=get_T(qtdd,discordancia);
get_max_min(dado,min,vizmin,max,vizmax,indicemax,indi
cemin,qtdd);
calcula_media(dado,qtdd,media,desvio,soma);
end;
begin

```



```

repeat
  recalculo;
  {$B-}
  if (desvio<>0) and (((media-mln)/desvio)>teorT) then
begin
  erase_dado(dado,qtdd,Indicemin);
  testeTminok:=false;
end else testeTminok:=true;
  recalculo;
  if (desvio<>0) and (((max-media)/desvio)>teorT) then
begin
  erase_dado(dado,qtdd,Indicemax);
  testeTmaxok:=false;
end else testeTmaxok:=true;
  until ((testeTminok) and (testeTmaxok));
end;
end.

```

UNIT UNTECL;

INTERFACE

function leteclado(back:boolean):char;

IMPLEMENTATION

USES CRT;

function leteclado(back:boolean):char;

var charac:char;

begin

charac:=#0;

if (not back) then begin

if keypressed then charac:=readkey;

end

else

repeat

if keypressed then charac:=readkey;

until (charac<>#0);

leteclado:=charac;

end;

END.

UNIT Unregres;

INTERFACE

procedure calcula_media(var dado:array of real; qtdd:word;var media,desvio,soma:real);

procedure regressao_polinomial(var dx,dy,coef:array of real;ordem,qd:word;var corr,desvio:real);

var calculook:boolean;

implementation

procedure calcula_media;

var somaquad,somaloc:real;

i:word;

begin

calculook:=true;

somaloc:=0;

somaquad:=0;

if (qtdd>1) then begin

for i:=0 to qtdd-1 do begin

somaloc:=somaloc+dado[i];

somaquad:=somaquad+sqr(dado[i]);

end;

media:=somaloc/qtdd;

desvio:=somaquad/qtdd*sqr(media);

desvio:=sqr(desvio/(qtdd-1));

soma:=somaloc;

end

else if (qtdd=1) then begin

desvio:=0;

media:=dado[0];

soma:=dado[0];

end

else calculook:=false;

end;

procedure regressao_polinomial;

var i,j,k:word;

somax:array [1..20,1..21] of real;

parcial:array [1..21] of real;

somay2,aux,ycalc:real;

function potencia(base:word;arg:real):real;

var result:real;

i:word;

begin

result:=arg;

if (base>0) then for i:=1 to base-1 do result:=result*arg

else result:=1;

potencia:=result;

end;

begin

inc(ordem);

if (ordem<=qd) then calculook:=true

else begin

calculook:=false;

exit;

end;

dec(qd);

for i:=1 to 20 do

for j:=1 to 21 do somax[i,j]:=0;

for i:=1 to ordem do

for k:=0 to qd do

somax[i,ordem+1]:=somax[i,ordem+1]+dy[k]*potencia(i-1,dx[k]);

for i:=1 to ordem do begin

for j:=i to ordem do begin

for k:=0 to qd do

somax[i,j]:=somax[i,j]+potencia(i+j-2,dx[k]);

if (i<>j) then somax[j,i]:=somax[i,j];

end;

end;

somay2:=0;

for i:=0 to qd do somay2:=somay2+dy[i]*dy[i];

somay2:=somay2-(sqr(somax[1,ordem+1]))/(qd+1));

for i:=1 to ordem-1 do

parcial[i]:=somax[i+1,ordem+1]-

(somax[1,ordem+1]*somax[1,i+1]/(qd+1));

for k:=1 to ordem do begin

for i:=k to ordem-1 do begin

if (somax[i+1,k]>somax[k,k]) then begin

for j:=k to ordem+1 do begin

aux:=somax[i+1,j];

somax[i+1,j]:=somax[k,j];

somax[k,j]:=aux;

end;

end;

aux:=somax[k,k];

for j:=k to ordem+1 do somax[k,j]:=somax[k,j] / aux;

for i:=k+1 to ordem do begin

aux:=somax[i,k];

for j:=k to ordem+1 do somax[i,j]:=somax[i,j]-

somax[k,j]*aux;

end;

end;

for k:=1 to ordem do begin

for i:=k to ordem-1 do begin

aux:=somax[k,i+1];

for j:=k to ordem+1 do somax[k,j]:=somax[k,j]-

somax[i+1,j]*aux;

end;

end;

for i:=1 to ordem do coef[i-1]:=somax[i,ordem+1];

desvio:=0;

for i:=0 to qd do begin

ycalc:=0;

for j:=0 to ordem-1 do

ycalc:=ycalc+coef[j]*potencia(j,dx[i]);

```

desvio:=desvio+sqrt(dy[i]-ycalc);
end;
if qd>=ordem then desvio:=sqrt(desvio/(qd-ordem+1))
else desvio:=0;
corr:=0;
for i:=1 to ordem-1 do corr:=corr+coef[i]*parcial[i];
corr:=sqrt(corr/somay2);
end;
end.

```

UNIT UNSTRING;

INTERFACE

```

function strings(ordin:real;tam,dec:byte):string;
function escreve_bit(palavra:longint):string;

```

IMPLEMENTATION

```

function strings(ordin:real;tam,dec:byte):string;
var aux : string;
begin
str(ordin:tam:dec,aux);strings:=aux;
end;

```

```

function escreve_bit(palavra:longint):string;
var bitstring:array [1..32] of char;
    len,i:byte;
begin
len:=0;
repeat
inc(len);
if (odd(palavra)) then bitstring[len]='1' else
bitstring[len]='0';
palavra:=palavra div 2;
until (palavra=0);
for i:=1 to len do escreve_bit[i]:=bitstring[len+1-i];
escreve_bit[0]:=chr(len);
end;
end.

```

UNIT UNRS232;

INTERFACE

```

var dadostr:string;
    balancaoff:boolean;
    pr:word;
const com1 = 1;
    com2 = 2;
    desligada='ERROR';
    base : array [1..2] of word = ($03f8,$02f8);

```

```

procedure
serial_transmissao(com:word;nome:string);interrupt;
Procedure serial_recepcao;interrupt;
procedure
inicia_comunicacao(porta,bau:word;par:char;stop,pal:byte;tipo
:char);
Procedure Termina_Comunicacao;

```

implementation

```

uses dos,crt;
var pt,pt2:WORD;
    save0c:pointer;
    save8259:byte;
    buffer:array[0..500] of byte;
    com:byte;
const Vect : array [1..2] of byte = ($0c,$0B);
    Mask : array [1..2] of byte = ($EF,$F7);

```

```

procedure
serial_protocolo(com,baud:word;paridade:char;stop,palavra:byte);
var controle:byte;
    reg:registers;
begin

```

```

case baud of
110:controle:= 0;
150:controle:= 1;
360:controle:= 2;
600:controle:= 3;
1200:controle:= 4;
2400:controle:= 5;
4800:controle:= 6;
9600:controle:= 7;
end;
case upcase(paridade) of
'N' : controle:=(controle shl 2) or 0;
'O' : controle:=(controle shl 2) or 2;
'E' : controle:=(controle shl 2) or 3;
end;
case stop of
1 : controle:=(controle shl 1) or 0;
2 : controle:=(controle shl 1) or 1;
end;
case palavra of
7 :controle:=(controle shl 2) or 2;
8 :controle:=(controle shl 2) or 3;
end;
reg.al:=Controle;
reg.ah:=0;
reg.dh:=com-1;
intr($14,reg);
end;

```

```

procedure serial_transmissao(com:word;nome:string);
var reg:registers;
    z:real;
    c:char;

```

```

begin
pt:=length(nome);
pt2:=0;
while (pt>pt2) do begin
if (port[base[com]+5] and $20)=$20 then
begin
port[base[com]]:=ord(nome[pt2+1]);
DELAY(2);
inc(pt2);
end;
delay(80);
end;
pt:=0;
pt2:=0;
end;

```

```

procedure serial_recepcao;
var i:integer;
begin
buffer[pr]:=(port[base[com]]);
buffer[pr+1]:=0;
if (buffer[pr] = 10) and (buffer[pr-1] = 13) then begin
for i:=1 to pr-1 do DadoStr[i]:=chr(buffer[i]);
DadoStr[0]:=chr(pr-2);
pr:=0;
end;
inc(pr);
if (pr>450) then pr:=0;
port[$20]:=$20;
end;

```

```

procedure habilita_reg_serial(com:word);
begin
port[base[com]+4] := $0f;
port[base[com]+1] := $01;
port[$21] := port[$21] and Mask[Com];
end;

```

```

procedure
inicia_comunicacao(porta,bau:word;par:char;stop,pal:byte;tipo
:char);
var k:integer;

```

```

    ch:char;
begin
    dadostr:="";
    Com:=Porta;
    pr:=1;
    serial_protocolo(com,bau,par,stop,pal);
    getintvec(vec[com],save0c);
    save8259:=port[$21];
    if (upcase(tipo)='I') then
    setintvec(vec[com],@serial_recepcao)
    else if (upcase(tipo)='O') then
    setintvec(vec[com],@serial_transmissao)
    else exit;
    habilita_reg_serial(com);
    k:=0;
    repeat
        delay(100);
        inc(k)
    until (DadoStr[1]='+') or (DadoStr[1]='-') or (k>20);
    if (k>20) then begin
        balancaoff:=true;
        dadostr:=desligada;
    end
    else begin
        balancaoff:=false;
        repeat
            if keypressed then ch:=upcase(readkey);
            until ((dadostr[length(dadostr)]='G') or (ch=#27));
        end;
    end;
end;

procedure termina_Comunicacao;
begin
    setintvec(Vect[com],save0c);
    port[$21]:=save8259;
end;
end.

```

UNIT UNIRQMQI;

INTERFACE

```

var irq,savemsk : byte;
    SaveVec : pointer;
    clock : real;
    taxa : word;
procedure Inicia_Aquisicao (ponteiro : pointer);
procedure Termina_Aquisicao;
procedure termina_aquisicao_bombas;
procedure ChkClk(intirq:byte);
procedure end_of_interrupt;

```

IMPLEMENTATION

```

Uses Dos;
var cont,flag : integer;

Const
    MskInt : array [2..9] of byte =
    ($Fb,$F7,$EF,$DF,$BF,$7F,$FF,$FD);
    MskAdr : array [2..9] of byte =
    ($21,$21,$21,$21,$21,$21,$A1,$A1);
    VectInt : array [2..9] of byte =
    ($0A,$0B,$0C,$0D,$0E,$0F,$1C,$71);
    EolAdr : array [2..9] of byte =
    ($20,$20,$20,$20,$20,$20,$A0,$A0);
    Eol : byte = $20; { End of Interrupt }

procedure programa_timer(clock:real; taxa_timer:word);
var
    modo,tm1a,tm1b,tm2a,tm2b:byte;
    tm1,tm2:word;
    nx:real;
begin
    NX:=clock/taxa_timer;

```

```

    tm1:=(round(NX / 65536))+2;
    tm2:=round(NX/tm1);
    if tm2 < 2 then tm2:=2;
    tm1a:=lo(tm1);
    tm1b:=hi(tm1);
    tm2a:=lo(tm2);
    tm2b:=hi(tm2);
    modo := $34;
    port[$307] := modo;
    port[$304] := tm1a;
    port[$304] := tm1b;
    modo := $74;
    port[$307] := modo;
    port[$305] := tm2a;
    port[$305] := tm2b;
end;

procedure Inicia_Aquisicao;
begin
    getintvec(vecint[irq],SaveVec); {salva vetor de
interrupcao}
    SaveMsk := Port[MskAdr[IRQ]]; {salva
mascara}
    setintvec(vecint[irq],Ponteiro); {seta vetor de
interrupcao}
    programa_timer(clock,taxa); {programa timer }
    port[MskAdr[IRQ]] := savemsk and MskInt[IRQ];
{programa mascara}
    port[$310]:=$c8; {habilita timer}
end;

Procedure Termina_Aquisicao;
begin
    port[$310]:=$08; {desabilita timer}
    port[MskAdr[IRQ]] := SaveMsk; {desprograma
mascara}
    setintvec(vecint[irq],SaveVec); {restaura vetor de
interrupcao}
end;

procedure termina_aquisicao_bombas;
begin
    termina_aquisicao;
    port[$303]:=$89;
    port[$300]:=$10; {retira energia das
bobinas}
    port[$301]:=$10;
End;

procedure contador;
interrupt;
begin
    inc(cont);
    port[$20]:=$20;
end;

procedure bloqueia;
interrupt;
begin
    if flag=0 then begin
        SaveMsk := Port[MskAdr[IRQ]]; {Salva mascara
de interrupcao.}
        port[MskAdr[IRQ]]:=SaveMsk and MskInt[IRQ];
        { Esta porta corresponde ao endereco da mascara de
interrupcao, onde e
redirecionado o vetor de interrupcao, para outro
ponteiro, definido
por esta interrupcao }
        port[$310]:=$c8; {habilita timer da placa}
    end;
    inc(flag);
    if flag=11 then begin
        port[MskAdr[IRQ]] := SaveMsk; {retorna a antiga
mascara }
    end;
end;

```

```

    port[$310]=$08;          {desabilita timer da
placa}
end;
end_of_interrupt;
end;

procedure chkclk(intirq:byte);
var
    save2:pointer;
    taxa_clk:word;
begin
    taxa_clk:=3000;
    irq:=intirq;
    repeat                {verifica clock e numero da
interrupcao}
        inc(irq);
        clock := 2.5e6;    {mascaravel disponivel no
micro (IRQ)}
        getintvec(Veclnt[IRQ],SaveVec);
        getintvec($1c,save2);
        programa_timer(clock,taxa_clk);
        flag:=0;
        cont:=0;
        setintvec(Veclnt[IRQ],@contador); {Acessa as
interrupcoes

mascaraveis de numeros 1 a 9}
        setintvec($1c,@bloqueia);
{ inicia um contador de tempo reduzido que redireciona o
endereco da
interrupcao mascaravel (mascAra a interrupcao) }
        repeat
            until flag=11;
            setintvec(Veclnt[IRQ],SaveVec);
            setintvec($1c,save2);
            clock:=(cont/(0.5486666666))*clock/taxa_clk;
{ Se a interrupcao mascaravel de numero IRQ estiver
disponivel, a
variavel cont sera diferente de zero e terminarah o 'loop'.
}
            until (clock>0) or (irq>=7);
        end;

procedure End_of_Interrupt;
begin
    port[EolAdr[IRQ]] := Eol;
end;

begin
    taxa:=5000;
end.

UNIT UTIARQ;

INTERFACE
procedure abre_arquivo(var f:text;var
arq:string;ext,msg:string;tipo:char;var ch:char);
procedure fecha_arquivo(var f:text);
procedure
grava_dados(arqname:string;dadopointer:pointer;tam:word);
function
le_dados(arqname:string;dadopointer:pointer;tam:word):booie
an;

IMPLEMENTATION
uses utitexto,unarquiv,uncrt;

procedure abre_arquivo;
var f_sair:boolean;
    aux:string;
procedure abre(var tipo:char);
begin
    {$i-}

```

```

    assign(f,aux);
    case tipo of
        'O','?', 'W' : rewrite(f);
        'A' : append(f);
        'R','I' : reset(f);
    end;
    f_sair:=true;
    arq:=aux
end;
begin
    tipo:=upcase(tipo);
    if not (tipo in ['O','A','R','I','W']) then tipo:='?';
    repeat
        repeat
            ch:=#0;
            aux:=arq;
            f_sair:=false;
            texto_crt(10,10,length(aux)+8,msg,aux,ch,1,7);
            if (ch=#27) then exit;
            aux:=aux+'.'+ext;
            if (not nome_correto(aux)) then erro('NOME ARQ.
INCORRETO',ch,false);
            until (nome_correto(aux));
            if (existe_arquivo(aux)) then begin
                if (tipo='?') then begin
                    repeat
                        erro('ARQ. JA EXISTE.
OVERWRITE/APPEND/NEW ? [O/A/N/Esc]',ch,true);
                        until (ch in ['O','A','N',#27]);
                        if (ch in ['O','A']) then tipo:=ch;
                        if (ch=#27) then f_sair:=true;
                    end;
                    if (tipo in ['O','A','R','W','I']) then abre(tipo);
                end
                else begin
                    if (tipo in ['R','I']) then erro('ARQUIVO NAO
EXISTE.',ch,false);
                    if (tipo in ['O','A','?', 'W']) then abre(tipo);
                end;
            until (f_sair);
        end;

procedure fecha_arquivo(var f:text);
begin
    close(f);
    {$i+}
    if (IOResult<>0) then erro('ERRO NA
GRAVACAO.',ch,false);
end;

procedure grava_dados;
var option:char;
begin
    repeat
        option:='N';
        grava_config(arqname,dadopointer,tam);
        if (ioresult<>0) then
            erro('ERRO DE GRAVACAO. REPETIR ? [S/N]
',option,true);
        until (option in ['N',#27]);
    end;

function le_dados;
var option:char;
    i:byte;
    erroread:boolean;
begin
    erroread:=true;
    if (existe_arquivo(arqname)) then begin
        repeat
            option:=#0;
            carrega_config(arqname,dadopointer,tam);
            if (ioresult=0) then erroread:=false;
            if erroread then

```

```

        erro('ERRO DE LEITURA EM '+arqname+'.
REPETIR ? [S/N],option,true);
        until (not erroread) or (option in ['N',#27]);
        end;
        le_dados:=not erroread;
end;
END.

```

UNIT UTITEXTO;

INTERFACE

```

procedure texto_crt(col,lin,ext:byte;msg:string;var
nome:string;var ch:char;car,fn:byte);

```

IMPLEMENTATION

USES crt,uncrt;

```

procedure texto_crt;
var prima,len:byte;
    flag:boolean;

```

```

procedure escreve_nome;
begin
    gotoxy(col,lin);
    write(nome);
    gotoxy(col+prima,lin);write("");
end;

```

```

procedure trata_Home;
begin
    prima:=0;
    gotoxy(col+prima,lin);
end;

```

```

procedure trata_Up;
begin
end;

```

```

procedure trata_PgUp;
begin
end;

```

```

procedure trata_Lf;
begin
    if (prima>0) then begin
        dec(prima);
        gotoxy(col+prima,lin);
    end;
end;

```

```

procedure trata_Rg;
begin
    if (prima<len) then begin
        inc(prima);
        gotoxy(col+prima,lin);
    end;
end;

```

```

procedure trata_End;
begin
    prima:=len;
    gotoxy(col+prima,lin);
end;

```

```

procedure trata_Dn;
begin
end;

```

```

procedure trata_PgDn;
begin
end;

```

```

procedure trata_Ins;

```

```

begin
    if (len<ext) then begin
        insert(' ',nome,prima+1);
        inc(len);
        escreve_nome;
        gotoxy(col+prima,lin);
    end;
end;

```

```

procedure trata_Del;
begin
    if (prima<len) then begin
        delete(nome,prima+1,1);
        gotoxy(col+len-1,lin);write(' ');
        dec(len);
        escreve_nome;
        gotoxy(col+prima,lin);
    end;
end;

```

```

procedure trata_BS;
begin
    if (prima>0) then begin
        delete(nome,prima,1);
        gotoxy(col+len-1,lin);write(' ');
        dec(prima);
        dec(len);
        escreve_nome;
        gotoxy(col+prima,lin);
    end;
end;

```

```

procedure trata_Ch;
begin
    if (prima<ext) then begin
        if (prima=len) then inc(len);
        inc(prima);
        nome[0]:=chr(len);
        nome[prima]:=ch;
    end;
end;

```

```

begin
    gotoxy(col,lin);write(msg);
    col:=col+length(msg);
    cor(car,fn);
    gotoxy(col,lin);write(branco(ext));
    len := length(nome);
    prima:=0;
    flag:=true;
    escreve_nome;
    repeat
        ch:=upcase(Readkey);
        if (ch=#0) then begin
            ch:=ReadKey;
            if ch in[#71,#75,#77,#79,#82,#83] then flag:=false;
            if ch in[#72,#73,#80,#81] then flag:=true;
            ch:=chr(128+ord(ch));
        end;
        if ((ch>=#128) and flag) or (ch in [#13,#27,#09]) then
begin
            escreve_nome;
            if (ch>=#128) then ch:=chr(ord(ch)-128);
            cor(fn,car);
            gotoxy(col,lin);write(branco(ext));
            escreve_nome;
            exit;
        end;
        if flag then begin
            gotoxy(col,lin);write(branco(ext));
            nome:="";
            prima:=0;
            len:=0;
        end;
        flag:=false;
    until false;
end;

```

```

case ch of
  chr(ord('G')+128) : trata_Home;
  chr(ord('H')+128) : trata_Up;
  chr(ord('I')+128) : trata_PgUp;
  chr(ord('K')+128) : trata_Lf;
  chr(ord('M')+128) : trata_Rg;
  chr(ord('O')+128) : trata_End;
  chr(ord('P')+128) : trata_Dn;
  chr(ord('Q')+128) : trata_PgDn;
  chr(ord('R')+128) : trata_Ins;
  chr(ord('S')+128) : trata_Del;
  #8      : trata_BS;
  #32 .. #122 : trata_Ch;
end;
if (ch<#128) and (ch<>#8) then begin
  escreve_nome;
  gotoxy(col+prima,lin);write("");
end;
if ch=#0 then ch:=ReadKey;
until false;
end;
end.

UNIT UNCRT;

INTERFACE
const
Enter = #13; Esc = #27; F1 = #59; F6
      = #64; Up = #72; off = false;
PgUp = #73; Nul = #0; F2 = #60;
F7 = #65; Le = #75; Ed = #79;
Home = #71; Del = #83; F3 = #61; F8
      = #66; Ri = #77; on = true;
PgDn = #81; Ins = #82; F4 = #62; F9
      = #67; Dn = #80;
Space = #32; Tab = #09; F5 = #63; F10=
#68; Bs = #8;
var ch:char;
function branco(ct:byte) :string;
procedure cor(C,F:byte);
procedure erro(ms:string;var option:char;le:boolean);
Procedure beep;
procedure aviso_sonoro(freq,delayms,delaynosom:word);
procedure avisa_fim(freq,delayms,delaynosom:word);

IMPLEMENTATION
uses crt;

function branco(ct:byte):string;
var st:string;
begin
  fillchar(st,ct+1,#32);
  st[0] := chr(ct);
  branco:= st;
end;

procedure cor(c,f:byte);
begin
  textcolor(c);
  textbackground(f);
end;

procedure beep;
begin
  sound(800);delay(100);nosound;delay(1000);
end;

procedure aviso_sonoro(freq,delayms,delaynosom:word);
begin
  sound(freq);delay(delayms);nosound;delay(delaynosom);
end;

procedure avisa_fim(freq,delayms,delaynosom:word);

```

```

var i:word;
option:char;
begin
  option:=#0;
  repeat
    aviso_sonoro(freq,delayms,0);
    i:=0;
    repeat
      inc(i);
      delay(100);
      if keypressed then option:=upcase(readkey);
    until (i>=delaynosom div 100) or (option=#27);
  until (option=esc);
end;

procedure erro;
var a : array [0..79] of word;
i : word;
begin
  for i:=0 to 79 do a[i]:=memw[$b800:i*2+3680];
  gotoxy(10,24);
  write(branco(length(ms)+2));
  cor(1,7);
  gotoxy(11,24);write(' '+ms+' ');
  cor(7,1);
  beep;
  if (le) then begin
    option:=#0;
    repeat
      if keypressed then option:=upcase(readkey);
    until (option <>#0);
  end;
  for i:=0 to 79 do memw[$b800:i*2+3680]:=a[i];
end;
end.

```

UNIT UNTELA;

```

INTERFACE
procedure
divide_tela_crtx(posx,posy1,posy2:word;figura:byte);
procedure
divide_tela_crty(posx,posy1,posy2:word;figura:byte);
procedure
prepara_tela(ntit,nfr1,nfr2,nmsg:byte;strmsg:string);
procedure tela1;
procedure tela2;
procedure tela3;
procedure tela4;
procedure tela6;
procedure tela_ensaio;
procedure tela_analise;

IMPLEMENTATION
uses crt;

procedure
divide_tela_crty(posx,posy1,posy2:word;figura:byte);
var i:byte;
begin
  gotoxy(posx,posy1);write(chr(194));
  for i:=posy1+1 to posy2 do begin
    gotoxy(posx,i);write(chr(figura)); {179}
  end;
  gotoxy(posx,posy2);write(chr(193));
end;

procedure
divide_tela_crtx(posx1,posx2,posy:word;figura:byte);
var i:byte;
begin
  gotoxy(posx1,posy);write(chr(199));
  for i:=posx1+1 to posx2 do begin

```

```

    gotoxy(i, posy); write(chr(figura)); {196}
end;
gotoxy(posx2, posy); write(chr(182));
end;

procedure moldura;
var i : Integer;
    ls, ld : string[80];
    ch1, ch2, ch3, ch4, ch5, ch6, ch7, ch8, ch9, ch0 : char;
begin
    textbackground(1);
    textcolor(7);
    ld := ' '; ls := '';
    ch1 := #201; { É }
    ch2 := #187; { » }
    ch3 := #200; { È }
    ch4 := #188; { ¼ }
    ch5 := #199; { Ç }
    ch6 := #182; { ¶ }
    ch7 := #186; { ° }
    ch8 := #194; { Å }
    ch9 := #179; { ¸ }
    ch0 := #207; { ¡ }
    fillchar(ls, 79, #196); { Å }
    ls[0] := #78;
    fillchar(ld, 79, #205); { ¡ }
    ld[0] := #78;
    clrscr;
    gotoxy(1, 1); write(ch1+ld+ch2);
    for i:=2 to 24 do begin
        gotoxy(1, i); write(ch7);
        gotoxy(80, i); write(ch7);
    end;
    gotoxy(1, 3); write(ch5+ls+ch6);
    gotoxy(1, 5); write(ch5+ls+ch6);
    gotoxy(1, 23); write(ch5+ls+ch6);
    gotoxy(1, 25); write(ch3+ld);
    mem[$b800:3998] := 188;
    gotoxy(3, 24); write('MSG:');
    gotoxy(8, 23); write(ch8);
    gotoxy(8, 24); write(ch9);
    gotoxy(8, 25); write(ch0);
end;

procedure Monta_tela(titulo, frase1, frase2, msg: string);
begin
    moldura;
    gotoxy(41-round(length(titulo)/2), 2);
    write(titulo);
    gotoxy(79-length(frase2), 24); write(frase2);
    gotoxy(41-round(length(frase1)/2), 4); write(frase1);
    gotoxy(10, 24); write(msg);
end;

procedure
prepara_tela(ntit, nfr1, nfr2, nmsg: byte; strmsg: string);
var titulo, frase1, frase2, mensagem: string;
begin
    case ntit of
        0 : titulo := strmsg;
        1 : titulo := 'TENSIOMETRIA';
        2 : titulo := 'DETERMINAÇÃO DE CIANE
TO';
        3 : titulo := 'ELETRODISSOLUÇÃO';
        4 : titulo := 'LEITURA DE BALANCA';
        else titulo := '';
    end;
    case nfr1 of
        0 : frase1 := strmsg;
        1 : frase1 := 'MENU PRINCIPAL';
        2 : frase1 := 'ENSAIO VARIÁVEL';
        3 : frase1 := 'CADASTRO DE SERINGA';
        4 : frase1 := 'RAIO EFETIVO';
        5 : frase1 := 'ANÁLISE DADO VARIÁVEL';
        6 : frase1 := 'RETORNO DAS BOMBAS';
    end;

```

```

7 : frase1 := 'PASSO DA BOMBA';
8 : frase1 := 'ENSAIO SIMPLES';
9 : frase1 := 'ALTERAÇÃO DE PARÂMETROS';
10 : frase1 := 'ANÁLISE DADO SIMPLES';
11 : frase1 := 'CALIBRAÇÃO';
12 : frase1 := 'ANÁLISE';
13 : frase1 := 'REGRESSÃO DE DADOS';
14 : frase1 := 'SENHA DE ACESSO';
15 : frase1 := 'ENSAIO PROGRAMADO';
16 : frase1 := 'ANÁLISE DE AMOSTRA';
17 : frase1 := 'LIMPEZA DAS VIAS';
18 : frase1 := 'REGRESSÃO DENS. x CONC.';
19 : frase1 := 'MENU ENSAIO';
20 : frase1 := 'MEDIDA';
21 : frase1 := 'PARÂMETROS DE VALVULAS';
22 : frase1 := 'INJEÇÃO DE VALVULAS';
23 : frase1 := 'VERIFICAÇÃO DO DETECTOR';
24 : frase1 := 'L E N D O D A D O S';
    else frase1 := '';
end;
case nfr2 of
    0 : frase2 := strmsg;
    1 : frase2 := 'LABORATORIO 305/303/205';
    2 : frase2 := 'LABORATORIO 209';
    3 : frase2 := 'UFV';
    4 : frase2 := 'LABORATORIO 209/211';
    5 : frase2 := 'LEHMKUHL, A.';
    else frase2 := '';
end;
case nmsg of
    0 : mensagem := strmsg;
    1 : mensagem := '[ESC] EXIT';
    2 : mensagem := '[ESC] MENU PRINCIPAL';
    3 : mensagem := '[ESC] ABORTA';
    4 : mensagem := '[ESC] RETORNA';
    else mensagem := '';
end;
if (ntit <> 0) and (nfr1 <> 0) and (nfr2 <> 0) and (nmsg <> 0)
then
    mensagem := mensagem + strmsg;
    monta_tela(titulo, frase1, frase2, mensagem);
end;

Procedure tela1;
begin
    prepara_tela(1, 1, 1, 1, '');
    gotoxy(22, 06); write('[F1] ENSAIOS');
    gotoxy(22, 08); write('[F2] CADASTRO DE SERINGAS');
    gotoxy(22, 10); write('[F3] DETERMINAR RAO
EFETIVO');
    gotoxy(22, 12); write('[F4] ANÁLISE DE DADOS');
    gotoxy(22, 14); write('[F5] RETORNO DAS BOMBAS');
    gotoxy(22, 16); write('[F6] DETERMINAR PASSO DA
BOMBA');
    gotoxy(22, 18); write('[F7] ALTERAÇÃO DOS
PARÂMETROS');
    gotoxy(22, 20); write('[F8] CÁLCULO DA TENSÃO
(MANUAL)');
    gotoxy(22, 22); write('[F9] ENSAIO DISCRETO');
end;

procedure tela2;
begin
    prepara_tela(1, 1, 1, 1, '');
    gotoxy(22, 06); write('[F1] ENSAIOS');
    gotoxy(22, 08); write('[F2] CADASTRO DE SERINGAS');
    gotoxy(22, 10); write('[F3] DETERMINAR RAO
EFETIVO');
    gotoxy(22, 12); write('[F4] ANÁLISES DE DADOS');
    gotoxy(22, 14); write('[F5] RETORNO DAS BOMBAS');
    gotoxy(22, 16); write('[F6] DETERMINAR PASSO DA
BOMBA');
    gotoxy(22, 18); write('[F7] CONFERE DETECTOR');
    gotoxy(22, 20); write('[F8] ALTERAÇÃO DE
PARÂMETROS');
end;

```

```

end;

procedure tela3;
begin
  prepara_tela(1,3,1,2,"");
  gotoxy(10,10);write('[F1] INCLUE SERINGA');
  gotoxy(10,12);write('[F2] EXCLUE SERINGA');
  gotoxy(10,14);write('[F3] VER CADASTRO');
end;

procedure tela4;
begin
  prepara_tela(1,9,1,4,' [F3] GRAVA [F10] REGR.
DENS.);
  gotoxy(10,07);write('CONC. DA SERINGA.....[molar] :
');
  gotoxy(10,08);write('CONC. DO
MISTURADOR.....[molar] : ');
  gotoxy(10,09);write('VOLUME DO
MISTURADOR.....[ml] : ');
  gotoxy(10,10);write('DENSIDADE DA
SERINGA.....[g/ml] : ');
  gotoxy(10,11);write('DENSIDADE DO
MISTURADOR.....[g/ml] : ');
  gotoxy(10,12);write('PASSO DA BOMBA.....
[passo/cm] : ');
  gotoxy(10,13);write('RAIO EFETIVO CAPILAR.....[cm]
: ');
  gotoxy(10,14);write('VELOCIDADE DA
BOMBA.....[passo/s] : ');
  gotoxy(10,15);write('QTDADA GOTAS PARA
CALIBRAR..... : ');
  gotoxy(10,16);write('TEMPERATURA
EXPERIMENTAL.....[+chr(248)+k] : ');
  gotoxy(10,17);write('N MOLES total / N MOLES surf..... :
');
  gotoxy(10,18);write('ACELERACAO
GRAVIDADE.....[cm/s*s] : ');
  gotoxy(10,19);write('COEF. GRAU 0 (DENS. = f
(CONC.)) : ');
  gotoxy(10,20);write('COEF. GRAU 1 ([g/ml] = [MOLAR]
) : ');
  gotoxy(10,21);write('COEF. GRAU 2 ( " " " ) : ');
  gotoxy(10,22);write('VELOC. BOMBA MODO RAPIDO
[passo/s] : ');
end;

procedure tela6;
var j:word;
begin
  prepara_tela(1,18,1,4,' [F3] GRAVA [F10] COLUNA');
  divide_tela_crtx(1,80,7,196);
  divide_tela_crtx(1,80,21,196);
  for j:=1 to 5 do begin
    gotoxy(2+(j-1)*16,6);
    write(' DENS. CONC. ');
    divide_tela_crtx(9+(j-1)*16,5,21,179);
    if (j<5) then divide_tela_crtx(17+(j-1)*16,5,21,186);
  end;
end;

procedure tela_ensaio;
begin
  prepara_tela(1,19,1,4,"");
  gotoxy(10,10);write('[F1] CONCENTRACAO VARIADA
EXPONENCIAL');
  gotoxy(10,13);write('[F2] CONCENTRACAO VARIADA
LINEAR');
  gotoxy(10,16);write('[F3] FLUXO VARIADO');
  gotoxy(10,19);write('[F4] SEM VARIACAO');
  gotoxy(10,22);write('[F5] CONCENTRACAO VARIADA
DUAS BOMBAS');
end;

procedure tela_analise;

```

```

begin
  prepara_tela(1,19,1,4,"");
  gotoxy(10,10);write('[F1] CONCENTRACAO VARIADA');
  gotoxy(10,13);write('[F2] FLUXO VARIADO');
  gotoxy(10,16);write('[F3] SEM VARIACAO');
end;
end.

UNIT UNMTM;

INTERFACE
function log(arg,base:real):real;
function expbase(arg,base:real):real;
function uptrunc(arg:real):longint;
function pot(arg,base:real):real;
function potencia(arg:real;base:integer):real;
function raizN(arg,base:real):real;
function bascara(ca,cb,cc:real;var r1,r2:real):boolean;
function polinomio(ptox:real;ordem:word;coef:array of
real):real;
var calculomtmok:boolean;

IMPLEMENTATION
function log(arg,base:real):real;
begin
  if (arg>0) and (base>0) then log:=ln(arg)/ln(base)
  else log:=0;
end;

function expbase(arg,base:real):real;
begin
  if (base>0) then expbase:=exp(arg*ln(base))
  else expbase:=0;
end;

function uptrunc(arg:real):longint;
var sinal,code:integer;
    result:longint;
    aux:string;
begin
  if (arg<0) then sinal:=-1 else sinal:=1;
  arg:=arg*sinal;
  str(arg:15:0,aux);
  val(aux,result,code);
  inc(result);
  uptrunc:=result*sinal;
end;

function pot(arg,base:real):real;
begin
  if (arg>0) then pot:=exp(ln(arg)*base)
  else if (arg<0) then pot:=-exp(ln(-arg)*base)
  else if (arg=0) then pot:=0;
end;

function potencia(arg:real;base:integer):real;
var aux:real;
    i:word;
begin
  aux:=1;
  if (base>0) then for i:=1 to base do aux:=aux*arg;
  if (base<0) then begin
    for i:=1 to -base do aux:=aux*arg;
    aux:=1/aux;
  end;
  potencia:=aux;
end;

function raizN(arg,base:real):real;
var aux:real;
    negarg:boolean;
begin

```



```

raizN:=0;
calculomtmok:=true;
if (base=0) then calculomtmok:=false;
if ((arg<0) and (base=int(base)) and (not
odd(trunc(base)))) then calculomtmok:=false;
if ((base<0) and (arg=0)) then calculomtmok:=false;
if (not calculomtmok) then exit;
if (arg<0) then begin
  negarg:=true;
  arg:=-arg;
end
else negarg:=false;
if (base<0) then begin
  arg:=1/arg;
  base:=-base;
end;
if (arg=0) then aux:=0
else begin
  aux:=(ln(arg))/base;
  aux:=exp(aux);
end;
if (negarg) then raizn:=-aux else raizn:=aux;
end;

```

```

function bascara(ca,cb,cc:real;var r1,r2:real):boolean;
var delta:real;
    raizreal:boolean;
begin
  delta:=cb*cb;
  delta:=delta - 4*ca*cc;
  if (delta>=0) then raizreal:=true
  else begin
    raizreal:=false;
    delta:=-delta;
  end;
  delta:=sqrt(delta);
  r1:=(-cb + delta) / (2*ca);
  r2:=(-cb - delta) / (2*ca);
  bascara:=raizreal;
end;

```

```

function polinomio(ptox:real;ordem:word;coef:array of
real):real;
var i:word;
    aux:real;
begin
  aux:=0;
  for i:=0 to ordem do begin
    aux:=aux+potencia(ptox,i)*coef[i];
  end;
  polinomio:=aux;
end;
end.

```

APÊNDICE V

MANUAL DO TENSÍOMETRO

OBSERVAÇÕES

1. As rotinas que envolvem qualquer tipo de experimento no qual se utilize a bomba de seringa somente poderão ser acessadas em computadores dotados da placa MQ112/8pcc, sob o risco de ser "derrubado" o sistema operacional.
2. O programa deve ser rodado a partir do DOS 3.0 ou superior. Rodar o programa em janela do MS Windows não é aconselhado, principalmente se a bomba de seringa for movimentada.
3. Os parâmetros que devem ser ajustados variam em função da rotina que será realizada, de tal forma que nem todos os parâmetros existentes precisarão ser alterados para que se possa realizar uma determinada rotina.
4. A rotina "REGR. DENS" (regressão de densidade em função da concentração) somente deverá ser realizada nos casos em que houver mudanças na densidade da solução em função da concentração. Em situação contrária os valores para os parâmetros 13, 14 e 15, ou seja COEF. DE GRAU 0, .. GRAU 1 e .. GRAU 2 serão normalmente 1, 0 e 0, respectivamente.
5. As seringas de 20 mL não devem conter mais que 10 mL de solução, pois que utilizando volumes maiores é grande a possibilidade de a seringa ficar trancada.
6. Para os procedimentos experimentais, nos quais não há variação de concentração, pode-se eliminar a câmara de mistura e fazer a conexão direta da seringa ao capilar.

MEDIDAS DE TENSÃO SUPERFICIAL EM FUNÇÃO DA CONCENTRAÇÃO

Essa é a principal rotina do aparelho e provavelmente a que mais será utilizada. Os procedimentos preliminares para realizar essa rotina são a calibração do raio do capilar e a regressão da densidade.

A sequência de opções a ser seguida, a partir do MENU PRINCIPAL é:

1. Teclar [F1] ENSAIO.
2. Teclar [F1] CONCENTRAÇÃO VARIADA EXPONENCIAL.
3. Fazer a conexão dos tubos condutores (pag. 2).
4. Ajustar os valores para todos os parâmetros, exceto os parâmetros 6, 9 e 16 (pag.4).
5. Digitar uma sequência de dados que caracterize o experimento (nome do experimento).

6. Digitar um nome de arquivo (até oito caracteres) válidos para o DOS, sem extensão (a extensão utilizada é .tns).
7. Teclar [ENTER] para iniciar o experimento.
8. O término do experimento é alcançado após o fim de curso da bomba de seringa, após falha de comunicação com a balança ou após ser teclado [ESC].

CONEXÃO DOS TUBOS CONDUTORES

O sistema de fluxo deve ser adaptado na seguinte sequência:

1. Inserir as respectivas soluções na seringa e na câmara de mistura (o volume da solução inserida na câmara de mistura deve ser meticulosamente medido);
2. Posicionar adequadamente a bomba de seringa (pag.2);
3. Conectar a saída da seringa à entrada da câmara de mistura;
4. Injetar no tubo condutor que une a câmara de mistura ao capilar de vidro, por meio de uma seringa auxiliar, a mesma solução que foi utilizada para preencher a câmara de mistura. O volume injetado e a vazão devem ser o suficiente para garantir a expulsão de todas as bolhas de ar que se formam inicialmente no tubo condutor e abaixo do capilar.
5. Esvaziar todo o tubo condutor, deixando a solução fluir por gravidade. Eliminar também a solução que fica aderida à parte inferior do capilar.
6. Conectar a saída da câmara de mistura à entrada do capilar.
7. Conectar todo o sistema de termostatização.

POSICIONAMENTO ADEQUADO DA BOMBA DE SERINGA

Para a realização adequada do posicionamento da seringa é necessário primeiro o preenchimento da seringa com a solução que será utilizada no experimento, excetuando-se é claro, para os casos onde não houver injeção de fluidos pela seringa. Em seguida deve-se prender a seringa na bomba. O orifício central existente no prendedor da bomba é utilizado para a introdução do bico da camisa de termostatização (saída). A bomba de seringa é então movimentada, utilizando a sequência de instruções "MOVIMENTANDO A BOMBA DE SERINGA" (pag.3), até que haja o contato físico entre o êmbolo da seringa e o carrinho de movimentação da bomba. Na realidade o carrinho de movimentação da bomba deve movimentar o êmbolo da seringa por alguns milímetros de forma a eliminar todas as bolhas de ar que possam existir no tubo condutor de Teflon.

MOVIMENTANDO A BOMBA DE SERINGA

Essa rotina é realizada anteriormente a qualquer rotina que envolva a movimentação da bomba de seringa (a maioria das rotinas realizam tal procedimento). A ausência da placa MQ12/8pcc acarretará o travamento do programa. Nessa rotina, o sensor ótico de gotas deve estar conectado e

desobstruído. A inobservância desses dois itens ocasionará o cancelamento imediato da rotina. A função de cada tecla para o ajuste das bombas está descrito abaixo. Inicialmente uma descrição dos comandos e termos utilizados.

[LE] = SETA À ESQUERDA
[RI] = SETA À DIREITA
[UP] = SETA PARA CIMA
[DN] = SETA PARA BAIXO
inverte seleção [ANDA/PARA] = Inverte a seleção entre andar e parar. Assim, se estiver parada fará seleção de ANDAR e se estiver andando fará seleção de PARAR.

[F1] - Inverte a seleção ANDA/PARA da bomba 1.
[F2] - Inverte a seleção ANDA/PARA da bomba 2.
[LE] - Inverte seleção do movimento FRETE/TRÁS para a bomba 1.
[RI] - Inverte seleção do movimento FRENTE/TRÁS para a bomba 2.
[UP] - Coloca as duas bombas em movimento para trás.
[DN] - Coloca as duas bombas em movimento para frente.
[ESPAÇO] - Interrompe imediatamente o movimento de ambas as bombas.
[+] - Aumenta a velocidade da bomba, até um valor limite.
[-] - Diminui a velocidade da bomba, até um valor limite.
[ENTER] - Prossegue a rotina para a próxima etapa.
[ESC] - Aborta a rotina previamente selecionada.

Ps. A mudança na velocidade da bomba de seringa é muito útil para o ajuste de posição, pois que boa parte do movimento pode ser realizado em alta velocidade.

ALTERAÇÃO DE PARÂMETROS

Algumas considerações devem ser feitas inicialmente:

1. O parâmetro "ACELERAÇÃO DA GRAVIDADE" somente será modificado quando o instrumento mudar de cidade ou altitude. Em situação contrária, esse valor será sempre constante (normalmente 980,7).
2. O parâmetro "PASSO DA BOMBA" somente deverá ser alterado quando a bomba de seringa for substituída por outra de outro modelo. Esse parâmetro é de pouca importância pois quase todas as rotinas não necessitam desse valor.
3. O parâmetro "VELOC. BOMBA MODO RÁPIDO" somente é utilizado na rotina "FLUXO VARIADO" do MENU DE ENSAIO, e devido a isso também é de pouca importância. Esse parâmetro indica a velocidade da bomba de seringa logo após a queda de uma gota. Decorrido cerca de 60% de formação da gota subsequente, a velocidade da bomba retorna ao valor original (normal).
4. Se o experimento a ser realizado envolver mudanças na densidade da solução, os coeficientes da regressão devem ser calculados digitando-se [F10].

Uma breve descrição de todos os parâmetros disponíveis para alteração será realizada a seguir.

- 1 - **CONCENTRAÇÃO DA SERINGA;**
É a concentração da solução da seringa.
- 2 - **CONCENTRAÇÃO DO MISTURADOR;**
É a concentração inicial da solução da câmara de mistura.
- 3 - **VOLUME DO MISTURADOR;**
É o volume total da solução inserida na câmara de mistura.
- 4 - **DENSIDADE DA SERINGA;**
É a densidade da solução contida na seringa (normalmente = 1).
- 5 - **DENSIDADE DO MISTURADOR;**
É a densidade inicial da solução do misturador (normalmente=1).
- 6 - **PASSO DA BOMBA;**
É dependente de cada bomba de seringa utilizada. A sua calibração é realizada por rotina própria. Esse parâmetro somente é utilizado na rotina "CONCENTRAÇÃO VARIADA DUAS BOMBAS" e "INCLUI SERINGA".
- 7 - **RAIO EFETIVO DO CAPILAR;**
É determinado automaticamente em uma rotina de calibração.
- 8 - **VELOCIDADE DA BOMBA;**
É o fluxo da bomba de seringa. Para uma seringa de 20 mL, e bomba MQBSGI/302, $1\text{ passo/s} = 0,0600\text{ mL}\cdot\text{min}^{-1}$.
- 9 - **QTDDE GOTAS PARA CALIBRAR;**
É a quantidade de gotas que será lida nos procedimentos de calibração antes que a rotina termine de forma automática.
- 10 - **TEMPERATURA EXPERIMENTAL;**
Temperatura do experimento, em Kelvin.
- 11 - **N MOLES TOTAL/N MOLES SURF;**
É o parâmetro x da equação de excesso. Ver Nunez-Tolin, V.; Hoebregs, H.; Léonis, J.; Paredes, S., Surface tension measurements by the drop-weight method for continuously varying surfactant concentration., J. Coll. Interf. Sci., v.85, n.2, p.597-600, (1982).
- 12 - **ACELERAÇÃO DA GRAVIDADE;**
É a aceleração da gravidade local (normalmente = 981.0).
- 13 - **COEFICIENTE DE GRAU 0;**
É o coeficiente de grau 0 da regressão polinomial de ordem 2 para os valores de densidade e concentração (normalmente = 1). Ver REGRESSÃO DE DENSIDADE
- 14 - **COEFICIENTE DE GRAU 1;**
Ver item 13. Normalmente = 0.
- 15 - **COEFICIENTE DE GRAU 2;**
Ver item 13. Normalmente = 0.
- 16 - **VELOC. BOMBA MODO RÁPIDO;**
É a velocidade da bomba de seringa que será utilizada no período entre a queda de uma gota e cerca de 60% da formação da gota subsequente. Esse parâmetro somente tem validade na rotina "FLUXO VARIADO".

REGRESSÃO DE DENSIDADE X CONCENTRAÇÃO

Deve ser inserida uma tabela de até 65 dados de densidade e concentração da solução. A determinação desses valores deve ser feita por instrumentação própria, não competindo a esse aparelho a veracidade dos valores utilizados.

1. A tecla [F10] é utilizada para alteração entre a coluna densidade e a coluna concentração.
2. A tecla [F3] é utilizada para realizar a regressão polinomial de ordem 2 e para salvamento automático dos coeficientes.
3. A tecla [ESC] é utilizada para abortar a rotina e retornar aos parâmetros.

ANÁLISE DE DADOS

É a 2ª rotina na ordem de importância. A seguinte sequência deve ser realizada, a partir do menu principal.

1. Teclar [F4] ANÁLISE DE DADOS;
2. Teclar [F1] CONCENTRAÇÃO VARIADA;
3. Digitar nome do arquivo de dado que se deseja ler;
4. A movimentação do cursor central é obtida utilizando-se as teclas de setas do teclado não numérico;
5. A barra de espaços é utilizada para alterar o tipo de movimentação do cursor e de qualquer uma das 5 retas disponíveis. A movimentação pode ser automática (necessitando um único toque do teclado) ou não automática (a movimentação é obtida enquanto a tecla estiver pressionada).
6. A inserção e seleção de uma reta é feita digitando-se as teclas de F1 a F5. A movimentação da reta selecionada é feita digitando-se as setas do teclado numérico (com NUM LOCK selecionado). A inclinação da reta selecionada é obtida digitando-se as teclas *, /, Insert e Delete, da seguinte forma:
 - * - Aumenta a inclinação da reta, mantendo constante a coordenada X;
 - / - Diminui a inclinação da reta, mantendo constante a coordenada X;
 - Delete - Aumenta a inclinação da reta, mantendo constante a coordenada Y;
 - Insert - Diminui a inclinação da reta, mantendo constante a coordenada Y.
7. As teclas + e - são utilizadas para alterar o valor do incremento utilizado para movimentar o cursor e movimentar e inclinar as retas.
8. Após ajustadas as retas é possível obter o valor de vários parâmetros físico-químicos, digitando-se TAB e em seguida as teclas de função (F1 a F5) correspondentes às retas anterior e posterior à CMC, respectivamente. Teclando-se [ESC] retorna-se à edição dos dados.
9. Para obter-se a regressão dos dados (construção da isoterma de adsorção) é necessário colocar o cursor no ponto final para o cálculo da regressão

(normalmente muito próximo à CMC). Em seguida tecla-se [F10] e então escolhe-se a ordem da regressão (de 1 a 9). A melhor curva é traçada e o salvamento da isoterma é realizado teclando-se [ENTER] e em seguida o nome do arquivo de dados (a extensão é .exc).

10. Um novo arquivo pode ser editado, teclando-se [F7] e retornando ao item 3.

CALIBRAÇÃO DO RAIOS EFETIVO

Em princípio essa rotina deve ser realizada a cada dia, mas sobretudo quando o capilar é substituído, quando a superfície do capilar é modificada (lixada), quando o aparelho permanece desativado por um longo intervalo de tempo, quando o adaptador de acrílico do capilar é removido do local original, ou sempre que transcorrer um longo intervalo de tempo desde a última calibração. Normalmente a solução utilizada para a calibração do raio efetivo é água pura e o fluxo escolhido deve ser bastante baixo (parâmetro 8).

A sequência de opções a ser seguida, a partir do MENU PRINCIPAL é:

1. Ajustar os valores para os parâmetros 4, 8, 9, 10 e 12 (os demais parâmetros são irrelevantes).
2. Teclar [F3] DETERMINAR RAIOS EFETIVO.
3. Fazer a conexão dos tubos condutores (não é necessário utilizar a câmara de mistura).
4. Digitar a tensão superficial da solução utilizada, respeitando a temperatura experimental. O programa sugere o valor da tensão superficial da água na temperatura selecionada. Porém, se outra solução for utilizada em substituição à água, basta digitar o valor correto da tensão superficial.
6. Aceitar ou não o raio do capilar calculado pelo programa.

CONFERE DETECTOR

Escolhendo [F7] a partir do menu principal é realizada a rotina de verificação do detector ótico de gotas, em 3 etapas.

1. O detector deve permanecer desobstruído e então tecla-se [ENTER]. Serão realizadas 3.000 leituras e o resultado é apresentado em seguida.
2. O detector deve permanecer obstruído e então tecla-se [ENTER]. Serão realizadas 3.000 leituras e o resultado é apresentado em seguida.
3. É realizada a contínua leitura do detector e apresentação do resultado. Essa etapa somente será concluída após teclar-se [ESC].