

Universidade Federal do Rio de Janeiro  
Escola Politécnica  
Departamento de Eletrônica e de Computação

***Monitoramento da Destilação Fracionada de Petróleo via  
Processamento Digital de Imagens***

Autor:

---

Flávio Pacheco Reis

Orientador:

---

Prof. José Gabriel Rodriguez Carneiro Gomes, Ph.D.

Examinador:

---

Prof. Antônio Cláudio Gómez de Sousa, M.Sc.

Examinador:

---

Prof. Heraldo Luis Silveira de Almeida, D.Sc.

DEL

Março de 2009

## **DEDICATÓRIA**

Dedico este trabalho ao povo brasileiro que contribuiu de forma significativa à minha formação e estada nesta Universidade. Este projeto é uma pequena forma de retribuir o investimento e confiança que me foram depositados.

## **AGRADECIMENTO**

Aos meus pais, meus mestres, meus amigos de universidade e a todos aqueles que estiveram ao meu lado me apoando direta ou indiretamente.

Agradeço em especial ao Engenheiro Eletrônico Henrique Oscar Duran Lira pela enorme contribuição por seus conhecimentos de Processamento Digital de Imagens, pois sem estes meu trabalho levaria muito mais tempo para ser concluído.

## **RESUMO**

O presente projeto trata do desenvolvimento de um sistema que visa monitorar o processo de destilação fracionada de petróleo fazendo uso de técnicas de processamento digital de imagens. Para um planejamento e desenvolvimento o mais profissional possível, fiz o uso dos conhecimentos adquiridos em Engenharia de Software: nesta cadeira aprendi como devem ser todas as etapas do desenvolvimento de um projeto bem como as documentações e normas a serem seguidas.

O objetivo geral é, então, a construção de um sistema de tempo real de detecção do desprendimento da primeira gota (que indica o início do processo de destilação de petróleo), além do acompanhamento do volume depositado na garrafa coletora, em um processo de destilação de petróleo em batelada baseado em técnicas de processamento digital de imagens. Os métodos aplicados incluem a utilização de técnicas de processamento digital de imagens, para processar as imagens adquiridas pelas câmeras de vídeo; teoria de circuitos para a elaboração dos circuitos utilizados no projeto; técnicas de CAD para a simulação dos circuitos, além da geração de esquemáticos; técnicas de Orientação a Objetos e Linguagens de Programação, que corresponde à codificação em uma linguagem de programação específica, que neste caso foi C++. Os resultados finais foram satisfatórios, tendo em vista a robustez do sistema, que durante o processo de teste de desempenho apresentou uma alta taxa de acerto.

Mesmo já havendo outras tecnologias para tal monitoramento, não foi encontrada nenhuma que fosse baseada em processamento digital de imagens, o projeto também apresenta um baixo custo se comparado com os já existentes, é inovador e demonstra a capacidade de desenvolvimento e inovação do profissional de Engenharia Eletrônica formado pela Universidade Federal do Rio de Janeiro.

**Palavras-Chave:** processamento de imagens, monitoração de processos, controle de processos, sistemas de tempo real.

## **ABSTRACT**

This project is about the development of a system to monitor oil distill using image processing techniques. For a professional planning and development, I have used knowledge acquired in Software Engineering: in this course I have learned the steps for project development, the needed documentation and rules to be followed.

The general objective is then to build a system for real-time detection of the detachment of the first drop (which indicates the beginning of the distillation of petroleum), in addition to monitoring the amount deposited in the bottle collector, in a process of distillation oil-in batch processing techniques based on digital images. The methods include the use of techniques of digital image processing, to process the images acquired by video cameras; theory of circuits for the preparation of the circuits used in the design, CAD techniques for the simulation of circuits, in addition to the generation of schematic; technical guidance to Objects and programming languages, which is the coding in a programming language specific, which in this case was C++. The final results were satisfactory in view of the robustness of the system, which during the test performance showed a high rate of success.

Even though there are others techniques to oil monitoring, I have not found out one that was based on image processing, not to mention low-cost: the system developed in this project is cheap compared to others. It is novel and it shows the development and innovation capabilities of Electronics Engineers graduating from the Federal University of Rio de Janeiro.

Keywords: image processing, process monitoring, process control, real-time systems.

## SIGLAS

UFRJ – Universidade Federal do Rio de Janeiro.

CAT – Circuito Adaptador de Tensão.

CA – Circuito Atuador.

CLP – Controlador Lógico Programável.

LPT – *Line Print Terminal* (Porta paralela).

MTH – Módulo de Teste de Hardware.

CT – Circuito de Teste.

OPC – *OLE for Process Control*.

OLE – *Object Linking and Embedding*.

VNC – *Virtual Network Computing*

# Sumário

<b>1. O PROJETO DE MONITORAMENTO.....</b>	<b>ERRO! INDICADOR NÃO DEFINIDO.</b>
1.1. TEMA .....	1
1.2. DELIMITAÇÃO.....	1
1.3. JUSTIFICATIVA .....	1
1.4. OBJETIVOS.....	2
1.5. METODOLOGIA .....	3
1.6. DESCRIÇÃO.....	4
1.7. COMPONENTES DO SISTEMA .....	4
1.8. FUNCIONAMENTO GERAL DO SISTEMA.....	5
<b>2. O SISTEMA DE MONITORAMENTO.....</b>	<b>8</b>
2.1. DIAGRAMA DE CASOS DE USO DO SISTEMA DE MONITORAMENTO.....	8
2.2. MÓDULO DE DETECÇÃO DA QUEDA DA 1 <sup>a</sup> GOTA .....	9
2.2.1. Caso de Uso Módulo Gota.....	9
2.2.2. Diagrama Físico do Módulo Gota.....	11
2.2.3. Classe do Módulo Gota.....	12
2.2.4. Diagrama de Atividades do Módulo Gota.....	14
2.2.5. Ambiente de Simulação .....	14
2.2.6. Calibragem do Sistema Gota.....	16
2.2.7. Procedimento de Detecção.....	18
2.2.8. Resultados dos Testes de Desempenho.....	19
2.3. MÓDULO DE LEITURA E IDENTIFICAÇÃO DO CÓDIGO DE BARRAS .....	24
2.3.1. Caso de Uso Módulo Código de Barras.....	25
2.3.2. Classe do Módulo Código de Barras.....	25
2.3.3. Diagrama de Atividades do Módulo Código de Barras .....	28
2.3.4. Garrafa Coletora .....	28
2.3.5. Padrão de Codificação .....	29
2.3.6. Testes de Desempenho .....	30
2.4. MÓDULOS AUXILIARES .....	33
2.4.1. Detecção da Gota .....	33
2.5. MÓDULOS DE HARDWARE .....	34
2.5.1. Circuito Atuador (CA).....	34
2.5.2. Módulo de Teste de Hardware (MTH).....	37
2.5.2.1. Classe do Módulo de Teste de Hardware .....	37
2.5.2.2. Sub-Módulo Teste CA.....	39
2.5.2.3. Sub-Módulo Teste CAT.....	40
2.5.2.4. Expressões Lógicas das Saídas do CLP .....	41

2.5.2.5. Blocos Lógicos das Saídas do CLP .....	41
<b>3. O PROCESSAMENTO DE IMAGEM .....</b>	<b>43</b>
3.1. FUNDAMENTOS PARA PROCESSAMENTO DIGITAL DE IMAGENS.....	43
3.1.1. <i>Imagen em Escala de Cinza</i> .....	43
3.1.2. <i>Definição Matemática de Imagem</i> .....	44
3.1.3. <i>Imagen Digital</i> .....	44
3.1.4. <i>Representação de Imagens</i> .....	44
3.1.5. <i>Propriedades de um Pixel</i> .....	46
3.2. ETAPAS PARA O PROCESSAMENTO DIGITAL DE IMAGENS .....	48
3.2.1. <i>Aquisição de Imagens</i> .....	48
3.2.2. <i>Realce de Imagem</i> .....	49
3.2.3. <i>Segmentação de Imagens</i> .....	55
3.2.4. <i>Classificação de Imagem</i> .....	56
3.3. PROCESSAMENTO DIGITAL DE IMAGEM E O PROJETO .....	57
3.3.1. <i>O Algoritmo</i> .....	58
3.3.2. <i>A Detecção da Gota</i> .....	59
3.3.3. <i>Etapas do Projeto</i> .....	59
3.3.3.1. Aquisição da Imagem Real .....	59
3.3.3.2. Imagem em Níveis de Cinza .....	60
3.3.3.3. Imagem Binarizada (Limiarização ou <i>Thresholding</i> ).....	61
<b>4. ELABORAÇÃO DOS CIRCUITOS .....</b>	<b>62</b>
4.1. CIRCUITO TEÓRICO .....	62
4.2. CIRCUITO PRÁTICO .....	63
A CAIXA DA FIGURA 4.6 ABAIXO ABRIGA O CIRCUITO ATUADOR. ESTE CIRCUITO É RESPONSÁVEL PELO ACIONAMENTO DAS CÂMERAS E ILUMINAÇÕES DO SISTEMA. O DIAGRAMA ESQUEMÁTICO DESTE CIRCUITO ENCONTRA-SE NA FIGURA A.1 DO APÊNDICE A.....	66
<b>5. CONCLUSÃO .....</b>	<b>69</b>
<b>APÊNDICE A.....</b>	<b>70</b>
A.1. DIAGRAMA DO CIRCUITO ATUADOR.....	70
A.2. DIAGRAMA DO CIRCUITO ATUADOR.....	71

# **Lista de Figuras**

Figura 1.1: Visão Geral do Sistema de Monitoramento.....	5
Figura 2.1: Modulo Controlador .....	7
Figura 2.2: Diagrama de Casos de Uso do Sistema de Monitoramento .....	8
Figura 2.3: Diagrama de Contexto Módulo Gota .....	11
Figura 2.4: Diagrama de Atividades do Módulo Gota.....	14
Figura 2.5: Esquema de Montagem do Módulo Gota.....	14
Figura 2.6: Tomada do Bico Mostrando o Retângulo da Área de Busca.....	15
Figura 2.7: Módulo de Calibragem.....	16
Figura 2.8: Calibragem para Definição da Área de Busca.....	17
Figura 2.9: Altura Relevante da Área de Busca .....	17
Figura 2.10: Definição da Área de <i>offset</i> .....	18
Figura 2.11: <i>Offset</i> Contendo uma Gota .....	18
Figura 2.12: Tomada Lateral .....	18
Figura 2.13: Tomada Frontal.....	18
Figura 2.14: Tomada Lateral .....	18
Figura 2.15: Tomada Frontal.....	18
Figura 2.16: Vista Superior da Montagem .....	19
Figura 2.17: Diagrama de Atividades do Módulo Código de Barras .....	28
Figura 2.18: Garrafa Coletora.....	28
Figura 2.19: Modelo de Código de Barras do Tipo Intercalado 2 de 5.....	29
Figura 2.20: Código de Barras Intercalado 2 de 5 Invertido .....	29
Figura 2.21: Tomada de Fotos com o Eixo Focal Alinhado ao Centro da Garrafa.....	30
Figura 2.22: Tomada de Fotos com o Eixo Focal Alinhado ao Fundo.....	31

Figura 2.23: Tomada de Fotos com a Câmera Inclinada Enquadrando toda a Garrafa .	31
Figura 2.24: Programa de Testes para Identificação do Código de Barras.....	32
Figura 2.25: Programa Gerador de Experimentos .....	34
Figura 2.26: Diagrama de Blocos do Circuito CA .....	35
Figura 2.27: Diagrama Esquemático de entrada/saída.....	35
Figura 2.28: Protocolo de Sinalização com o CLP.....	36
Figura 2.29: Diagrama de Blocos do CA com MTH_CA Incorporado .....	39
Figura 2.30: Blocos Lógicos do CLP.....	41
Figura 2.31: Diagrama de Contatos do CLP .....	42
Figura 3.1: Conjunto E.....	43
Figura 3.2: Produto Cartesiano E x K.....	43
Figura 3.3: Imagem em Níveis de Cinza.....	44
Figura 3.4: Representação Vetorial [5].....	45
Figura 3.5: Representação Matricial [5] .....	45
Figura 3.6: Vizinhança 4 de um <i>Pixel</i> .....	46
Figura 3.7: Vizinhança 8 de um <i>Pixel</i> .....	46
Figura 3.8: Estrutura do Arquivo <i>PNG</i> [6].....	47
Figura 3.9: Estrutura do Arquivo <i>Bitmap</i> [6] .....	48
Figura 3.10: Estrutura do Arquivo <i>TIFF</i> [6] .....	48
Figura 3.11: Câmera Digital.....	49
Figura 3.12: Histograma de Imagens [8].....	50
Figura 3.13: Função Linear .....	51
Figura 3.14: Função Raiz Quadrada .....	51
Figura 3.15: Função Logaritmica .....	52
Figura 3.16: Função Inversa.....	52

Figura 3.17: Máscara de Roberts .....	53
Figura 3.18: Máscara de Sobel .....	54
Figura 3.19: Máscara da Média .....	54
Figura 3.20: Bico Real .....	59
Figura 3.21: Garrafa Real.....	59
Figura 3.22: Bico em Tom de Cinza.....	60
Figura 3.23: Garrafa em Tom de Cinza .....	60
Figura 3.24: Bico Binarizado .....	61
Figura 3.25: Garrafa Binarizada .....	61
Figura 4.1: Circuito de Linhas Pares.....	62
Figura 4.2: Circuito de Linhas Ímpares .....	62
Figura 4.3: Circuito de Linhas Pares.....	63
Figura 4.4: Circuito de linhas Ímpares.....	64
Figura 4.5: Painel de Leds.....	65
Figura 4.6: Circuito Atuador .....	66
Figura 4.7: Circuito de Teste Vista Superior.....	66
Figura 4.8: Circuito de Teste Vista Traseira .....	67
Figura 4.9: Circuito de Teste Vista Frontal.....	67
Figura 4.10: Circuito Teste Vista Lateral.....	68
Figura A.1: Diagrama Esquemático do CA .....	70
Figura A.2: Circuito CA Simulado no ORCAD.....	71
Figura A.3: Circuito de Teste .....	72

# **Lista de Tabelas**

Tabela 1: Ensaios 1 e 2.....	19
Tabela 2: Ensaio 3.....	20
Tabela 3: Ensaio 4.....	21
Tabela 4: Ensaio 5.....	21
Tabela 5: Ensaio 6.....	22
Tabela 6: Ensaio 7.....	22
Tabela 7: Ensaio 8.....	22
Tabela 8: Ensaio 9.....	23
Tabela 9: Valores dos Parâmetros de Montagem .....	23
Tabela 10: Relação entre Contraste e Limiar do <i>offset</i> .....	24
Tabela 11: Descrição da Pinagem da LPT com o CA .....	36
Tabela 12: Descrição da Pinagem da LPT com o CA e MTH_CA .....	40
Tabela 13: Tabela-Verdade das Saídas do CLP .....	41

# **Capítulo 1**

## **Introdução**

### **1.1. Tema**

O trabalho trata do desenvolvimento de um sistema visando a monitoração automatizada, via processamento digital de imagens, do processo de destilação fracionada do petróleo. No desenvolvimento deste trabalho serão usados conhecimentos de Eletrônica Analógica, Eletrônica Digital, Engenharia de Software, Processamento Digital de Imagens e Linguagens de Programação.

### **1.2. Delimitação**

O projeto que foi desenvolvido envolve o monitoramento da destilação fracionada do petróleo, usando como técnica o processamento digital de imagem. A demanda por este produto é de empresas e indústrias do segmento petrolífero. Como será explicado adiante, existem diferentes métodos de monitoramento da destilação fracionada do petróleo, mas ou são caros, ou imprecisos e de difícil calibragem. Já o sistema baseado em processamento de imagem requer apenas um computador, câmeras, iluminações e alguns sistemas eletrônicos auxiliares, todos de baixo custo quando comparados com os equipamentos já utilizados para tal monitoramento. Outra característica importante é o bom índice de acerto apresentado pelo sistema baseado em processamento digital de imagem.

### **1.3. Justificativa**

Industrialmente, o petróleo é destilado em grandes quantidades, produzindo frações conhecidas como diesel, gasolina, nafta, etc. Cada uma destas frações tem uma faixa de temperatura característica e propriedades físicas e químicas padronizadas. No entanto, um dado petróleo, dependendo de sua origem geoquímica e do seu processo de formação, não necessariamente produz todas as frações ou não as produz nos valores padronizados das propriedades de cada fração. Sendo assim, uma mistura de petróleo é composta, acrescida de outras substâncias produzidas na refinaria, de forma que as

diferentes frações sejam produzidas de acordo com as especificações e em volume adequado à demanda do mercado (ASTM, 2002).

Para determinar em laboratório os diferentes cortes ou frações de subprodutos existentes em uma determinada amostra, bem como o volume que cada uma das faixas produz, utiliza-se a destilação em batelada, sendo um dos métodos o padrão ASTM D5236 (ref. ASTM, 2002). Esta determinação é vital não só para a destilação futura do mesmo óleo executada em regime de produção em uma refinaria, mas também na formação de seu preço, pois o valor do petróleo é função do volume produzido de suas frações mais nobres.

O processo de destilação em batelada já possui uma monitoração automatizada, todavia os instrumentos utilizados apresentam certas dificuldades, tais como: pouca robustez; pouca intercambialidade e dificuldade de operação. Surge daí a proposta da monitoração através de técnicas de processamento de imagens, pois estas apresentam vantagens (Seção 1.4) em relação às técnicas correntes.

#### **1.4. Objetivos**

No processo de destilação em batelada ao qual este projeto se aplica, as diferentes frações do petróleo são separadas em frascos localizados em um carrossel na parte inferior da unidade de destilação utilizada. As frações são coletadas em intervalos determinados a partir do instante em que se desprende a primeira gota, indicando que o processo de destilação teve início. A cada intervalo, que corresponde a encher o frasco coletor parcialmente (quando um subproduto termina) ou totalmente, o carrossel de troca de frasco sofre um movimento de rotação, trocando o frasco recipiente do petróleo destilado.

Além disto, é preciso determinar a temperatura do início da destilação, ou seja, a temperatura mínima da primeira faixa ou fração. Como a retirada é contínua e pela própria dinâmica do processo, o início da destilação corresponde ao instante da formação da primeira gota no condensador lateral (ASTM, 2002).

Com a instrumentação atual, a identificação da formação da primeira gota apresenta algumas dificuldades em termos de instrumentação. Os dispositivos existentes se baseiam na interrupção de um feixe de luz, e apresentam três grandes inconvenientes:

- a) são específicos para cada unidade (eletrônica dedicada).
- b) a regulagem do feixe ao caminho da gota é difícil e instável, pois a queda da gota é sujeita a perturbação.

c) os instrumentos não apresentam recursos que facilitem a regulagem.

Note-se que a perda da queda da primeira gota equivale à perda de todo o processo de destilação até aquele instante e que um erro na determinação da gota pode comprometer todo o trabalho de medida.

O objetivo principal ao qual este projeto se propõe é: criação de um sistema automático empregando técnicas de processamento de imagens para realizar a detecção da queda da primeira gota, detecção do código de barras que virá na parte inferior da garrafa, e medição do menisco (nível de líquido na garrafa coletora).

O referido projeto apresenta as seguintes vantagens frente às técnicas atuais:

a) uso de instrumentação padrão de mercado (câmera e computador), sendo as diferenças entre os equipamentos de destilação tratadas por programa.

b) a regulagem é muito mais fácil, não sendo necessário o ajuste do equipamento ao caminho da gota, podendo o operador utilizar a própria imagem gerada pelo dispositivo para enquadrar/regular o posicionamento da câmera.

## **1.5. Metodologia**

Para o desenvolvimento do projeto busquei como metodologia inicial o uso de técnicas aprendidas na cadeira de Engenharia de Software. Sendo assim montei as seguintes documentações:

- a) Plano de Gerenciamento de Projeto de Software;
- b) Especificação de Requisitos de Software.

No primeiro documento foi feito o levantamento de itens tais como finalidade, escopo, objetivo, cronograma, orçamento, papéis e responsabilidades, plano para aquisição e alocação de recursos, plano de relatórios, plano para treinamento etc.

No segundo foi feito o levantamento de itens tais como finalidade, escopo, perspectivas do produto, funções do produto, restrições, pressupostos e dependências, interfaces, requisitos funcionais e não funcionais.

Também lancei mão da reutilização de componentes, artefatos de software, hardware e documentações produzidas em projetos anteriores, pois estes já haviam sido testados, usados, validados e apresentavam as características necessárias ao nosso projeto.

Com o auxílio da documentação produzida e da reutilização de componentes e artefatos foi possível a economia de tempo, dinheiro e a minimização dos erros encontrados ao longo do processo de desenvolvimento do sistema.

## 1.6. Descrição

No Capítulo 2 será apresentada uma explicação e decomposição detalhada do projeto. Sua forma de apresentação é uma composição dos documentos: Plano de Gerenciamento de Projeto de Software, Especificação de Requisitos de Software e outras informações que julgamos necessárias.

No Capítulo 3 serão apresentados os fundamentos do processamento digital de imagens bem como as técnicas utilizadas no projeto e as imagens resultantes.

No Capítulo 4 serão apresentados os esquemáticos de alguns circuitos do projeto bem como fotos do sistema montado em laboratório.

No Apêndice A serão apresentados o diagrama do Circuito Adaptador de Tensão, Circuito de Teste.

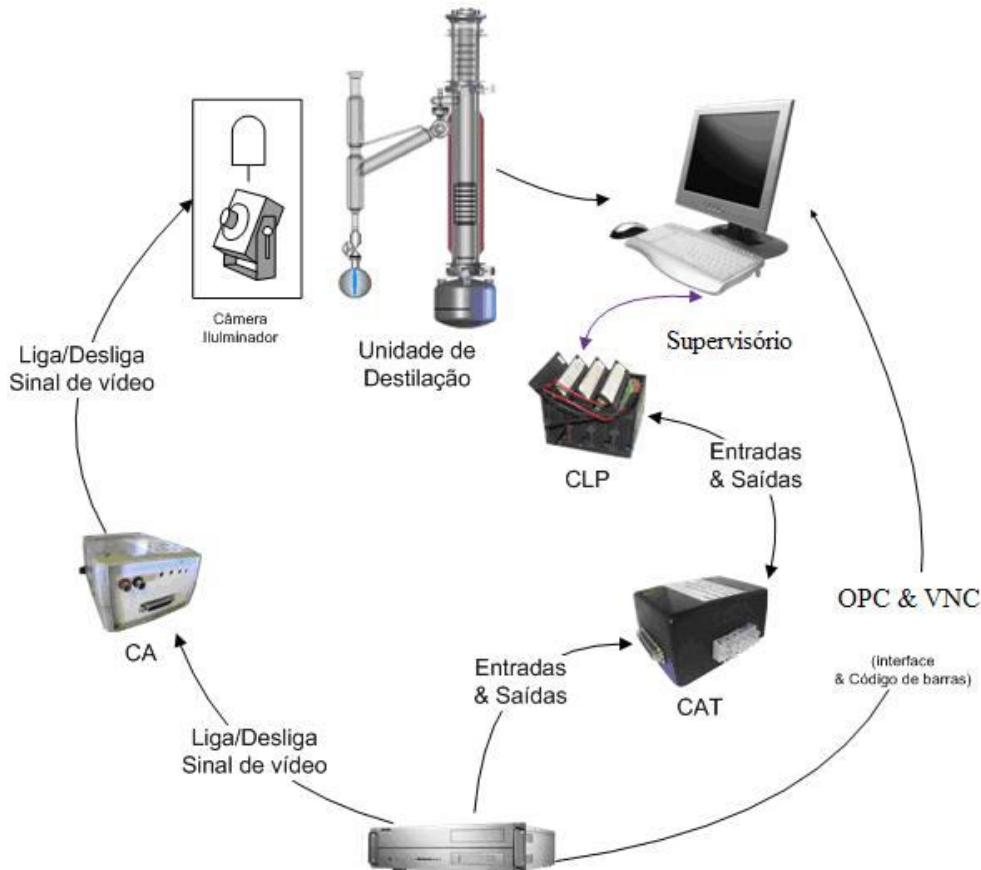
## 1.7. Componentes do Sistema

O projeto desenvolvido fará parte de um sistema maior, ou seja, será um subsistema, e como tal poderá ser acionado de maneira remota. O operador do Sistema Supervisório é quem fará tal acionamento; segue abaixo uma decomposição do sistema de monitoramento desenvolvido.

- **Unidade de destilação:** Unidade dedicada à destilação do petróleo;
- **Supervisório:** Software do sistema de Supervisão do processo de destilação através do CLP;
- **CLP:** Controlador Lógico Programável. Tem a função de realizar a interface entre os sensores da unidade de destilação e o Sistema Supervisório;
- **CA:** Circuito Atuador. Este circuito tem a função de acionar a iluminação e as câmeras de monitoramento do menisco, código de barras e gota;
- **CAT:** Circuito Adaptador de Tensão. Este circuito tem a função de casar o nível lógico da porta paralela (LPT), nível TTL, com o nível lógico do CLP;
- **Sistema de Monitoramento por Vídeo:** Software projetado para analisar e identificar os requisitos da destilação por monitoramento em vídeo.

A Figura 1.1 mostra o esquema da integração das partes que compõem o sistema de monitoramento: a unidade de destilação em batelada, o *subsistema de controle*

composto por câmeras, CA, CAT e CLP (Controlador Lógico Programável), o Sistema Supervisório.



**Figura 1.1: Visão Geral do Sistema de Monitoramento**

## 1.8. Funcionamento Geral do Sistema

O sistema de monitoramento compõe-se de diversos módulos (**Erro! Fonte de referência não encontrada.1.1**) que serão descritos brevemente nesta seção.

O *sistema supervisório* é o encarregado de processar todos os dados coletados pelos sensores do processo de destilação. Com estes dados o *sistema supervisório* controla a destilação, alterando variáveis como, por exemplo, temperatura e vazão segundo a necessidade. Este sistema é executado em um computador exclusivo para não ser sobrecarregado e dedicar-se à somente ao controle do processo de destilação.

Os sensores não interagem diretamente com o supervisório, mas sim com um Controlador *lógico programável* (*CLP*) através de uma rede do tipo *FieldBus*. O *CLP* faz o papel de servidor central na rede *FieldBus*, controlando o tráfego das informações, repassando-as depois para o sistema supervisório. Os sensores da destilação podem ser vistos como os clientes desta rede. Devido o sistema de monitoração ser considerado como mais um sensor da unidade, este também se conectará com o *CLP*. Esta comunicação será efetuada através da porta paralela do computador onde estará sendo executado o *sistema de monitoração por imagem* (computador diferente ao do *sistema supervisor*). Em função de a porta paralela reconhecer e enviar sinais elétricos de nível *TTL* (5 volts) e o *CLP* trabalhar com sinais de 24 volts, foi projetado um circuito *adaptador de tensão* (*CAT*). Este adaptador eleva ou diminui o sinal elétrico para a correta comunicação entre o *CLP* e o computador do *sistema de monitoração por imagem*.

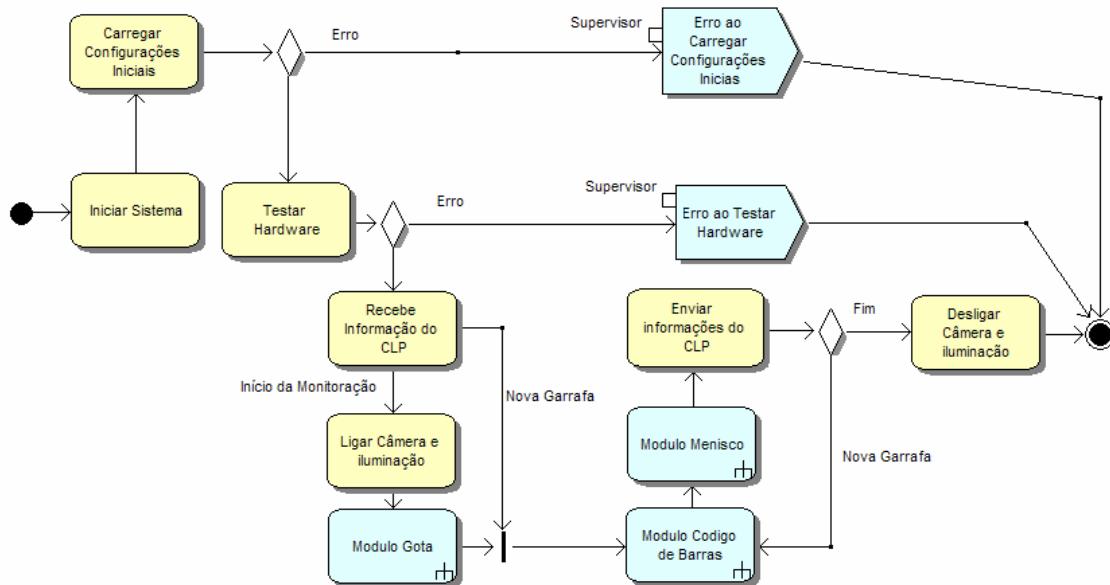
Quando o Módulo Controlador é executado, a primeira coisa que o sistema faz é solicitar ao operador do sistema que realize as calibragens dos Módulos de processamento de imagens. A calibragem consiste basicamente em delimitar as áreas de busca para os algoritmos de processamento de imagens.

Em seguida é realizada a verificação de todos os módulos de hardware que compõem o sistema. Se o teste de hardware for realizado com sucesso, o sistema é habilitado para iniciar o processo de monitoramento da destilação, senão o operador do sistema será notificado da(s) falha(s) em nível de hardware, e o processo de destilação não poderá ser iniciado até que a falha seja solucionada.

Passando pela fase de teste de hardware, o sistema fica aguardando o sinal vindo do CLP (o supervisório é que manda o sinal de início para o CLP), que indica o começo da monitoração do processo de destilação. Quando o Supervisório indica ao sistema o começo da monitoração, a câmera e iluminação posicionadas no bico do destilador são acionadas simultaneamente. Neste ponto, o módulo Controlador cria a thread do módulo da gota, que é responsável pela indicação da queda da primeira gota, e começam a ser processados os quadros de imagens vindos do bico do destilador. O algoritmo quando detecta a primeira gota informa ao módulo Controlador da queda gota dando início a uma série de ações por parte do Módulo Controlador: a câmera e a iluminação do bico são desligadas, um pulso é enviado ao supervisório, seguindo o caminho LPT, CAT e por último CLP, a fim de indicar a este a queda da primeira gota; por último a *thread* da gota é destruída.

Detectada a queda da primeira gota, inicia-se o ciclo de monitoração do Menisco. Neste ponto, o Módulo Controlador fica aguardando o recebimento do sinal de nova garrafa vinda do Supervisório via CLP este sinal corresponderá à leitura do código de barras, seguido da leitura do menisco. Quando o CLP envia o sinal de nova garrafa, o Módulo Controlador liga a câmera do Menisco/Código de barras, liga a iluminação do código de barras e cria a *thread* do Módulo Código de Barras, que será o módulo de processamento de imagens responsável pela detecção do código de barras. Sendo detectado o código de barras pelo Módulo Código de Barras, o mesmo notificará o fato ao módulo Controlador, que irá suspender a *thread* do Módulo do Código de Barras, desligará a iluminação e a câmera do Menisco/Código de Barras, ligará a câmera e a iluminação do Menisco, criará a *thread* do Módulo Menisco, e a partir deste momento será monitorado o menisco. A cada incremento de volume detectado por esse módulo, uma notificação é feita ao supervisório pelo Módulo Controlador (sinal pulso de menisco), que calcula assim o volume presente na garrafa. Quando o subproduto acaba ou a garrafa fica cheia, um sinal de nova garrafa é enviado pelo CLP, dando início ao processo descrito anteriormente (a partir da leitura do código de barras).

O processo descrito nos parágrafos anteriores continua até que seja sinalizado pelo Supervisório, via CLP, o fim processo de monitoração. Neste ponto as câmeras e iluminações são desligadas, e as *threads* são todas destruídas.



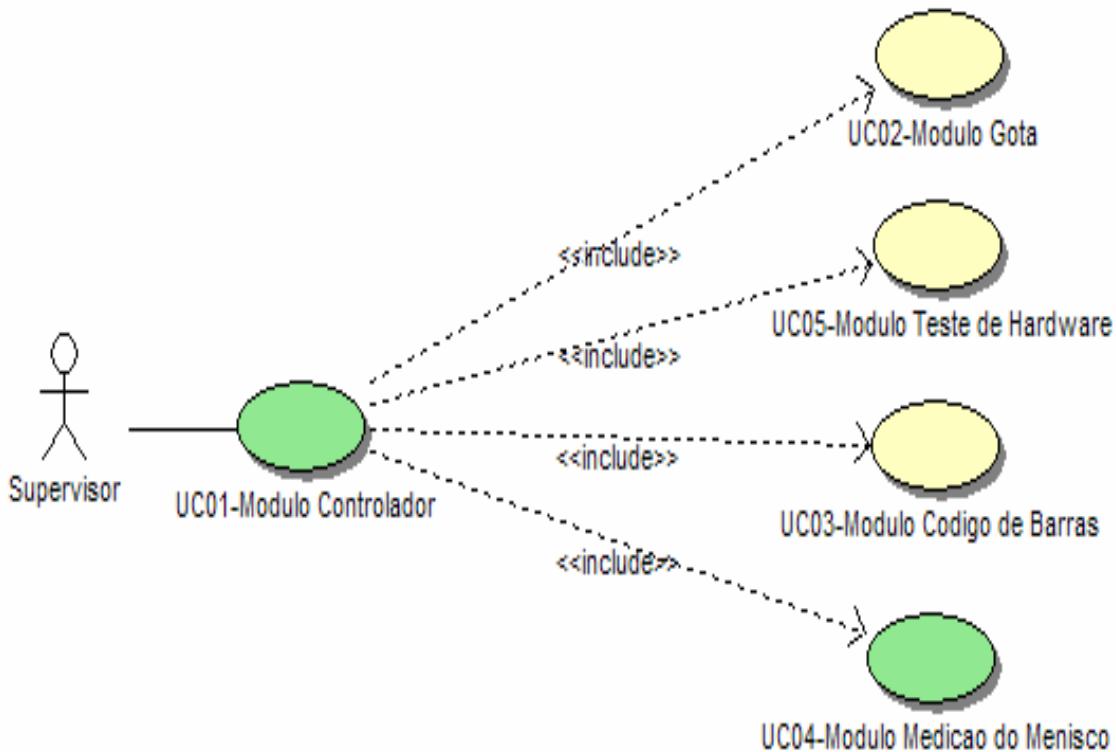
**Figura 2.1: Módulo Controlador**

# Capítulo 2

## O Sistema de Monitoramento

O software do sistema de monitoramento divide-se em quatro módulos independentes: Módulo de detecção da 1<sup>a</sup>. Gota, leitura e identificação do código de barras, monitoramento do menisco e Módulo Controlador (módulo central do processamento) além de outros sub-módulos para teste de hardware, para calibragem do sistema, gerenciamento de rede, disco e outros recursos.

### 2.1. Diagrama de Casos de Uso do Sistema de Monitoramento



**Figura 2.2: Diagrama de Casos de Uso do Sistema de Monitoramento**

*OBS: Os Módulos em bege foram desenvolvidos neste projeto, já os módulos em verde foram desenvolvidos em [1], porém o Modulo Código de Barras foi desenvolvido em*

conjunto. Para maiores detalhes sobre o Modulo Controlador e o Modulo Medicao do Menisco ler[1] juntamente com este projeto.

## 2.2. Módulo de Detecção da Queda da 1<sup>a</sup> Gota

### Descrição:

Este módulo monitora o sistema bico-garrafa para detectar o instante de queda da primeira gota. O objetivo do módulo é monitorar as imagens capturadas pela câmera e retornar um valor estabelecido para identificar que a gota foi detectada. Uma vez detectada a gota o algoritmo termina.

OBS: Falta definir e implementar o procedimento a ser executado caso não seja encontrada a gota (*time-out*).

### 2.2.1. Caso de Uso Módulo Gota

**Objetivo:** Detecção da queda da primeira gota indicativa de início do processo de destilação.

#### Pré-Condições:

1. Módulo Controlador aberto.
2. Módulo Controlador enviar sinal de criar *thread* (Módulo Gota).

#### Fluxo Principal:

1. Módulo Gota lê as configurações iniciais.
2. Cria *thread* Captura de quadros.
3. *Thread* Captura de quadros começa a salvar quadros numa fila.
4. Calibragem da localização do bico.
5. Cria *thread* Análise dos quadros.
6. *Thread* Análise dos quadros pega quadros da fila para processá-los.
7. *Thread* Análise de quadros envia notificações tais como número do quadro onde primeira gota foi detectada, hora, data para a fila notificação.
8. *Thread* Análise de quadros envia o quadro com a primeira gota detectada, bem como alguns quadros anteriores e posteriores a queda para a fila de gota detectada.
9. *Thread* Geração dos Logs é criada.

10. Thead Geração dos Logs retira informações das filas para armazenamento em arquivos de log.

**Pós-Condições:**

1. Queda de primeira gota detectada.
2. Módulo Controlador recebe sinal indicativo de queda de primeira gota.
3. Módulo Gota encerrado.

**Fluxos de Exceção:**

1. *Os fluxos de exceção não foram tratados.*

## 2.2.2. Diagrama Físico do Módulo Gota

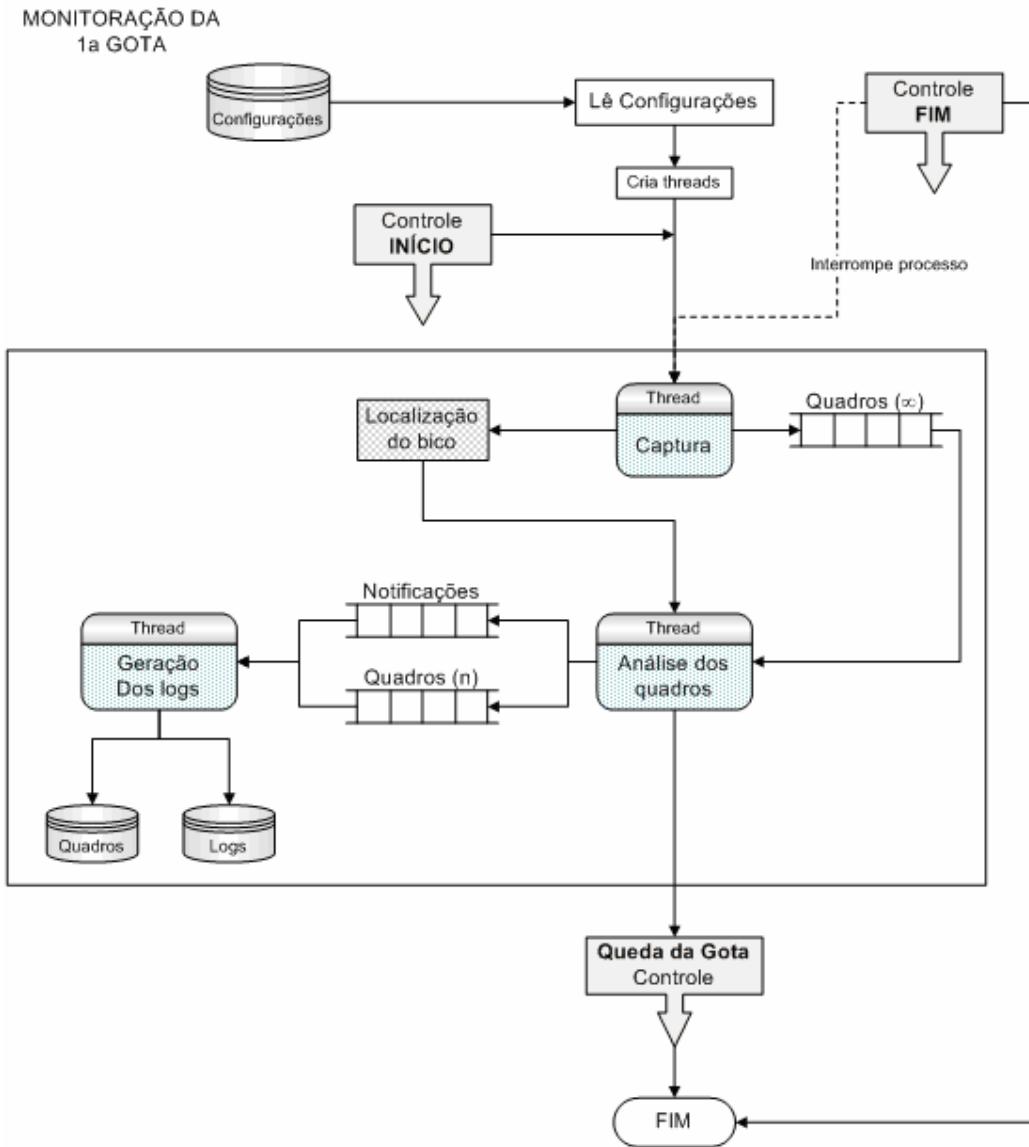


Figura 2.3: Diagrama Físico do Módulo Gota

### 2.2.3. Classe do Módulo Gota

```
//-----
/*
DetectorGota.h
Deteccao de gota (versao 1)
*/
//-----
#ifndef DetectorGotaH
#define DetectorGotaH
//-----
#include "Projecoes.h"
#ifndef TCoordsDefined
/*
struct TCoords
Define as coordenadas (x1,y1) e (x2,y2)
*/
struct TCoords { int x1, y1, x2, y2; };
#define TCoordsDefined
#endif

/*
class TDetectorGota
Deteccao de gota (versao 1)
*/
class TDetectorGota: public TProjecoes
{
    void __fastcall ComputaLimites(void);
    void __fastcall CortaAtualizaComputaProj(void);
    void __fastcall CortaBitmap(TCoords &coords);
    int __fastcall TotalProj(int *hProj, int hSize);
    int __fastcall CalculaOffset(void);
    int somaProjReferencia, offset, threshold_gota, dif;
    TCoords dCoords;
```

```

int metodo;
bool debug;
int dbCnt;
Graphics::TBitmap *iDebugV, *iDebugH;
Graphics::TBitmap *quadroDeReferencia;
Graphics::TBitmap *imgProj;

public:
void __fastcall DefineThresholdGota(int tg=15);
void __fastcall UsaMetodo(int met);
int __fastcall DetectaGotaDeUmaLista(TList *listaDeQuadros);
void __fastcall InicializaDetectaGotaDeUmQuadro(void);
bool __fastcall DetectaGotaDeUmQuadro(Graphics::TBitmap *quadro);
void __fastcall IniciaDeteccao(TCoords &coords, Graphics::TBitmap
*referencia);
__fastcall TDetectorGota(int limBin=200, int limProjs=2, int altArea=20, int
altPre=10);
void __fastcall SetDebug(bool modo);
__fastcall ~TDetectorGota(void);
};

#endif

```

#### 2.2.4. Diagrama de Atividades do Módulo Gota

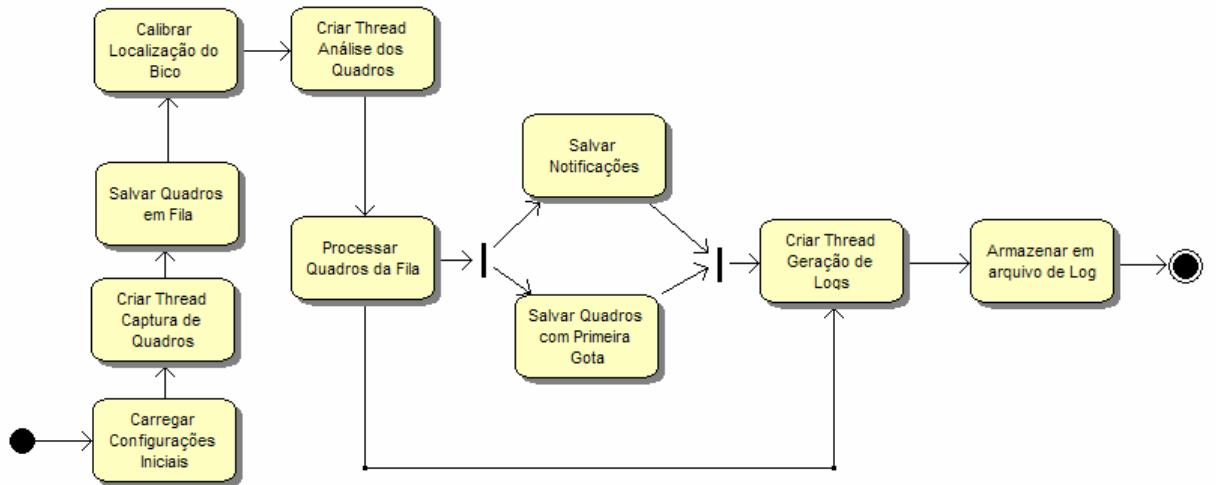


Figura 2.4: Diagrama de Atividades do Módulo Gota

#### 2.2.5. Ambiente de Simulação

Na Figura 2.5 abaixo vemos o ambiente de simulação para desenvolvimento e experimentação que consta de:

- Um destilador (condensador) de vidro
- Um suporte para o condensador
- Iluminadores de leds (2 lanternas)
- Uma garrafa (não padrão)

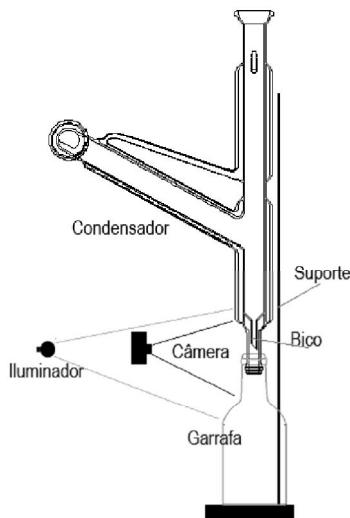
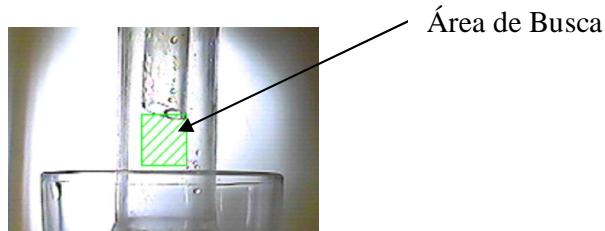


Figura 2.5: Esquema de Montagem do Módulo Gota

O algoritmo desenvolvido para a detecção da gota utiliza como área de busca (Figura 2.6), um retângulo abaixo do bico (retângulo de calibragem) que é monitorado até a queda de uma gota ser detectada.

### **Parâmetros Relevantes para a Detecção:**

- Contraste da imagem
- Limites laterais do bico e do condensador (que contém o bico)
- Altura da área de busca



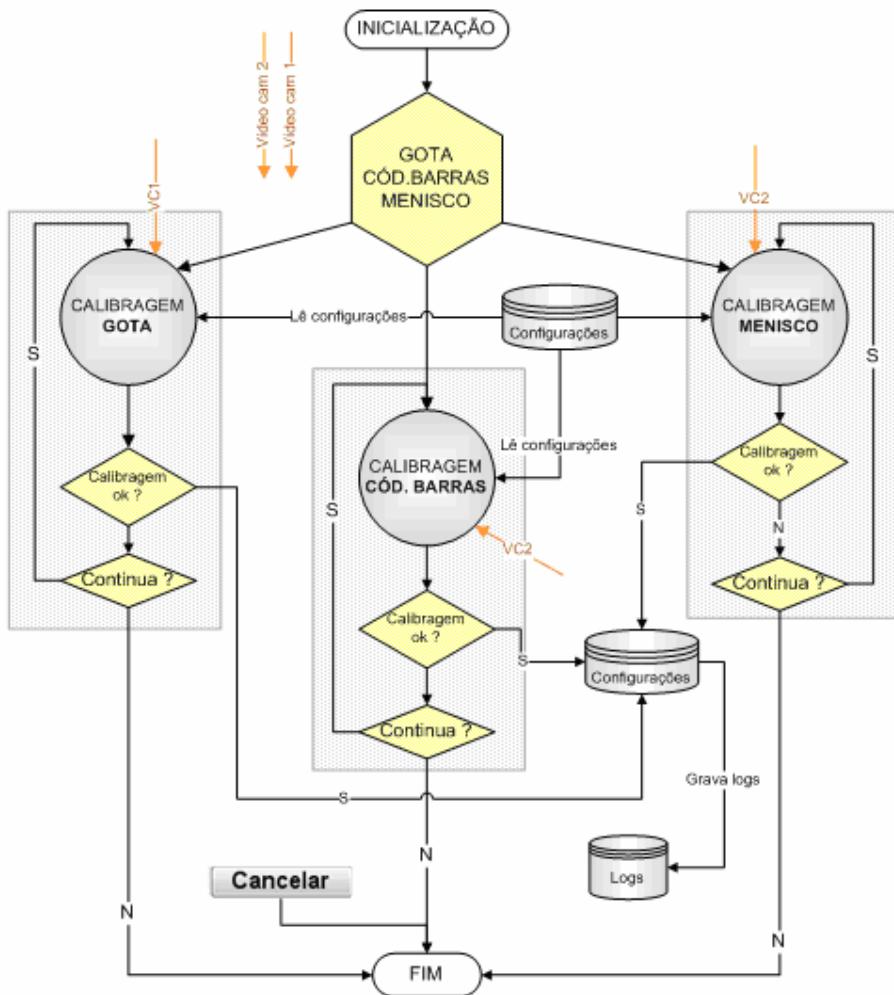
**Figura 2.6: Tomada do Bico Mostrando o Retângulo da Área de Busca**

O retângulo da área de busca é automaticamente definido a partir de um procedimento de calibragem, porém poderá ser manualmente ajustado caso o resultado automático não esteja adequado.

### **Procedimento de Calibragem:**

O sub-módulo de calibragem implementa algoritmos distintos para ajustar os parâmetros específicos de cada módulo, a fim de reduzir erros e aumentar a precisão do desempenho de cada módulo.

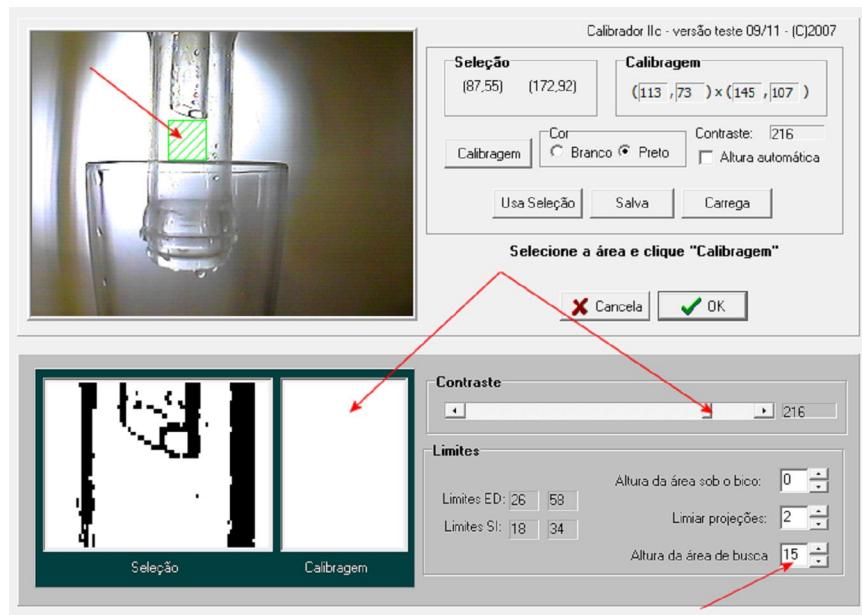
Cada módulo requer uma calibragem específica para o processo a que se refere. Por exemplo: O módulo da gota requer definir a área da busca (retângulo abaixo do bico representando o caminho da queda); O módulo do menisco requer a localização da garrafa e ambos necessitam ajustes de contraste.



**Figura 2.7: Módulo de Calibragem**

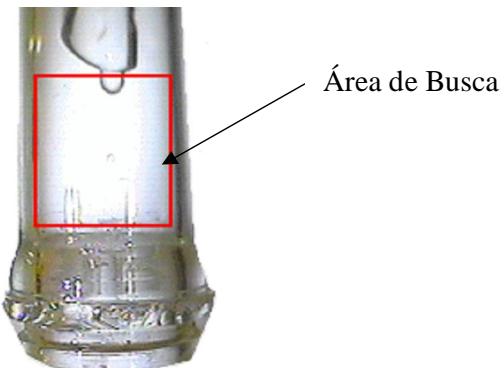
### **2.2.6. Calibragem do Sistema Gota**

Esta sessão mostra a imagem, em tempo real, da câmera que focaliza o Bico do condensador. O operador deverá então desenhar (com o mouse) um retângulo em torno do Bico. O algoritmo deverá, automaticamente, mostrar o retângulo de busca abaixo do bico (como mostrado no retângulo verde na Figura 2.8). Caso este retângulo não apareça ou apareça erroneamente o operador poderá ajustá-lo manualmente; ainda não definidos quais opções de ajustes estarão presentes na versão final como, por exemplo, o contraste (Figura 2.8).



**Figura 2.8: Calibragem para Definição da Área de Busca**

Para melhor desempenho do algoritmo é preciso que a área de busca seja a mais extensa possível. Por isso é necessário que haja uma altura relevante entre o bico e o topo da garrafa inserida (Figura 2.9).



**Figura 2.9: Altura Relevante da Área de Busca**

### 2.2.7. Procedimento de Detecção

A gota é detectada quando a densidade de *pixels* alcança um limiar calculado dentro da área de *offset* (Figura 2.10 e 2.11).

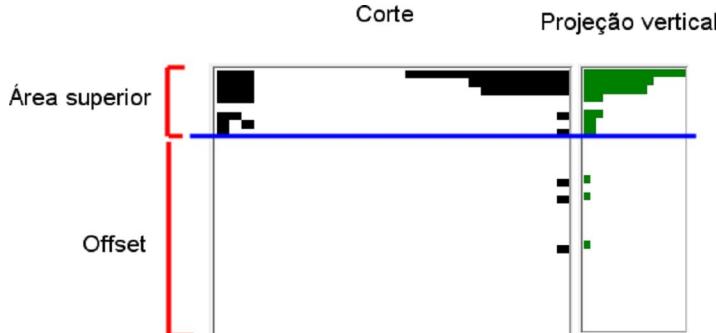


Figura 2.10: Definição da Área de Offset

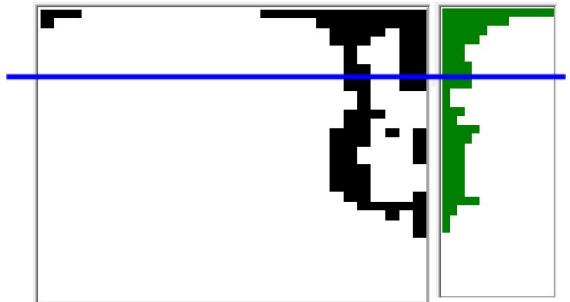


Figura 2.11: Offset Contendo uma Gota

Para análise do sistema e objetivando investigar o comportamento da gota (análise de elementos como velocidade de queda, tempo de formação, tamanho da gota e outros) no ambiente de testes, foram realizadas inúmeras montagens com diversas configurações como mostrado nas Figuras de 2.12 a 2.15.

**Eixo Focal Alinhado ao Bico (acima da borda da garrafa):**

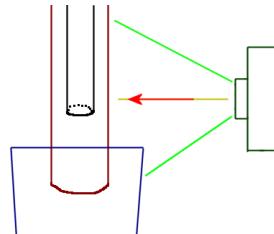


Figura 2.12: Tomada Lateral

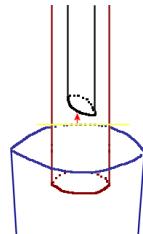


Figura 2.13: Tomada Frontal

**Eixo Focal Alinhado à Borda da Garrafa:**

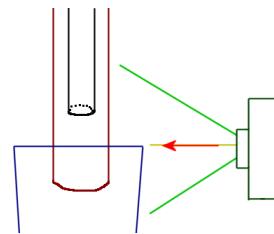


Figura 2.14: Tomada Lateral

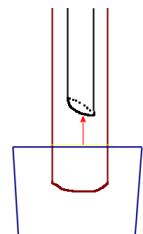
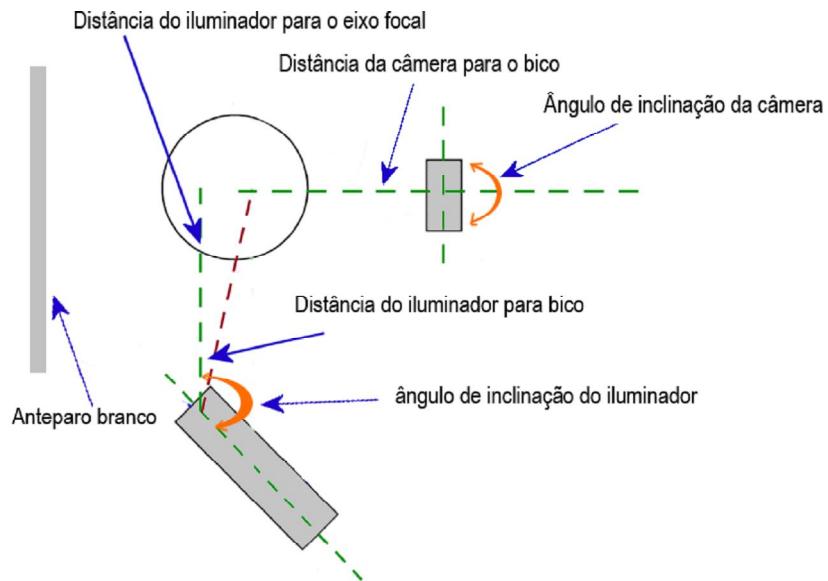


Figura 2.15: Tomada Frontal

Deve se notar aqui que a área de busca é otimizada com a câmera posicionada perpendicular à borda da garrafa.



**Figura 2.16: Vista Superior da Montagem**

## 2.2.8. Resultados dos Testes de Desempenho

**Tabela 1: Ensaios 1 e 2**

ENSAIO 1	
Data	29/10/2007
Material da amostra	água
Quantidade	77
Contraste	231
Calibragem	113, 73, 145, 105
Câmera	
Distância para o objeto	12 cm
Iluminação	
Distância para o objeto	19 cm
Distância para o eixo	14 cm
Tipo	Indireta
Ângulo de incidência	30°
Luminosidade	
Tons escuros na imagem <sup>1</sup>	56 %
Tons escuros na seleção <sup>1</sup>	2,5 %
Tons claros na imagem	44 %
Tons claros na seleção	97,5 %
Detecção	

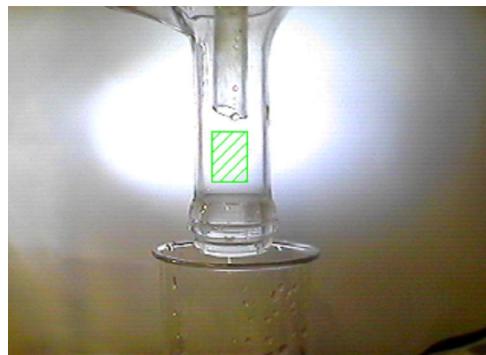
<sup>1</sup> Para a análise da luminosidade, a imagem foi convertida para 4 tons de cinza, considerado tons escuros a quantidade de pretos (nível 0-63) e cinzas escuros (64-127) e tons claros a quantidade de cinzas claros (128-191) e brancos (192-255).

Threshold	<b>12</b>	Acertos	67	Erros	0	Desvios <sup>2</sup>	7(1)	Desempenho	<b>100 %</b>
<b>ENSAIO 2</b>									
Data									29/10/2007
Material da amostra									água
Quantidade									104
Contraste									<b>231</b>
Calibragem									113, 73, 145, 105
Câmera									
Distância para o objeto									12 cm
Iluminação									
Distância para o objeto									20 cm
Distância para o eixo									15 cm
Tipo									Indireta
Ângulo de incidência									25°
Luminosidade									
Tons escuros na imagem <sup>1</sup>									47,7 %
Tons claros na imagem									52,6 %
Tons escuros na seleção <sup>1</sup>									4,6 %
Tons claros na seleção									95,4 %
Detecção									
Threshold	<b>4</b>	Acertos	104	Erros	0	Desvios <sup>2</sup>	12(1)	Desempenho	<b>100 %</b>



Tabela 2: Ensaio 3

ENSAIO 3									
Data									11/14/2007
Material da amostra									água
Quantidade									133
Contraste									<b>232</b>
Calibragem									134, 83, 158, 117
Câmera									
Distância para o objeto									18 cm
Iluminação									
Distância para o objeto									17 cm
Distância para o eixo									15 cm
Tipo									Indireta
Ângulo de incidência									70°
Luminosidade									
Tons escuros na imagem <sup>1</sup>									44 %
Tons claros na imagem									56 %
Tons escuros na seleção <sup>1</sup>									1,1 %
Tons claros na seleção									98,9 %
Detecção									
Threshold	<b>5</b>	Acertos	133	Erros	0	Desvios <sup>2</sup>	26(2)	Desempenho	<b>100 %</b>



<sup>2</sup> Desvio de quadros em relação ao quadro encontrado visualmente. O valor entre parentes representa a tolerância aplicada de 1 ou 2 quadros.

Tabela 3: Ensaio 4

ENSAIO 4	
<b>Data</b>	7/11/2007
<b>Material da amostra</b>	água
<b>Quantidade</b>	139
<b>Contraste</b>	<b>176</b>
<b>Calibragem</b>	117, 62, 159, 122
<b>Câmera</b>	
<b>Distância para o objeto</b>	10 cm
<b>Iluminação</b>	
<b>Distância para o objeto</b>	17 cm
<b>Distância para o eixo</b>	15 cm
<b>Tipo</b>	Indireta
<b>Ângulo de incidência</b>	70°
<b>Luminosidade</b>	
Tons escuros na imagem <sup>1</sup>	41,9 %
Tons escuros na seleção <sup>1</sup>	1,4 %
<b>Detecção</b>	
<b>Threshold</b>	<b>8</b>
<b>Acertos</b>	139
<b>Erros</b>	0
<b>Desvios<sup>2</sup></b>	64(1)
<b>Desempenho</b>	<b>100 %</b>

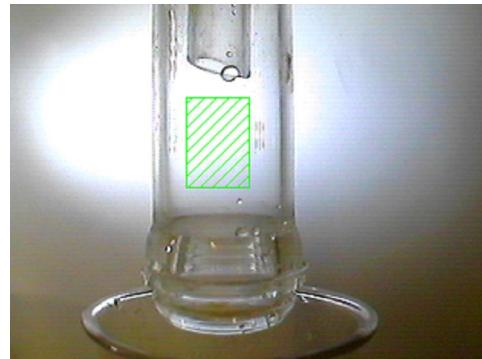


Tabela 4: Ensaio 5

ENSAIO 5	
<b>Data</b>	13/11/2007
<b>Material da amostra</b>	água
<b>Quantidade</b>	119
<b>Contraste</b>	<b>92</b>
<b>Calibragem</b>	96, 97, 184, 187
<b>Câmera</b>	
<b>Distância para o objeto</b>	4 cm
<b>Iluminação</b>	
<b>Distância para o objeto</b>	15 cm
<b>Distância para o eixo</b>	13 cm
<b>Tipo</b>	Indireta
<b>Ângulo de incidência</b>	40°
<b>Luminosidade</b>	
Tons escuros na imagem <sup>1</sup>	45,6 %
Tons escuros na seleção <sup>1</sup>	2,7 %
<b>Detecção</b>	
<b>Threshold</b>	<b>2</b>
<b>Acertos</b>	115
<b>Erros</b>	4
<b>Desvios<sup>2</sup></b>	19(1)
<b>Desempenho</b>	<b>96,6 %</b>

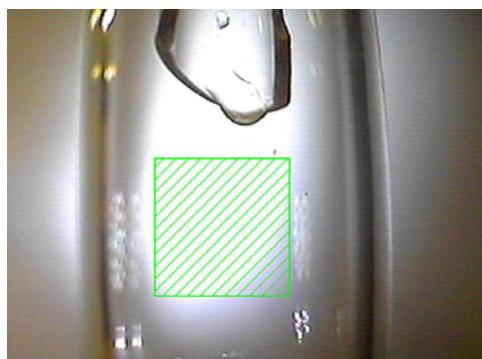


Tabela 5: Ensaio 6

ENSAIO 6	
Data	13/11/2007
Material da amostra	água
Quantidade	66
Contraste	141
Calibragem	96, 98, 184, 179
Câmera	
Distância para o objeto	4 cm
Iluminação	
Distância para o objeto	12 cm
Distância para o eixo	11 cm
Tipo	Indireta
Ângulo de incidência	40°
Luminosidade	
Tons escuros na imagem <sup>1</sup>	41,3 %
Tons escuros na seleção <sup>1</sup>	0,6 %
Detecção	
Threshold	16
Acertos	66
Erros	0
Desvios <sup>2</sup>	22(1)
Desempenho	100 %

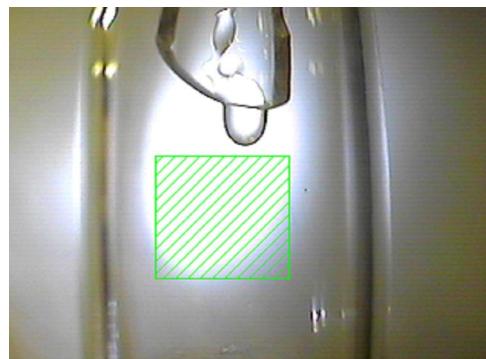


Tabela 6: Ensaio 7

ENSAIO 7	
Data	7/11/2007
Material da amostra	água
Quantidade	151
Contraste	186
Calibragem	117, 62, 159, 122
Câmera	
Distância para o objeto	10 cm
Iluminação	
Distância para o objeto	20 cm
Distância para o eixo	29 cm
Tipo	Indireta
Ângulo de incidência	40°
Luminosidade	
Tons escuros na imagem <sup>1</sup>	39,2 %
Tons escuros na seleção <sup>1</sup>	0,5 %
Detecção	
Threshold	10
Acertos	151
Erros	0
Desvios <sup>2</sup>	29(1)
Desempenho	100 %

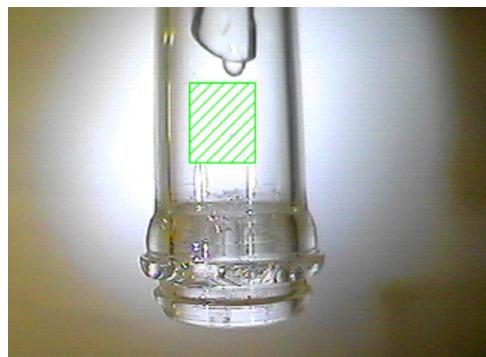
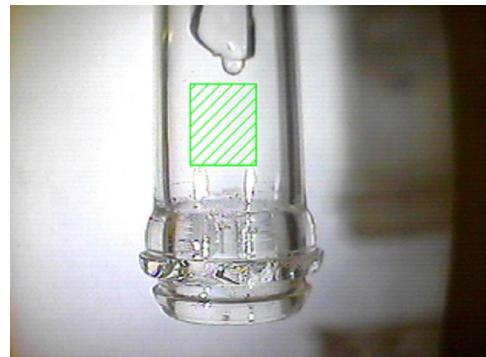


Tabela 7: Ensaio 8

ENSAIO 8	
Data	21/11/2007
Material da amostra	água
Quantidade	151
Contraste	176
Calibragem	119, 52, 163, 106
Câmera	
Distância para o objeto	10 cm
Iluminação	
Distância para o objeto	20 cm
Distância para o eixo	29 cm
Tipo	Indireta
Ângulo de incidência	40°
Luminosidade	

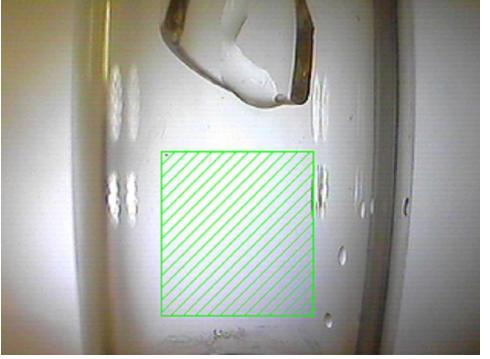


<b>Tons escuros na imagem<sup>1</sup></b>	36,6 %	<b>Tons claros na imagem</b>	63,2 %						
<b>Tons escuros na seleção<sup>1</sup></b>	0,1 %	<b>Tons claros na seleção</b>	99,9 %						
<b>Detecção</b>									
<b>Threshold</b>	<b>10</b>	<b>Acertos</b>	<b>151</b>	<b>Erros</b>	<b>0</b>	<b>Desvios<sup>2</sup></b>	<b>61(2)</b>	<b>Desempenho</b>	<b>100 %</b>

Notas: Igual o Ensaio 7, a menos a distância do anteparo ao bico, 5 cm, contra 15 cm do Ensaio 7

**Tabela 8: Ensaio 9**

<b>ENSAIO 9</b>	
<b>Data</b>	23/11/2007
<b>Material da amostra</b>	água
<b>Quantidade</b>	170
<b>Contraste</b>	132
<b>Calibragem</b>	106, 100, 206, 208
<b>Câmera</b>	
<b>Distância para o objeto</b>	4 cm
<b>Iluminação</b>	
<b>Distância para o objeto</b>	26 cm
<b>Distância para o eixo</b>	24 cm
<b>Tipo</b>	Indireta
<b>Ângulo de incidência</b>	60°
<b>Luminosidade</b>	
<b>Tons escuros na imagem<sup>1</sup></b>	29,6 %
<b>Tons escuros na seleção<sup>1</sup></b>	1,2 %
<b>Detecção</b>	
<b>Threshold</b>	<b>11</b>
<b>Acertos</b>	<b>170</b>
<b>Erros</b>	<b>0</b>
<b>Desvios<sup>2</sup></b>	<b>40(2)</b>
<b>Desempenho</b>	<b>100 %</b>



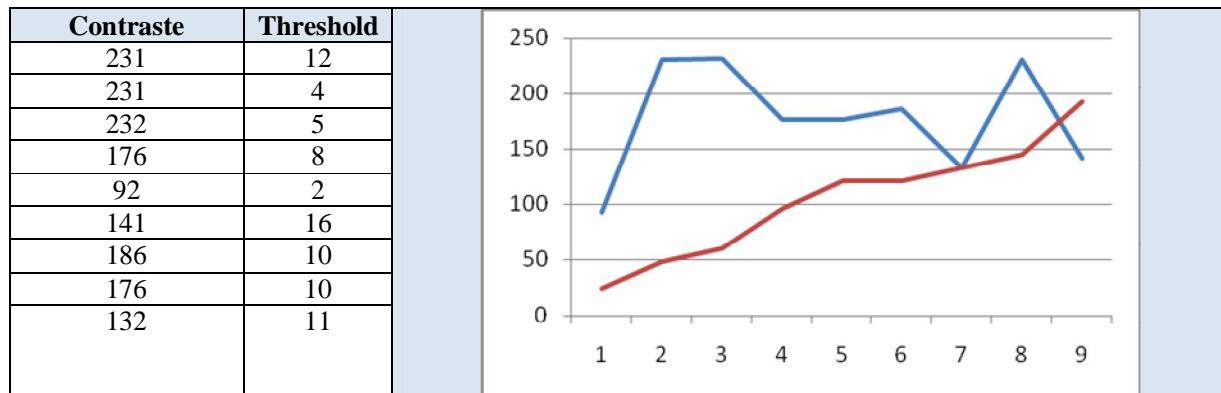
A partir dos dados gerados com os testes realizados, foram definidos os valores mostrados na tabela abaixo.

**Tabela 9: Valores dos Parâmetros de Montagem**

<b>ENSAIO</b>	
<b>Distância da câmera para o bico</b>	<b>7 cm ± 2 cm</b>
<b>Distância do iluminador para o bico</b>	<b>19 cm ± 2 cm</b>
<b>Distância do iluminador para o eixo focal</b>	<b>15 cm ± 1 cm</b>
<b>Ângulo de incidência do iluminador</b>	<b>40° ± 5°</b>

A relação entre o contraste da imagem e o limiar (*threshold*) da densidade de pontos na área de *offset* não é linear e pode ser ajustado com base nos valores da tabela abaixo.

**Tabela 10: Relação entre Contraste e Limiar do offset**



Estes experimentos foram realizados com uma mini-câmera com as seguintes características:

Dispositivo de Imagem: 1/4" Sharp color CCD  
 Ângulo: 52°.  
 Resolução Horizontal: 420 Linhas  
 Iluminação mínima: Color: 0.7 Lux (F:2.0)  
 Lente: 16.0mm

### 2.3. Módulo de Leitura e Identificação do Código de Barras

#### Descrição:

Os fracionamentos do composto destilado deverão ser colhidos em garrafas diferentes. Para identificar estas garrafas e, consequentemente, seus conteúdos, optou-se por utilizar uma etiqueta com um número de dois algoritmos codificado como código de barras.

Este módulo visa ler e identificar este código de barras que é impresso na parte inferior da garrafa. Após o posicionamento de uma nova garrafa, o algoritmo identifica o valor e o retorna ao Controlador.

### **2.3.1. Caso de Uso Módulo Código de Barras**

#### **Objetivo:**

1. Leitura de código de barras via processamento de imagem para associação das garrafas com seus subprodutos.

#### **Pré-Condições:**

1. Módulo Controlador ativo.
2. Módulo Gota ter detectado a queda da primeira gota.
3. Módulo Controlador envia sinal de inicialização do Módulo Código de barras.

#### **Fluxo Principal:**

1. Módulo Código de Barras identifica o código de barras.
2. Módulo Código de Barras envia ao Módulo Controlador o sinal de detecção do código de barras.
3. O Módulo Código de Barras é encerrado

#### **Pós-Condições:**

1. Serão desligadas a iluminação e a câmera do Código de Barras, e serão ligadas a câmera e a iluminação do Menisco.

#### **Fluxos de Exceção:**

*OBS: Esta sendo usado temporariamente um módulo comercial*

### **2.3.2. Classe do Módulo Código de Barras**

```
//-----
#ifndef TCodigoBarrasH
#define TCodigoBarrasH
//-----
#include <Graphics.hpp>
#include <Dialogs.hpp>
#include <SysUtils.hpp>
#include <Printers.hpp>
#include <algorithm>
#include <iostream>
```

```

#include <string>
#include <vector>
#include <map>
#include <cstring>

typedef Graphics::TBitmap TBitmap;
class TCodigoBarras{

    class TCodBParam {
        public:
            int largBarra;
            int largBarraMin;
            int largBarraMax;
            int largImg;
            int altCodBMin;
            int numDigCodB;
            float limiarSim;
            Byte limiarBin;
            TRect dimImgImp;
            TCodBParam(): largBarra(5),largBarraMin(1),largBarraMax(15),
                           largImg(750),altCodBMin(25),numDigCodB(2),
                           limiarSim(0.65),limiarBin(90),
                           dimImgImp(50,50,5650,900){ };
        };
    private:
        //Atributos Privados
        bool seg;
        TBitmap *imgCapturada,*imgBin,*imgRot,*imgSeg,
                *imgGerada,*imgComConexos;
        AnsiString strCodBarrasRec;
        AnsiString strCodBarrasGer;
        TRect rectCodBarras;

        //Métodos Privados

```

```

bool validarSeg(int X,int Y,int XFim, int *YFim, TBitmap *pImgSeg);
void gerarImgSeg(TRect pRect, TBitmap *pImgSeg);

public:
    TCodBParam param;
    //Construtores
    TCodigoBarras(TBitmap *pImgCap);
    TCodigoBarras();
    ~TCodigoBarras();
    //Métodos Públicos
    void binarizar(TBitmap *pImgBin);
    void rotacionar(TBitmap *pImgRot);
    bool segmentar(TBitmap *pImgSeg);
    bool reconhecer(TBitmap *pImgRec);
    void compConexos(TBitmap *pImgComConexos);
    void gerarImgCodB(AnsiString pCodBarraDec);
    void gerarImgCodB(AnsiString pCodBarraDec,
                      int pLargBarra, int pLarg);
    void imprimir(TBitmap *pImgImprimir, AnsiString pCodBarraDec);
    void imprimir(TBitmap *pImgImprimir, AnsiString pCodBarraDec,
                  TRect r);
    void atribuirCodBarras(AnsiString pCodBarras);
    void atribuirImgCap(TBitmap *pImgCapturada);
    AnsiString obterCodBarrasRec();
    AnsiString obterCodBarrasGer();
    TRect obterRectCodBarras();
    TBitmap* obterImgCap();
    TBitmap* obterImgBin();
    TBitmap* obterImgRot();
    TBitmap* obterImgSeg();
    TBitmap* obterImgGerada();
};

#endif

```

### 2.3.3. Diagrama de Atividades do Módulo Código de Barras

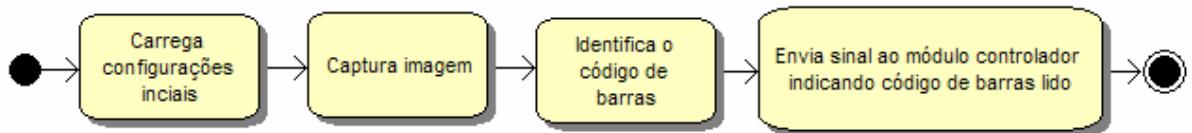


Figura 2.17: Diagrama de Atividades do Módulo Código de Barras

### 2.3.4. Garrafa Coletora

Para o desenvolvimento dos módulos de código de barras e menisco, foi utilizada uma garrafa construída segundo as especificações abaixo:

- Raio interno: 5,1cm
- Altura útil: 13cm
- Cor: transparente
- Altura do fundo falso para o código de barras: 5cm
- Capacidade: 1,0litro

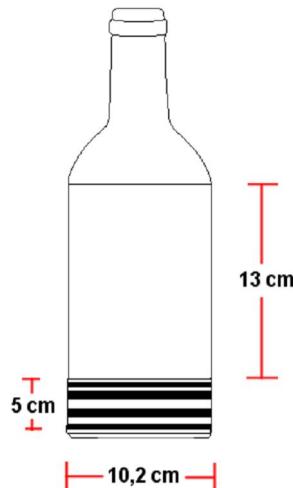


Figura 2.18: Garrafa Coletora

## Problemas Encontrados nas Garrafas Fornecidas:

- Deformações na área em volta do fundo: Estrias em torno da garrafa na altura do fundo. Estas estrias distorcem a imagem nesta área.
- Deformações no fundo: O fundo inserido na garrafa não é plano, possui ondulações concêntricas. Estas deformações mascaram a medida de volume nesta área.

### 2.3.5. Padrão de Codificação

O tipo de código de barras escolhido para o projeto, definido pela equipe anterior, foi o intercalado 2 de 5 (Figura 2.19), que é popular na indústria de transportes. Sua característica mais importante é ser bem compacto. A concepção da direção horizontal (Figura 2.20) visa otimizar a área de visualização na garrafa.

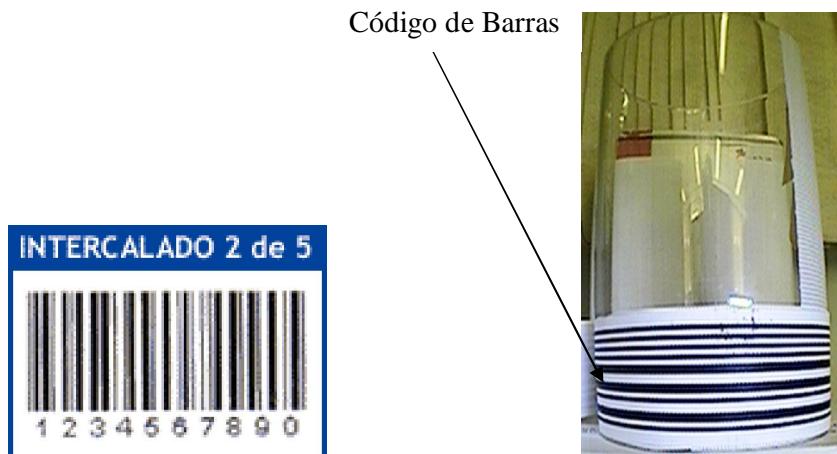


Figura 2.19: Modelo de Código de Barras do Tipo  
Intercalado 2 de 5

Figura 2.20: Código de Barras Intercalado 2  
de 5 Invertido

### 2.3.6. Testes de Desempenho

O algoritmo desenvolvido foi revisado e modificado em uma versão experimental para testes. O objetivo inicial foi verificar o desempenho do algoritmo e identificar falhas.

Para a avaliação de desempenho do algoritmo desenvolvido, foram feitas duas séries de tomadas de imagens seguindo as configurações abaixo. Uma série de imagens repete as configurações para resolução de 320x240 e outra série para imagens de 640x480.

Para cada experimento, foram utilizadas as tiras de código de barras impressas em um conjunto de 40 tiras seqüenciadas como:

- De 01 a 10 -> código de barras de 5 linhas\*
- De 11 a 20 -> código de barras de 6 linhas
- De 21 a 30 -> código de barras de 7 linhas
- De 31 a 40 -> código de barras de 8 linhas

\*Unidade relacionada ao tamanho das barras e espaços

Experimento 1

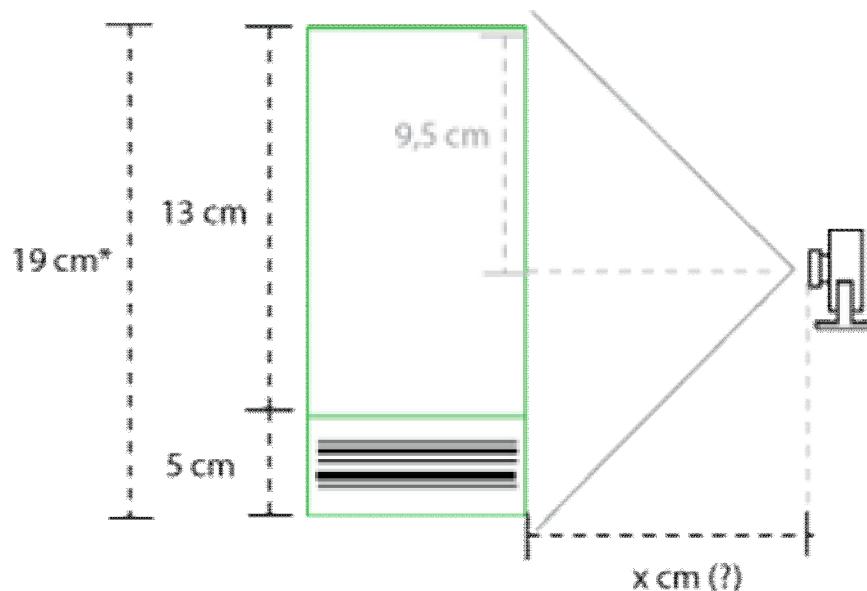


Figura 2.21: Tomada de Fotos com o Eixo Focal Alinhado ao Centro da Garrafa

\*19 cm é a altura da garrafa incluindo a altura da divisão de vidro

## Experimento 2

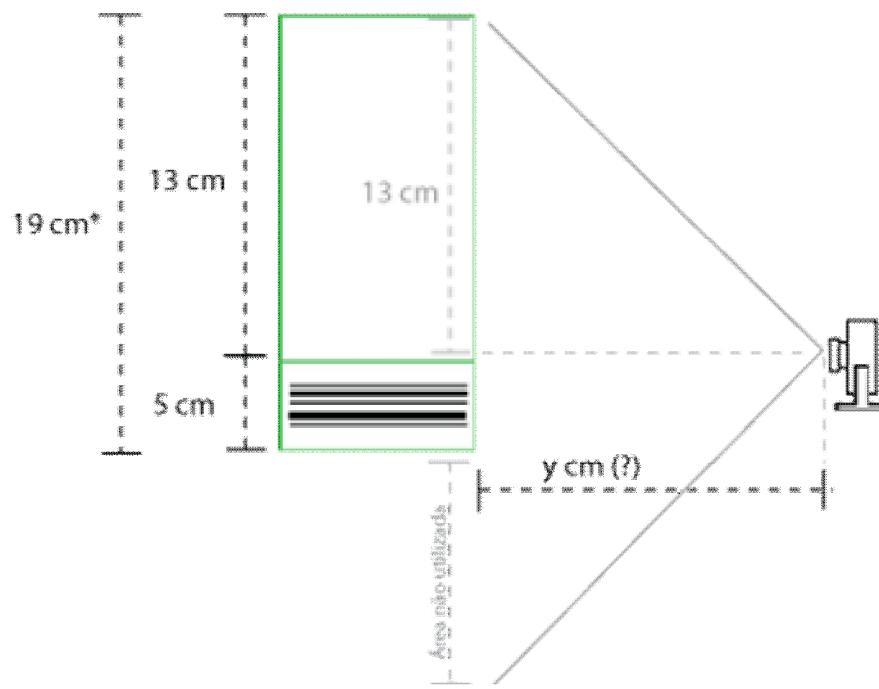


Figura 2.22: Tomada de Fotos com o Eixo Focal Alinhado ao Fundo

## Experimento 3

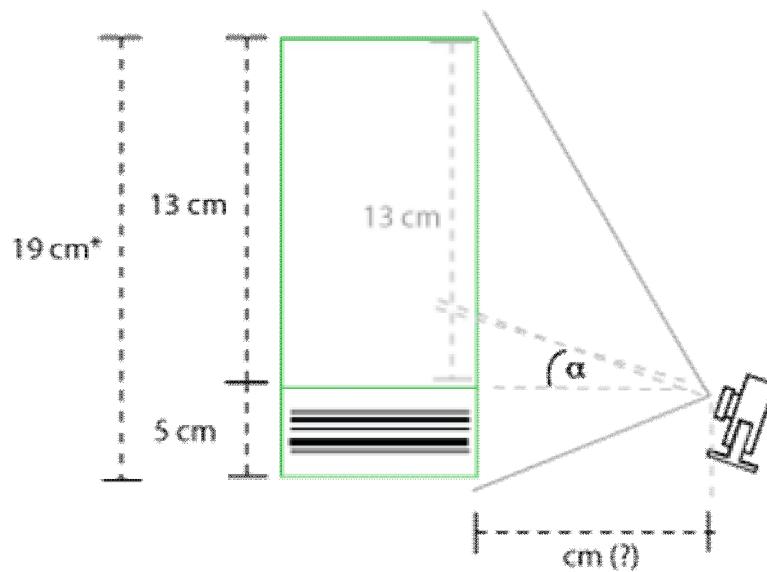
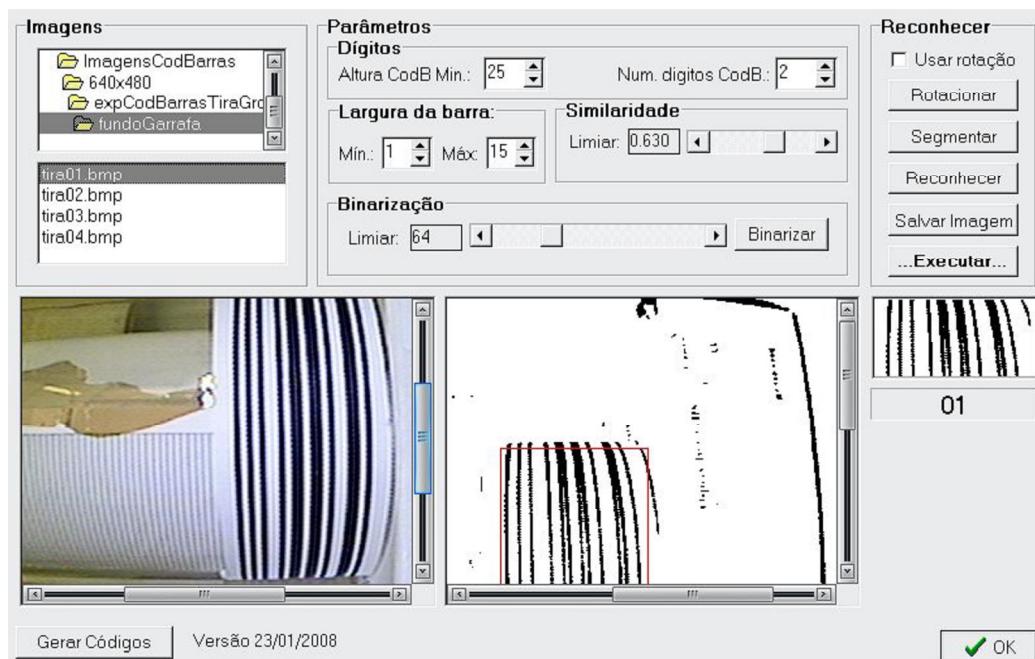


Figura 2.23: Tomada de Fotos com a Câmera Inclinada Enquadrando toda a Garrafa

OBS:

Estou trabalhando no código do programa abaixo, pré-existente, fiz algumas alterações, mas os resultados não estão bons, por conta disso estou usando um componente comercial que não possui interface, ele recebe um bitmap e informa o código de barras para um arquivo tipo txt o qual será enviado para o supervisório, via OPC.



**Figura 2.24: Programa de Testes para Identificação do Código de Barras**

Os testes preliminares mostraram que o algoritmo tem um desempenho melhor utilizando imagens com resolução de 640x480.

## **2.4. Módulos Auxiliares**

### **Ambientes de Testes para Avaliação de Desempenho**

#### **2.4.1. Detecção da Gota**

Com o objetivo de avaliar o desempenho do algoritmo de detecção da 1<sup>a</sup> gota e encontrar a configuração ideal do sistema desenvolveu-se o programa de apoio, Módulo de Testes da Gota.

Numa primeira etapa o sistema de monitoramento foi montado em diferentes configurações e a captura de quadros foi iniciada. Estas imagens foram divididas em diretórios, “experimentos”, sendo que as fotos contidas em cada diretório correspondiam a uma determinada configuração do sistema de monitoramento, ou seja, posição de câmeras, iluminação etc.

Numa segunda etapa, com o uso do Módulo de Testes da Gota cada um dos diretórios citados foi pego e uma marcação manual dos quadros, onde apareciam a queda da gota, foi realizada; ou seja em cada diretório continha um número de imagens onde aparecia uma gota e outras em que não aparecia a gota, todas as imagens em que a gota aparecia foi feita a marcação do quadro. Esta é uma das finalidades do Módulo de Teste da Gota, possibilitar esta marcação manual do quadro que apresenta a queda da primeira gota.

Na terceira etapa foi rodado, sobre cada diretório, o algoritmo de detecção da 1<sup>a</sup> gota que lê os quadros destes para detectar a queda da gota, e assim fazer a marcação automática destes quadros, e então armazenar o resultado nos respectivos arquivos de configuração do sistema.

Na quarta etapa, através da aba de estatística do Módulo de Teste da Gota, rodou-se a rotina de avaliação de desempenho. Esta usa as informações armazenadas nos arquivos de configuração do sistema para montar uma tabela de estatísticas. Em outras palavras, a rotina de estatística compara se os quadros marcados automaticamente estão dentro da tolerância de cada quadro marcado manualmente, levantando assim a estatística de desempenho do algoritmo da queda da 1<sup>a</sup> gota.

A partir desta observação foi possível avaliar o desempenho do algoritmo. Estes procedimentos são importantes, uma vez que o desenvolvimento do algoritmo foi realizado sobre uma pequena base de dados que não contempla todas as possibilidades

de eventos. Estas etapas servem também para se avaliar as condições ambientais ótimas, que são mandatórias para o processo final.

Na Figura 2.25 abaixo temos a interface de um programa auxiliar de teste de desempenho do Módulo Gota.

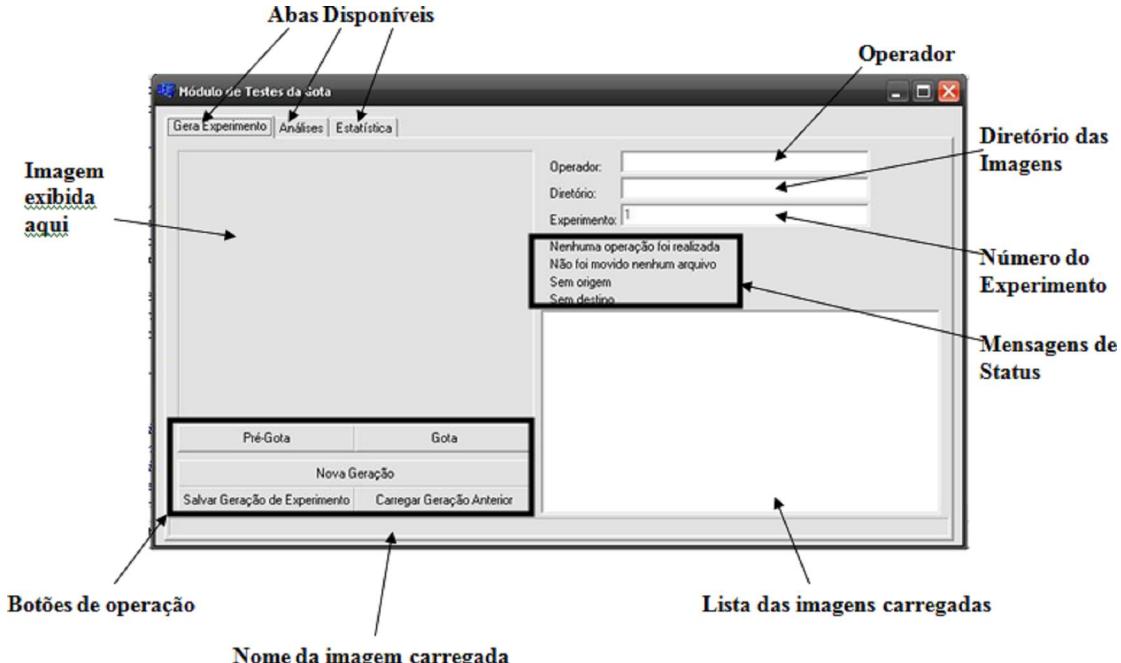


Figura 2.25: Programa Gerador de Experimentos

## 2.5. Módulos de Hardware

### 2.5.1. Circuito Atuador (CA)

Este circuito tem a função de acionar a iluminação e as câmeras de monitoração do menisco, código de barras e gota. A especificação funcional do CA é a seguinte: Quando começa a monitoração da primeira gota, a LPT envia um nível lógico alto (5V) para o CA, este por sua vez acopla a câmera e a iluminação à fonte de 12V. Quando a primeira gota é detectada, a LPT envia um nível lógico baixo (0V) para o CA, desacoplando assim a câmera e a iluminação da fonte DC de 12V. Encerrando-se a monitoração da primeira gota, inicia-se a monitoração do código de barras e do menisco.

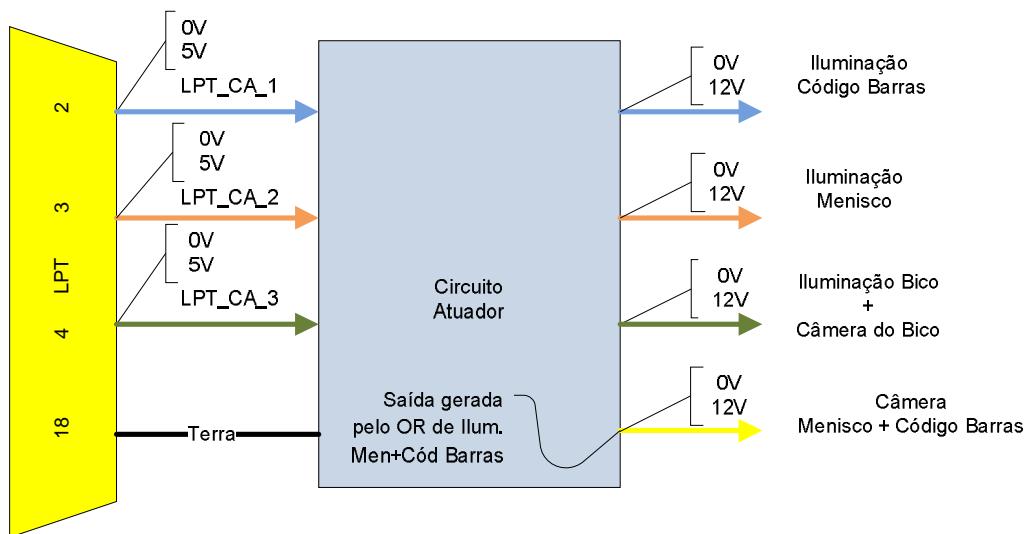
A lógica de controle do CA para o código de barras e menisco é a seguinte: Quando se inicia a monitoração do código de barras, a câmera do código de barras mais

menisco (uma única câmera para os dois processos) é acionada (conectada a uma fonte DC de 12V) além da iluminação do código de barras. Enquanto a leitura do código de barras estiver em andamento, a iluminação do menisco estará desligada; encerrando-se a leitura do código de barras, sua iluminação é apagada e a do menisco é acesa.

Vale a pena salientar que a câmera é ligada e mantida neste estado enquanto um dos dois monitoramentos estiver ocorrendo.

O isolamento entre a LPT e o CA foi realizado com o emprego do buffer 74LS541, visando proteger a LPT do computador.

### **Diagrama de Blocos**



**Figura 2.26: Diagrama de Blocos do Circuito CA**

### **Lógica de Relacionamento da Entrada com a Saída (CA)**

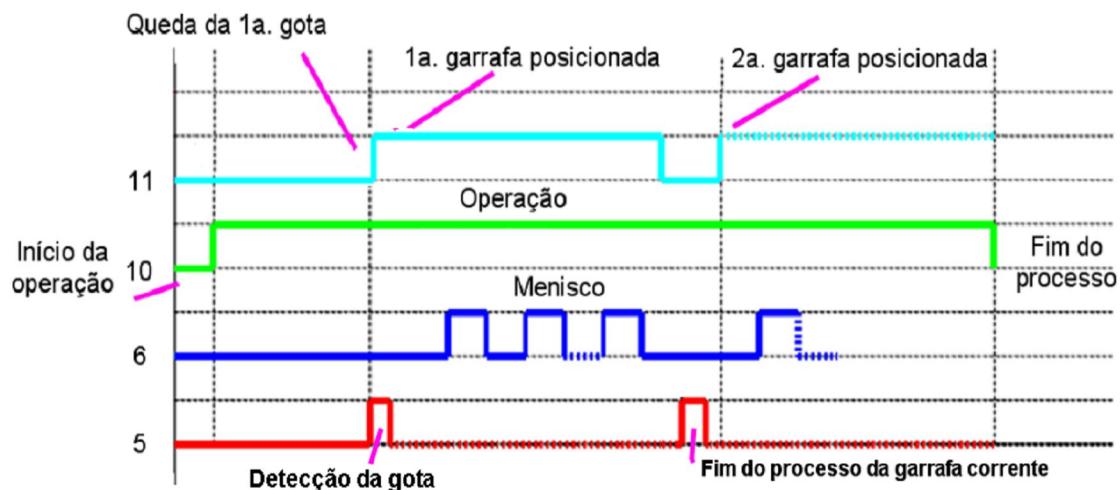
$$\begin{aligned}
 \text{LPT\_CA\_1: } & \left\{ \begin{array}{l} 0V \rightarrow 0V \\ 5V \rightarrow 12V \end{array} \right\} : \text{Ilum. Cód. Barras} \\
 & \left\{ \begin{array}{l} 0V \rightarrow 0V \\ 5V \rightarrow 12V \end{array} \right\} : \text{Ilum. do Menisco} \\
 \text{LPT\_CA\_2: } & \left\{ \begin{array}{l} 0V \rightarrow 0V \\ 5V \rightarrow 12V \end{array} \right\} : \text{Ilum. do Bico + Câmera}
 \end{aligned}$$

**Figura 2.27: Diagrama Esquemático de entrada/saída**

**Tabela 11: Descrição da Pinagem da LPT com o CA**

Pino	Nome/Descrição	Tipo
02	LPT_CA_1/Illuminação do Cód. de Barras + Câmera	Saída
03	LPT_CA_2/Illuminação do Menisco + Câmera	Saída
04	LPT_CA_3/Illuminação do Bico + Câmera	Saída
18		Terra

## Operação



**Figura 2.28: Protocolo de Sinalização com o CLP**

## **2.5.2. Módulo de Teste de Hardware (MTH)**

### **Descrição**

Este módulo terá a função de realizar rotinas de testes dos circuitos, antes do começo do processo de monitoração, com a finalidade de identificar possíveis falhas em cabos de sinais ou módulos eletrônicos (CA, CAT ou CLP) que compõe o sistema. O MTH será composto de dois sub-módulos: sub-módulo teste CA e sub-módulo teste CAT. O primeiro será destinado à identificação de falhas no CA ou no seu cabo de conexão de sinais; o outro será para identificar possíveis falhas no CAT, no CLP ou nos cabos de conexões de sinais.

### **2.5.2.1. Classe do Módulo de Teste de Hardware**

```
//-----
#ifndef UTestaHardwareH
#define UTestaHardwareH
#include <Dialogs.hpp>
#include ".\ParallelPort\UParallelPort.h"
//-----estados-----
#define E_AGUARDANDO_INICIO_MONITORACAO 0
#define E_MONITORACAO_INICIADA 1
#define E_GOTA_DETECTADA 2
#define E_GARRAFA_POSICIONADA 3
#define E_LEITURA_CODIGO_BARRA_REALIZADA 4
#define E_FIM_PROCESSO_MONITORACAO 5
#define INICIAR_TESTE_CA 6
#define INICIAR_TESTE_CAT_CLP 7
#define BICO_LIGADO "Bico Ligado"
//-----
class TTestaHardwareLPT
{
    TLabel *label;
    TMemo *memo;
    int pino10,
        pino13,
```

```
pino12; // entradas da LPT vindas do CLP com interface do CAT
bool estado[6];
TParallelPort *pLPT;

public:
    void __fastcall testarCA () // prototipo da funcao de teste do CA
    void __fastcall testarCAT () // prototipo da funcao de teste do CAT
    int __fastcall lerLPT (int pino); // prototipo da funcao de leitura da LPT
    void __fastcall escreverLPT (int pino, int valor); // prototipo da funcao de escrita
da LPT
    void __fastcall MedeNivel(bool x);
    TTestaHardwareLPT(TLabel *label);
    ~TTestaHardwareLPT(void);
};

//-----
#endif
```

### 2.5.2.2. Sub-Módulo Teste CA

#### Diagrama de Blocos do CA com o MTH do CA (MTH\_CA)

Na Figura 2.29 abaixo temos o circuito atuador (CA) já com o Sub-Módulo de Teste de Hardware do CA incorporado.

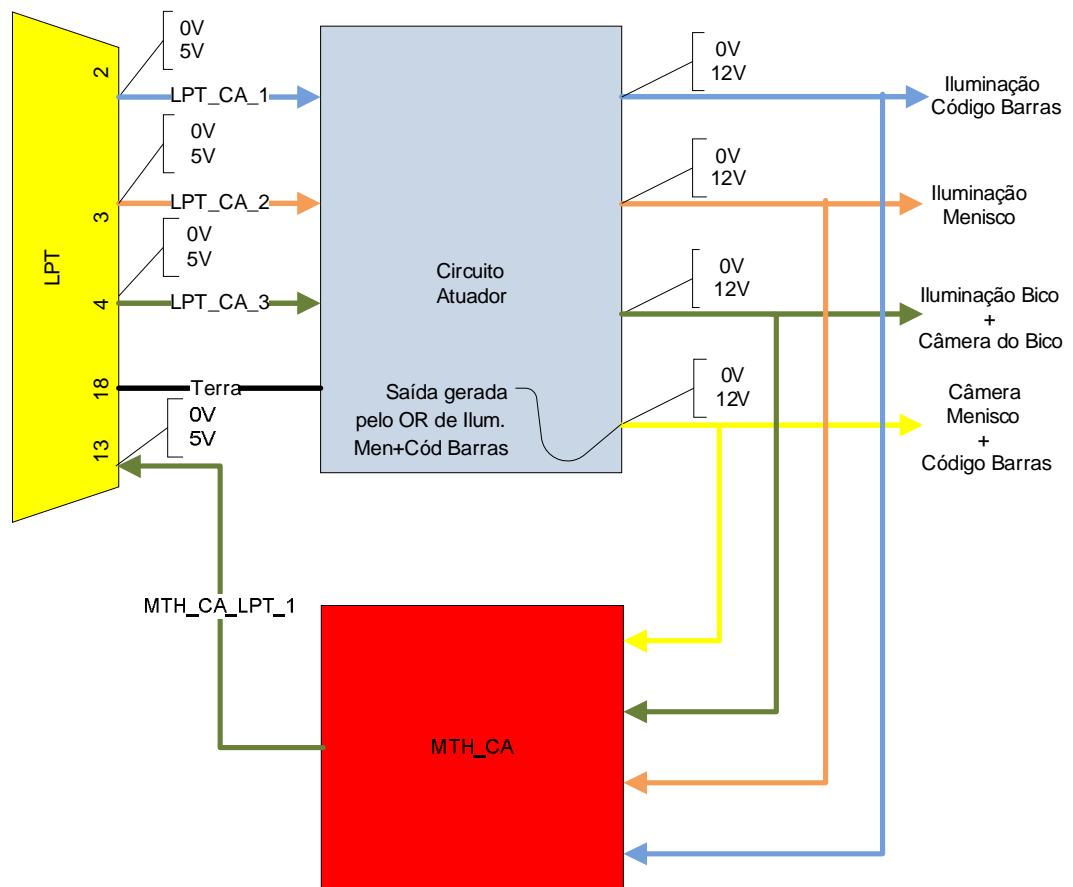


Figura 2.29: Diagrama de Blocos do CA com MTH\_CA Incorporado

**Tabela 12: Descrição da Pinagem da LPT com o CA e MTH\_CA**

Legenda		
Pino	Nome/Descrição	Tipo
02	LPT_CA_1/Illuminação do Cód. de Barras + Câmera	Saída
03	LPT_CA_2/Illuminação do Menisco + Câmera	Saída
04	LPT_CA_3/Illuminação do Bico + Câmera	Saída
13	MTH_CA_LPT_1/Aviso de Teste Hardware (CA)	Entrada
18		Terra

## **Descrição em Alto Nível da Rotina de Teste do CA**

Através dos pinos da LPT ligados ao CA - pino\_02, pino\_03, pino\_04 e pino\_13 será realizada a rotina de testes com o intuito de verificar alguma falha em nível de hardware, antes de iniciar o processo de destilação.

A lógica de teste consistirá em gerar uma seqüência de bits para o CA, a partir dos pinos 2, 3 e 4 da LPT, que através da lógica implementada no MTH\_CA, responderá com um bit para a entrada da LPT, através do pino 13. Se o cabo de conexão de sinais do CA ou o próprio módulo CA estiverem apresentando alguma falha, o algoritmo implementado no MTH irá detectá-la.

### **2.5.2.3. Sub-Módulo Teste CAT**

## **Descrição em Alto Nível da Rotina de Teste**

Através dos pinos da LPT ligados ao CAT, pino\_01 (queda da gota), pino\_10 (começo da monitoração), pino\_12 (nova garrafa) e pino\_14 (pulso de menisco), será realizada a rotina de testes com o intuito de verificar alguma falha a nível de hardware, antes de iniciar o processo de destilação.

A lógica de teste consistirá em gerar uma seqüência de bits para o CLP, a partir dos pinos 1 e 14 da LPT, que através das lógicas implementadas no CLP, responderá com uma seqüência de bits para a entrada da LPT, através dos pinos 10 e 12. Se todos os cabos de conexão, módulo CAT e CLP estiverem funcionando corretamente, o algoritmo implementado dará continuidade para a próxima etapa de teste, que será o CA.

As lógicas implementadas no CLP, correspondem a um XOR para o pino\_10 e a um XNOR para o pino\_12.

**Tabela 13: Tabela-Verdade das Saídas do CLP**

pino_01↔X0	pino_14↔X1	pino_10↔Y0	pino_12↔Y1
0	0	0	1
0	1	1	0
1	0	1	0
1	1	0	1

Na tabela 13 acima temos a tabela-verdade correspondente às funções lógicas a serem implementadas no CLP para que o algoritmo de teste do *Controlador* possa detectar possíveis falhas no hardware do CAT.

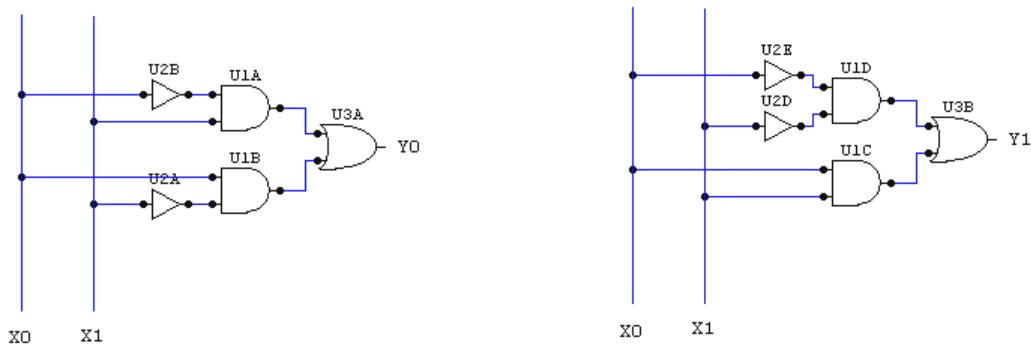
#### 2.5.2.4. Expressões Lógicas das Saídas do CLP

$$Y_0 = (\bar{x}_0 \cdot x_1) + (x_0 \cdot \bar{x}_1);$$

$$Y_1 = (\bar{x}_0 \cdot \bar{x}_1) + (x_0 \cdot x_1).$$

Acima temos as expressões lógicas a serem implementadas no CLP.

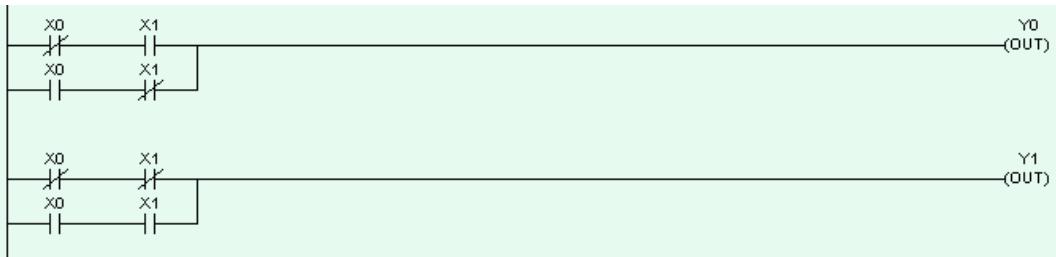
#### 2.5.2.5. Blocos Lógicos das Saídas do CLP



**Figura 2.30: Blocos Lógicos do CLP**

Na Figura 2.30 acima temos os blocos lógicos a serem implementadas no CLP.

## Diagrama de Contatos



**Figura 2.31: Diagrama de Contatos do CLP**

Finalmente, na Figura 2.31 temos em nível de programação CLP (linguagem de Ladder) as lógicas da rotina de testes no CLP.

# Capítulo 3

## O Processamento de Imagem

### Fundamentação Teórica

Este capítulo descreve os fundamentos, as etapas do processamento digital de imagens e como as técnicas de processamento de imagem foram aplicadas ao projeto.

#### 3.1. Fundamentos para Processamento Digital de Imagens

Esta seção apresenta as propriedades da imagem, passando pela definição, a forma de representação, o armazenamento e as propriedades do pixel.

##### 3.1.1. Imagem em Escala de Cinza

Segundo [2], a definição de imagem em escala de cinza pressupõe a existência de dois conjuntos: O conjunto **E** formado por quadrados adjacentes, dispostos ao longo de um número de linhas e colunas (Figura 3.1), e o conjunto **K**, formado pelos níveis de cinza ou escalas de cinza, com **K** variando de 0 a 255.

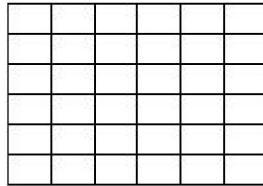


Figura 3.1: Conjunto E

Para [2] um pixel é um elemento do produto cartesiano de um quadrado localizado em **E** e um nível de cinza localizado em **K**, ou seja, o par ordenado  $(x, s)$ , onde **x** é um elemento do conjunto **E** e **s** um elemento do conjunto **K** (Figura 3.2). O mesmo autor define a imagem em escala de cinza como sendo um gráfico de mapeamento de **E** para **K**.

$$(X, S) = (\boxed{\boxed{}} \ , 5) = \boxed{5}$$

Figura 3.2: Produto Cartesiano E x K

### **3.1.2. Definição Matemática de Imagem**

Segundo [3] uma imagem pode ser definida matematicamente como uma função  $f(x, y)$ , onde o valor nas coordenadas espaciais  $xy$  corresponde ao brilho (intensidade) da imagem nessa coordenada (Figura 3.3).



**Figura 3.3: Imagem em Níveis de Cinza**

### **3.1.3. Imagem Digital**

Uma imagem de um objeto real é em princípio contínua tanto na sua variação espacial como nos níveis de cinza [3]. Visando o seu processamento computacional a imagem deve ser digitalizada, ou seja, discretizar tanto no espaço quanto na amplitude. A digitalização das coordenadas espaciais é chamada amostragem da imagem e a digitalização da amplitude é chamada de quantização dos níveis de cinza [4].

### **3.1.4. Representação de Imagens**

Existem duas maneiras de representar uma imagem: representação vetorial e representação matricial. Estas representações diferem na natureza dos dados que compõe a imagem.

Segundo [5] uma imagem vetorial é uma imagem de natureza geométrica, ou seja, ela é definida em função de elementos geométricos e parâmetros (Figura 3.4).



**Figura 3.4: Representação Vetorial [5]**

Segundo [5] uma imagem matricial é uma imagem de natureza discreta, ou seja, a imagem é formada de elementos independentes, dispostos na forma de uma matriz, cada um contendo uma unidade de informação da imagem (Figura 3.5). Esta representação não armazena nenhuma informação geométrica dos objetos contidos na imagem o que torna difícil a manipulação de sua estrutura. No entanto é capaz de representar qualquer tipo de imagem.



**Figura 3.5: Representação Matricial [5]**

### 3.1.5. Propriedades de um Pixel

Uma propriedade importante de um *pixel* é sua vizinhança, esta seção define a vizinhança 4 e a vizinhança 8 de um *pixel*.

#### •Vizinhança 4

A vizinhança 4 de um pixel P é definida pelo conjunto dos pixels adjacentes a P, não levando em conta os pixels localizados nas diagonais passando por P (Figura 3.6).

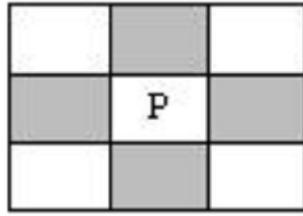


Figura 3.6: Vizinhança 4 de um *Pixel*

#### •Vizinhança 8

A vizinhança 8 de um pixel P é formada pelo conjunto de todos os pixels que são adjacentes a P (Figura 3.7).

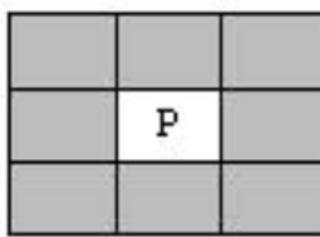


Figura 3.7: Vizinhança 8 de um *Pixel*

### 3.1.6. Formatos de Imagens

O formato de uma imagem refere-se ao modo como os dados da imagem estão armazenados em arquivos e como se faz a interpretação desses dados para a visualização da imagem. Neste projeto foram considerados os seguintes formatos de armazenamento de imagem:

- JPEG;
- PNG;
- BitMap;
- Tiff;

**•Formato JPEG**

Segundo [Ramos (2000)] o formato JPEG é o mais utilizado no armazenamento e na transmissão de imagens estáticas em multimídia e na *internet*. Dentre as principais características do JPEG estão:

- Grande capacidade de compressão;
- Padronização internacional;
- Pequena perda perceptível
- Ampla aceitação pela comunidade da *internet*

**•Formato PNG**

O formato PNG, segundo [6] é indicado para imagens que necessitam de compressão sem perdas, é um formato bastante aplicado na *internet* e em editoração de imagens. Dentre as principais características do PNG estão:

- Formato livre;
- Permite paleta de cores;
- Permite cores com oito, 16, 32, e 48 *bits* por *pixel*;
- Permite transparência em oito e 16 *bits*;
- Permite múltiplas plataformas;

Estrutura de dados do arquivo PNG:

Os primeiros oito bits são reservados para assinatura PNG

IHDR - Cabeçalho da imagem

PLTE - Paleta de cores usadas na imagem

IDAT - Chunks de dados da imagem

IEND - Indicador de fim de dados



**Figura 3.8: Estrutura do Arquivo PNG [6]**

**•Formato BMP**

O formato *bitmap* foi desenvolvido pela *Microsoft* para aplicações no sistema *Windows*. É um formato de estrutura muito simples, tornando mínimas as possibilidades de erros na interpretação dos dados. Os arquivos possuem geralmente a extensão. BMP

e aceitam imagens com um, quatro, oito, 16, 24 e 32 bits por *pixel* [6]. A Figura 3.9 abaixo mostra a estrutura do arquivo *bitmap*.

File Header	Image Header	Color Table	Pixel Data
-------------	--------------	-------------	------------

Figura 3.9: Estrutura do Arquivo Bitmap [6]

#### •Formato TIFF

O formato TIFF foi desenvolvido em 1986 em uma tentativa de criar um padrão para imagens geradas por equipamentos digitais. O formato é capaz de armazenar imagens em preto ou branco, escalas de cinza e em paletas de cores com 24 ou com 32 bits. O TIFF é reconhecido por praticamente todos os programas de imagem [6]. A Figura 3.10 ilustra a estrutura do arquivo TIFF.

Header	Image File Directory	Directory Entry	Data Block
--------	----------------------	-----------------	------------

Figura 3.10: Estrutura do Arquivo TIFF [6]

## 3.2. Etapas para o Processamento Digital de Imagens

Esta seção apresentará os passos para o processamento digital de imagens, enfatizando as etapas abordadas neste projeto, Aquisição de imagem, Realce de imagem, Segmentação de imagem e Classificação de imagem.

### 3.2.1. Aquisição de Imagens

Para obtenção de imagens digitais são necessários dois elementos: dispositivos físicos captadores de imagem e digitalizador. Dispositivos físicos são sensíveis a espectros de energia eletromagnética e o digitalizador converte o sinal elétrico desses dispositivos para o formato digital. Estes elementos são chamados de sistemas de imageamento. O exemplo mais conhecido deste sistema é câmera digital (Figura 58), outros são *scanners* e sensores presentes em satélites. Detalhes sobre aquisição de imagem podem ser vistos em [7].



**Figura 3.11: Câmera Digital**

### **3.2.2. Realce de Imagem**

O melhoramento de imagem é obtido através de técnicas, tais como o melhoramento de contraste e filtragem aplicadas com finalidades específicas enfatizando características de interesse ou recuperando imagens que sofreram algum tipo de degradação devido à introdução de ruído, perda de contraste ou borramento.

A aplicação dessas técnicas, designadas como realce de imagem, são transformações radiométricas que modificam o valor dos níveis de cinza dos pontos da imagem.

#### **•Melhoramento de Contraste**

Melhoramento de contraste busca melhorar a qualidade visual da imagem através da manipulação dos níveis de cinza. Uma imagem possui valores de intensidade de *pixel*, variando de 0 a 255. Quanto mais espalhados os *pixels* da imagem neste intervalo melhor é o seu contraste. Para [3] contraste consiste numa diferença local de luminância e pode ser definido como a razão dos níveis de cinza médios do objeto e do fundo.

O processo de melhoramento de contraste transforma a escala de cinza de forma pontual, ou seja, o novo valor do ponto depende somente do valor original do ponto. Uma função de transferência mapeia o valor de um ponto para um novo valor. Essa função é definida da seguinte forma:  $g(x, y) = T(f(x, y))$  onde,  $f(x, y)$  é o valor do nível de cinza original,  $T$  é a função de transferência e  $g(x, y)$  é o novo valor do ponto.

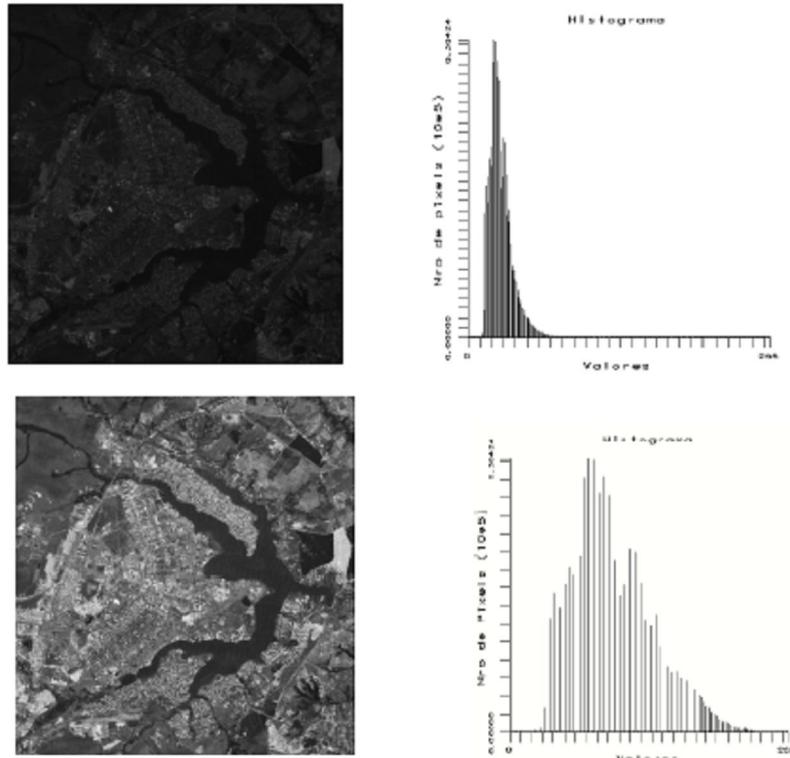
Uma boa forma de avaliar o contraste de uma imagem é analisar seu histograma. O histograma é um gráfico que representa a distribuição dos *pixels* para cada nível de

cinza da imagem. No eixo horizontal fica a escala de cinza e no eixo vertical fica a quantidade de *pixels*.

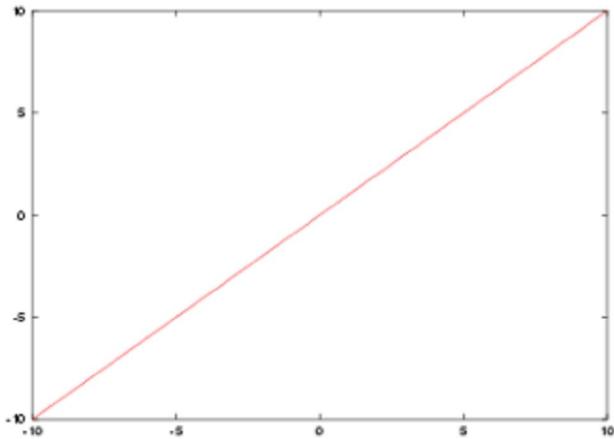
A (Figura 3.12) ilustra duas imagens com seus respectivos histogramas, note que se trata da mesma imagem. Observando o histograma da primeira imagem pode-se notar que a distribuição dos *pixels* concentra-se próximo ao nível zero, exemplificando uma imagem com baixo contraste. A segunda imagem possui alto contraste, seus *pixels* estão melhor distribuídos no histograma, possibilitando um melhor discernimento das informações contidas na imagem.

Funções de transferências modificam histograma para obter uma imagem com melhor contraste. As funções a seguir foram retiradas de [8].

- Função Linear =>  $g(x, y) = a*f(x, y) + b$ , onde os parâmetros  $a$  e  $b$  são valores de ajuste da função que variam de 0 a 255 (Figura 3.13). Esta função é utilizada para redistribuir os *pixels* da imagem de forma linear.

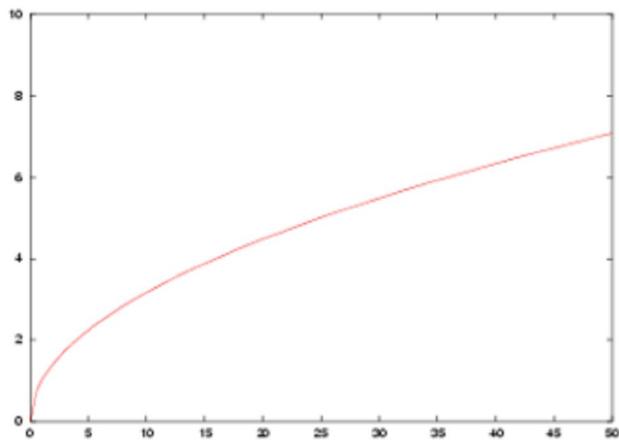


**Figura 3.12: Histograma de Imagens [8]**



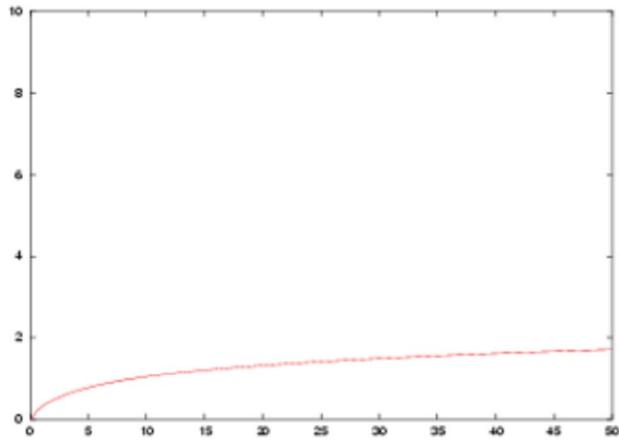
**Figura 3.13: Função Linear**

- Raiz Quadrada =>  $g(x, y) = a * [f(x, y)]^{1/2}$ , onde o parâmetro a é o fator de ajuste da função que varia de 0 a 255. Esta função é utilizada para realçar as áreas escuras da imagem. Este comportamento é facilmente verificado no gráfico da Figura 3.14, note que a inclinação da função é maior próximo a áreas escuras da imagem;



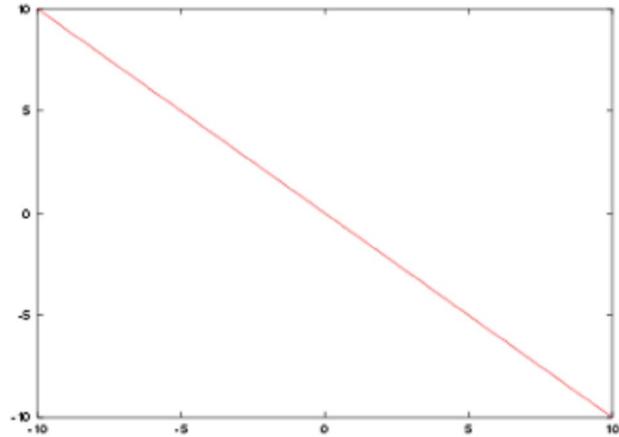
**Figura 3.14: Função Raiz Quadrada**

- Logaritima =>  $g(x, y) = a * \log(f(x, y) + 1)$ , onde o parâmetro a é o fator de ajuste da função que varia de 0 a 255. Esta função tem um comportamento semelhante ao da raiz quadrada, porém realçando um intervalo menor de áreas escuras (Figura 3.15);



**Figura 3.15: Função Logaritmica**

- Inversa =>  $g(x, y) = -(a*f(x, y)+b)$ , onde os parâmetros a e b são valores de ajuste da função que variam de 0 a 255. Esta função inverte os níveis de cinza da imagem, ou seja, valores de pixels pretos são mapeados para brancos, valores brancos são mapeados para pretos e valores intermediários são mapeados para valores intermediários (Figura 3.16);



**Figura 3.16: Função Inversa**

#### •Filtragem

O processo de filtragem procura extrair informações como as bordas da imagem ou corrigir algumas degradações na imagem, tais como: borrões, ruídos inseridos pelo processo de imageamento ou na transmissão da imagem. Os filtros são divididos em duas categorias: filtros no domínio do espaço e filtros no domínio da freqüência. Detalhes sobre filtragem no domínio da freqüência em [7].

A filtragem no espaço é considerada uma operação local, ou seja, o nível de cinza de um ponto depende do original e de sua vizinhança. O princípio de funcionamento de tal filtro está baseado em máscaras de deslocamento as quais são matrizes com pesos associados em cada posição. A máscara com centro na posição (x, y) é colocada sobre o *pixel* a ser modificado na imagem. O *pixel* correspondente na imagem é substituído por um valor que considera os *pixels* vizinhos e os pesos correspondentes na máscara. A soma de todos os produtos dos pesos da máscara pelos *pixels* correspondente na imagem resulta em um novo valor de cinza que substituirá o *pixel* central.

#### •Filtros Detectores de Bordas

Segundo [3], a borda é caracterizada por mudanças abruptas de descontinuidade na imagem. Através do processo de detecção de bordas pode-se localizar e distinguir os objetos presentes na imagem, bem como descobrir algumas de suas propriedades, tais como forma e tamanho.

Os detectores de bordas se baseiam na idéia de derivadas. A derivada mede a taxa de variação de uma função. Em imagens esta variação é maior nas bordas e menor em áreas constantes. Ao percorrer a imagem marcando os pontos onde a derivada possui uma variação maior, ou seja, pontos de máximo, no fim todas as bordas terão sido marcadas [9].

Como as imagens são bidimensionais, usam-se derivadas parciais nas direções vertical y e horizontal x, que são representadas por um vetor gradiente.

- Operador gradiente de Roberts

O operador gradiente de Roberts utiliza uma matriz ( 2 X 2 ), nas direções horizontal Gx e vertical Gy [9].

0	1
-1	0

1	0
0	-1

  
**Gx**
**Gy**

**Figura 3.17: Máscara de Roberts**

O gradiente é calculado pela seguinte formula:  $GR = [G(x \frac{1}{2}) + G(y \frac{1}{2})]^{\frac{1}{2}}$

- Operador gradiente de Sobel

O operador gradiente de Sobel utiliza matriz ( 3 X 3) [9].

1	2	1
0	0	0
-1	-2	-1

**Gx**

-1	0	1
-2	0	2
-1	0	1

**Gy**

**Figura 3.18: Máscara de Sobel**

O gradiente é calculado pela seguinte formula:  $GR = [G(x \frac{1}{2}) + G(y \frac{1}{2})]^{1/2}$

#### •Filtros de Ruídos

Para [9] os ruídos são um conjunto de *pixels* aleatórios diferentes dos dados da imagem.

- Filtro da Média

O filtro da média utiliza uma máscara que percorre a imagem e substitui cada *pixel* da imagem pela média de seus vizinhos. O objetivo desse filtro não é de eliminar o ruído e sim suavizá-lo. A mascara utilizada é uma matriz (3 X 3) baseada na vizinhança 8 [10].

$$\frac{1}{9} * \begin{array}{|c|c|c|} \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline \end{array}$$

**Figura 3.19: Máscara da Média**

- Filtro da mediana

O objetivo do filtro da mediana é substituir o *pixel* central de uma matriz, geralmente 3 X 3, pelo valor que estiver na mediana desses *pixels* ordenados em ordem crescente de nível de cinza [10].

Este filtro pode ser facilmente implementado da seguinte forma:

- 1) Para todos os pontos da imagem;
- 2) Escolha um ponto e o atribua a x;
- 2.1) Coloque em ordem crescente x e sua vizinhança 8, guarde esses valores

em um conjunto X;

2.2) O novo valor do ponto é o valor que está na mediana do conjunto X;

Este filtro baseia-se no fato de que se o ponto for um ruído, ao colocar em ordem crescente este ponto e sua vizinhança, o ruído tende a ficar em um dos extremos do conjunto X, sendo substituído pelo valor mediano do conjunto X.

### 3.2.3. Segmentação de Imagens

Por segmentação de uma imagem entende-se a extração ou identificação dos objetos contidos na imagem, onde o objeto é toda característica com conteúdo semântico relevante para a aplicação desejada [3]. A segmentação faz parte de um passo maior que é a classificação da imagem. A etapa de segmentação divide a imagem em regiões, sem considerar o processo de classificação. Para [3], a região em uma imagem é um conjunto de pontos ligados, ou seja, de qualquer ponto da região pode-se chegar a qualquer outro ponto por um caminho inteiramente contido na região, baseado na vizinhança 4 ou vizinhança 8.

#### Segmentação Baseado em Crescimento de Regiões

A segmentação baseada no crescimento de regiões pode ser feito de duas formas: por varredura ou por busca em todas as direções. Ambos os métodos baseiam-se no fato das regiões a serem detectadas apresentarem propriedades locais aproximadamente constantes como, por exemplo, o nível de cinza.

##### • Crescimento de Regiões por Busca em todas as Direções.

No crescimento por busca, em todas as direções uma única região é crescida por vez.

Algoritmo:

Inicialmente  $\text{região}(p) = 0$  para todo o ponto p da imagem.

1)  $k = 0$ ;

2) escolha um ponto x, tal que  $\text{região}(x) == 0$ , se não existe ponto nesta condição, então pare: fim;

3) inicie nova região:  $k = k + 1$ ;  $\text{região}(x) = k$ ;

4) ache todos os pontos p da vizinhança de x, tal que  $\text{região}(p) == 0$  e p possa ser adicionado à região de x sem violar o critério de homogeneidade: faça  $\text{região}(p) = \text{região}(x)$  e guarde estes pontos em um conjunto X;

5) se X está vazio, vá para 2; caso contrário, escolha e extraia um ponto x de X;

6) vá para 4;

Onde:

x é o ponto (x, y);

Os vizinhos de P é a sua vizinhança 4; região (p) indica a região do ponto p.

- **Crescimento de Regiões Segundo a Varredura.**

Neste método, a imagem é percorrida da esquerda para direita e de cima para baixo comparando um ponto inicial da imagem (x, y) com todos os outros pontos da imagem, adicionando ou não esses pontos a região do ponto inicial.

Algoritmo:

- 1) Percorra a imagem segundo a varredura;
- 2) Escolha um ponto x na imagem e o rotule;
- 3) Compare x com todos os outros pontos da imagem;
- 3.1) Adicione a região de x os pontos que não violarem o critério de homogeneidade;

### 3.2.4. Classificação de Imagem

A classificação é o processo de extração de informações da imagem. Consiste na divisão da imagem em classes, ou seja, segmentação e posterior identificação destas classes. Esta pode ser feita de forma não automática, onde um especialista humano extraí informações baseando-se na inspeção visual da imagem ou de forma automática, feita por computador [8]. A classificação de imagem por computador pode ser realizada usando técnicas de redes neurais. A rede é treinada de forma que ela possa identificar e agrupar todos os *pixels* em classes. O aprendizado da rede pode ser feito de duas formas: aprendizado supervisionado e aprendizado não supervisionado.

- Aprendizado supervisionado

Para [3] quando existem amostras disponíveis de classificação conhecida, define-se que o problema de aprendizado é supervisionado. A rede, tendo esta amostra conhecida inicialmente, tenta encontrar na imagem todos os pixels que podem ser agrupados a esta amostra conhecida, formando uma classe. Os pixels podem ser agrupados a uma mesma classe se por um critério de similaridade eles são iguais ou

semelhantes. Este critério é chamado de limiar, podendo ser a intensidade luminosa do pixel usada para decidir a igualdade ou semelhança.

- Aprendizado não supervisionado

No aprendizado sem supervisão as amostras na área de treinamento não são rotuladas [3]. Neste caso a rede tem que aprender sozinha quais pixels podem fazer parte de uma mesma classe, sem nenhum conhecimento inicial.

### **3.3. Processamento Digital de Imagem e o Projeto**

No projeto desenvolvido existem dois momentos distintos nos quais as técnicas de processamentos de imagem são utilizadas, são eles: Detecção da queda da primeira gota e acompanhamento do nível do líquido no interior do recipiente.

As etapas para o processamento da imagem nos dois momentos descritos acima apresentam semelhanças. São elas:

1. Aquisição da imagem por intermédio das câmeras de monitoramento.
2. Conversão das imagens de coloridas (RGB) para tons de cinza.
3. Conversão das imagens de tons de cinza para preto e branco (processo de binarização).
4. Comparação (subtração pixel a pixel) da imagem capturada com a imagem salva no momento em que o sistema foi calibrado. A calibragem é feita manualmente determinando a região da imagem a ser monitora, ou seja, faz-se a marcação da região retangular com auxílio das interfaces de calibragem do sistema (vide 2.2.6.).
5. Contagem da quantidade de valores que após a subtração ficaram acima de um limiar estipulado pelo operador do sistema.
6. Comparação percentual entre a quantidade acima descrita e o total de pixels no interior da região que está sendo monitorada; estando este percentual acima de um valor pré-determinado considera-se que a gota ou o nível de líquido foram detectados.

### 3.3.1. O Algoritmo

O sistema funciona em três estados possíveis. São eles: a) Pré-gota, correspondente ao período anterior à formação da gota; b) Formação-gota, referente ao período em que a primeira gota aparece nas imagens obtidas pelo sistema; e c) Pós-gota, período de repouso após o aparecimento da gota, indicando que a gota se desprendeu do bico. Sendo assim, cada quadro é analisado e ao mesmo é atribuído um destes três estados, baseado na classificação do mesmo como um quadro de movimento ou de repouso e na classificação do quadro anterior.

Para classificar os quadros como situações de movimento ou repouso, utilizou-se um procedimento simples de comparação de imagens, chamado `detecta_mov`. Este realiza a comparação *pixel* a *pixel* entre duas imagens, a imagem base (situação de repouso), denotada por  $F$ , e a imagem atual, denotada por  $I$ . Os parâmetros  $\theta$  e  $w$  representam, respectivamente, a diferença mínima entre o valor de dois *pixels* para que eles sejam considerados diferentes e o número mínimo de *pixels* diferentes em uma imagem para que a mesma corresponda à ocorrência de movimento. O procedimento retorna *verdadeiro* em caso positivo, e *falso* caso contrário.

**PROC** `detecta_mov` ( $I, F, \theta, w$ ): Boolean

`n=0`

**for**  $i = 0$  to  $ALTURA(I) - 1$  **do**

**for**  $j = 0$  to  $LARGURA(I) - 1$  **do**

**if**  $|(i; j) - F(i; j)| >= \theta$  **then**

`n=n+1`

**end if**

**end for**

**end for**

**if**  $n >= \theta$  **then**

**return** *verdadeiro*

**else**

**return** *falso*

**end if**

### ***3.3.2. A Detecção da Gota***

As etapas para a detecção da queda da primeira gota seguem como descritas no item anterior.

### ***3.3.3. Etapas do Projeto***

#### ***3.3.3.1. Aquisição da Imagem Real***

Nas Figuras 3.20 e 3.21 abaixo temos as imagens reais focalizadas no Bico e na Garrafa coletora.



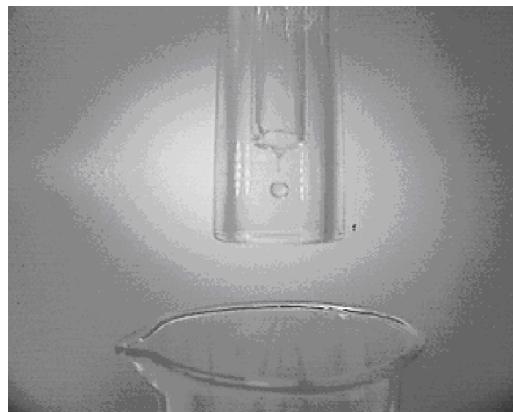
**Figura 3.20: Bico Real**



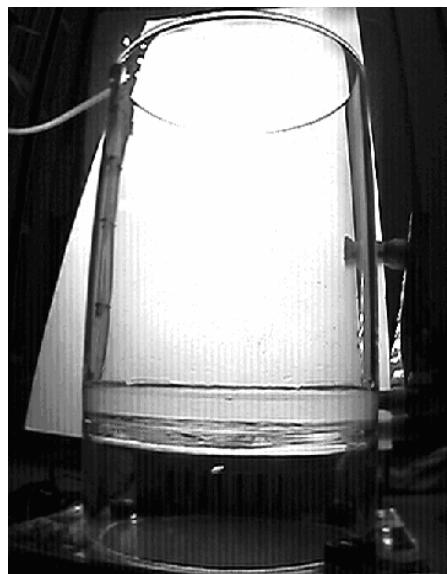
**Figura 3.21: Garrafa Real**

### **3.3.3.2. Imagem em Níveis de Cinza**

Nas Figuras 3.22 e 3.23 abaixo temos as imagens do Bico e da Garrafa convertidas para tom de cinza.



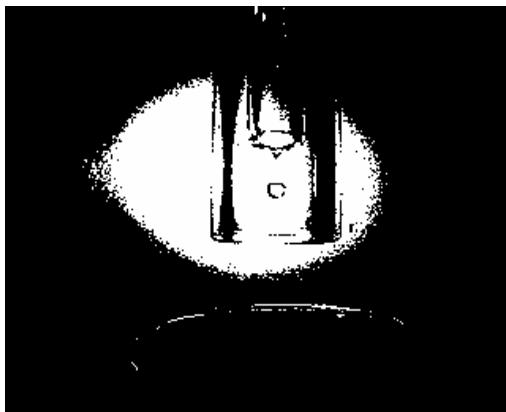
**Figura 3.22: Bico em Tom de Cinza**



**Figura 3.23: Garrafa em Tom de Cinza**

### **3.3.3.3. Imagem Binarizada (Limiarização ou *Thresholding*)**

Nas Figuras 3.24 e 3.25 abaixo temos as imagens do Bico e da Garrafa binarizadas.



**Figura 3.24: Bico Binarizado**



**Figura 3.25: Garrafa Binarizada**

# Capítulo 4

## Elaboração dos Circuitos

O projeto de montagem do painel de leds partiu da necessidade de uma iluminação que fosse suficiente tanto no aspecto do brilho quanto na área de cobertura da garrafa utilizada nos experimentos. Nas Figuras 4.1, 4.2 abaixo temos a representação do diagrama esquemático dos circuitos para o painel de leds do Menisco.

### 4.1. Circuito Teórico

**Linhas Impares:**

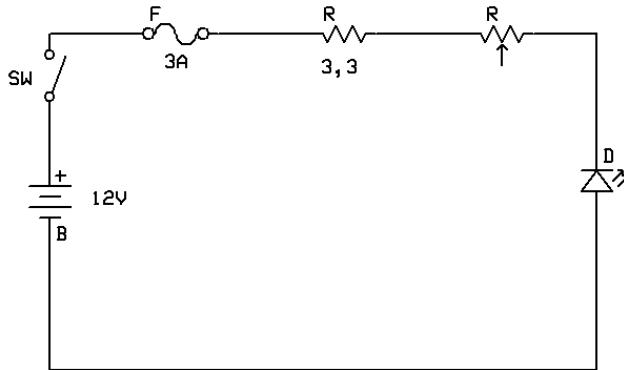


Figura 4.1: Circuito de Linhas Pares

**Linhas Pares:**

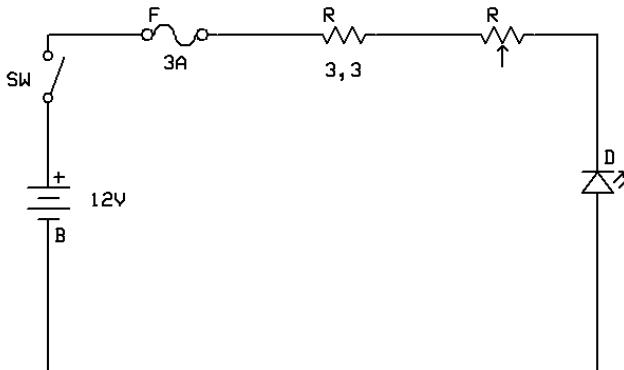


Figura 4.2: Circuito de Linhas Ímpares

Utilizando uma fonte de tensão de 12,0 V e consultando as especificações de corrente e queda de tensão nos leds utilizados no projeto, foi possível calcular a faixa de valores dos resistores, levando-se em conta que são 128 leds para cada circuito (totalizando 256). Para isso, divide-se a queda de tensão no resistor ( $12,0\text{ V} - 3,8\text{ V}$ )

pela corrente que passa no próprio ( $128 \times i$  mA). Onde  $i$  é a corrente recomendada pelo fabricante para cada led, que varia de 20,0 mA a 30,0 mA, obtendo uma faixa de valores para os resistores.

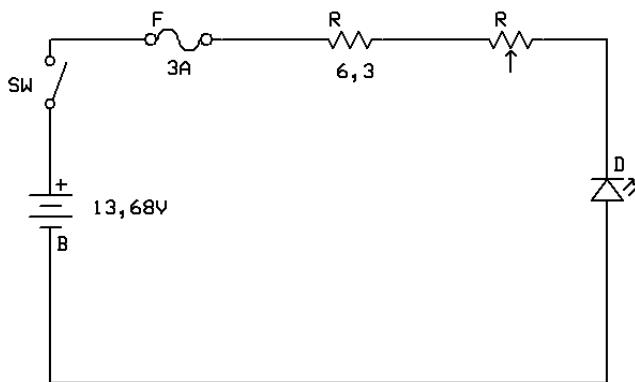
	Nº de Leds	I/Led(A)	Itotal(A)	Vfonte(V)	Vled(V)	R(ohm)	Ppot(W)>=
Linhas Pares:	128	0,02	2,56	12,0	3,8	3,203	20,992
Linhas Ímpares:	128	0,02	2,56	12,0	3,8	3,203	20,992

Onde:

- **I/Led(A)** é a corrente em cada led,
- **Itotal(A)** é a corrente total no circuito,
- **Vfonte(V)** é a tensão utilizada para alimentar o circuito,
- **Vled(V)** é a queda de tensão no conjunto de 128 leds,
- **R(ohm)** é o valor da resistência utilizado no projeto,
- **Ppot(W) >=** é o valor da potência mínima no potenciômetro.

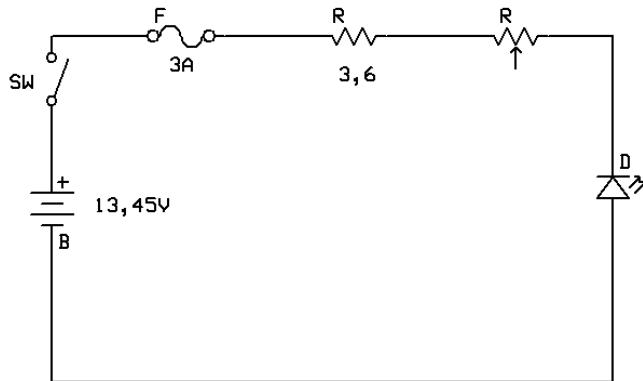
## 4.2. Circuito Prático

**Linhas Ímpares:**



**Figura 4.3: Circuito de Linhas Pares**

**Linhas Pares:**



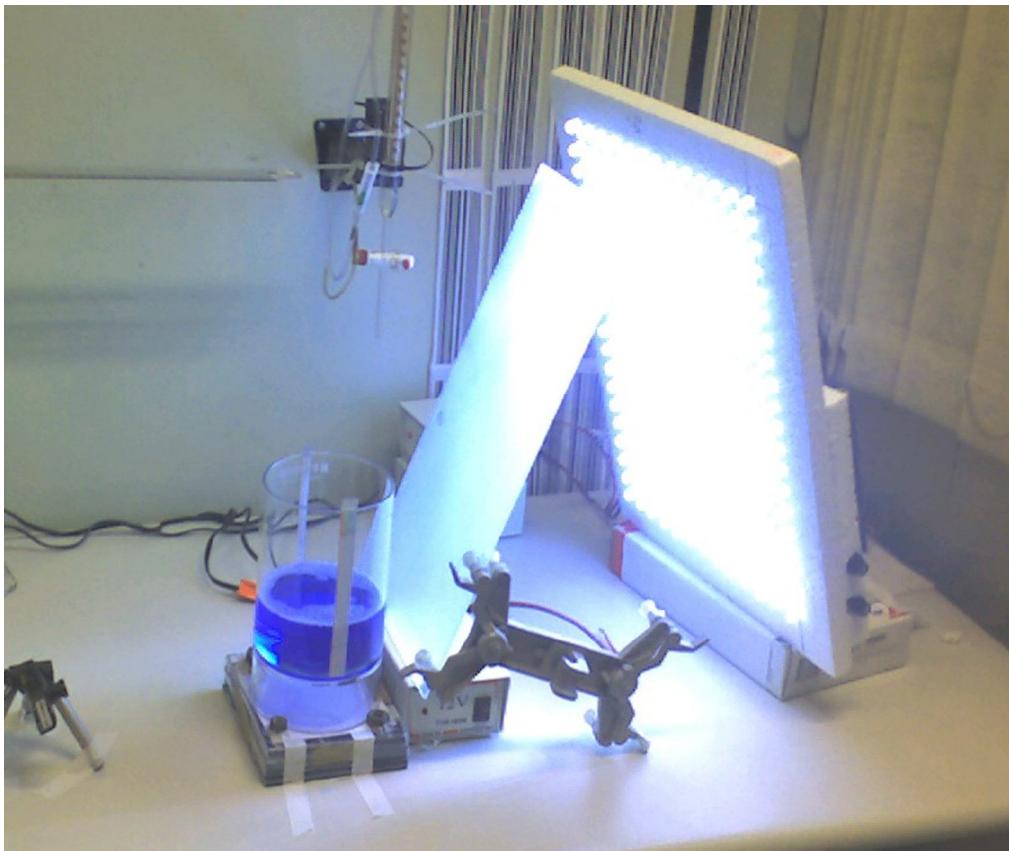
**Figura 4.4: Circuito de Linhas Ímpares**

Após a escolha dos resistores e a montagem do painel de leds, foi feita a verificação das variáveis calculadas na parte teórica ( $I/\text{Led}(A)$ ,  $I_{\text{total}}(A)$ ,  $V_{\text{fonte}}(V)$ ,  $V_{\text{led}}(V)$ ,  $P_{\text{pot}}(W) \geq$ ) utilizando multímetro. Os resultados obtidos estão na tabela abaixo:

	Nº de Leds	$I/\text{Led}(\text{mA})$	$I_{\text{total}}(\text{mA})$	$V_{\text{fonte}}(\text{V})$	$V_{\text{led}}(\text{V})$	$R(\text{ohm})$	$P_{\text{pot}}(\text{W}) \geq$
Linhas Pares:	128	0,020	2,583	13,45	4,15	3,6	24,025
Linhas Ímpares:	128	0,012	1,507	13,68	4,18	6,3	14,325

**Observação:** devido à dificuldade de obter resistores nesses valores que dissipem esta potência, foram utilizados resistores diferentes para os circuitos, como visto nas Figuras 4.3 e 4.4 acima.

Na Figura 4.5 abaixo temos em destaque o painel de leds usado para iluminar o Menisco.



**Figura 4.5: Painel de Leds**

A caixa da Figura 4.6 abaixo abriga o Circuito Atuador. Este circuito é responsável pelo acionamento das câmeras e iluminações do Sistema. O diagrama esquemático deste circuito encontra-se na Figura A.1 do Apêndice A.



**Figura 4.6: Circuito Atuador**

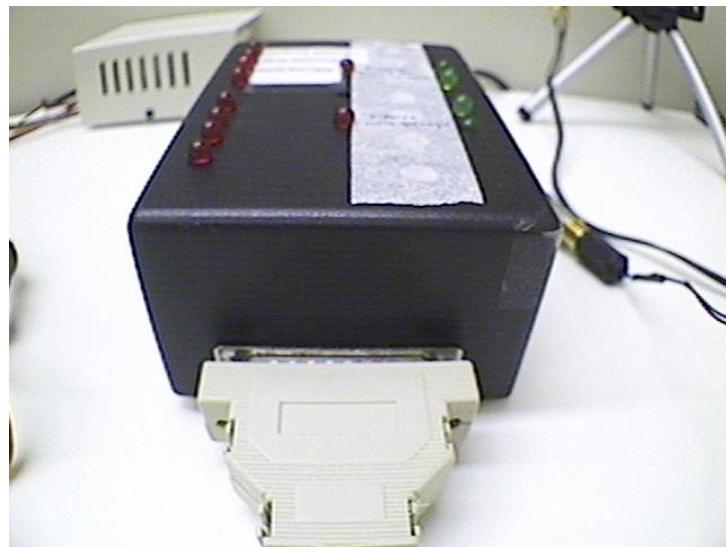
As Figuras 4.7, 4.8, 4.9 e 4.10 abaixo, correspondem à montagem da caixinha contendo o Circuito de Teste. Este circuito teve a função de auxiliar no desenvolvimento da parte de Software do Sistema. Todas as entradas e saídas do sistema da porta paralela eram simuladas por este circuito. Na Figura A.3 do Apêndice A temos o diagrama esquemático do Circuito de Teste.



**Figura 4.7: Circuito de Teste Vista Superior**



**Figura 4.8:** Circuito de Teste vista traseira



**Figura 4.9:** Circuito de Teste Vista Frontal

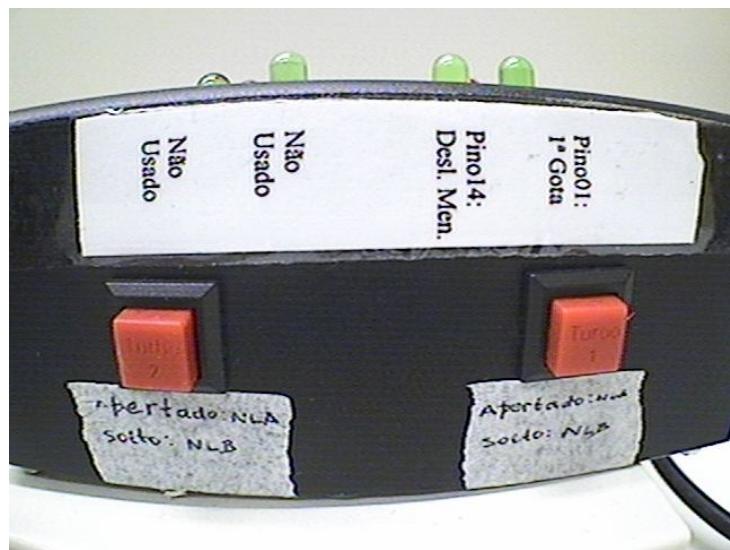


Figura 4.10: Circuito de Teste Vista Lateral

# Capítulo 5

## Conclusão

Em busca do desenvolvimento do projeto tive a oportunidade de aprender com maiores detalhes sobre a área de processamento de imagem. Foram pesquisadas as técnicas mais usadas, os algoritmos implementados, as revistas e congressos sobre o assunto e também as mais diversas aplicações para este campo de pesquisa.

Como exemplo de áreas onde podemos encontrar o processamento digital de imagem temos: identificação de pessoas pela face, pela íris, pelas digitais, Ultrasonografia 3D.

O objetivo deste projeto era fazer o uso de técnicas de processamento de imagem para detecção do início do processo de destilação de um subproduto do petróleo e também desenvolver um sistema para testar os periféricos (hardwares) que compõem o sistema de monitoramento. O início do processo de destilação é determinado com a queda da primeira gota do subproduto destilado através do bico do destilador, e para tal detecção implementou-se um algoritmo chamado detecta\_mov.

Como resultados foi possível chegar a um algoritmo robusto no que tange os testes de hardwares necessários, contudo com relação à detecção da queda da primeira gota o algoritmo ainda apresenta dependências da configuração do ambiente de laboratório, tal como qualidade de iluminação, resíduos de materiais entorno do bico do destilador, reflexos no vidro do destilador.

O uso do processamento de imagem no projeto mostrou-se uma ferramenta poderosa possibilitando inúmeras ramificações de idéias, o desenvolvimento de sistemas a custos menores e certamente foi de grande valia ter adquirido este conhecimento.

# Apêndice A

## A.1. Diagrama do Circuito Atuador

O circuito da Figura A.1 (CA) abaixo, tem a função de acionar as câmeras e iluminações utilizadas no sistema. Este circuito recebe como entradas os sinais da porta paralela do computador onde o Sistema está rodando. O CI 74LS541 tem a função de isolar a porta paralela do circuito.

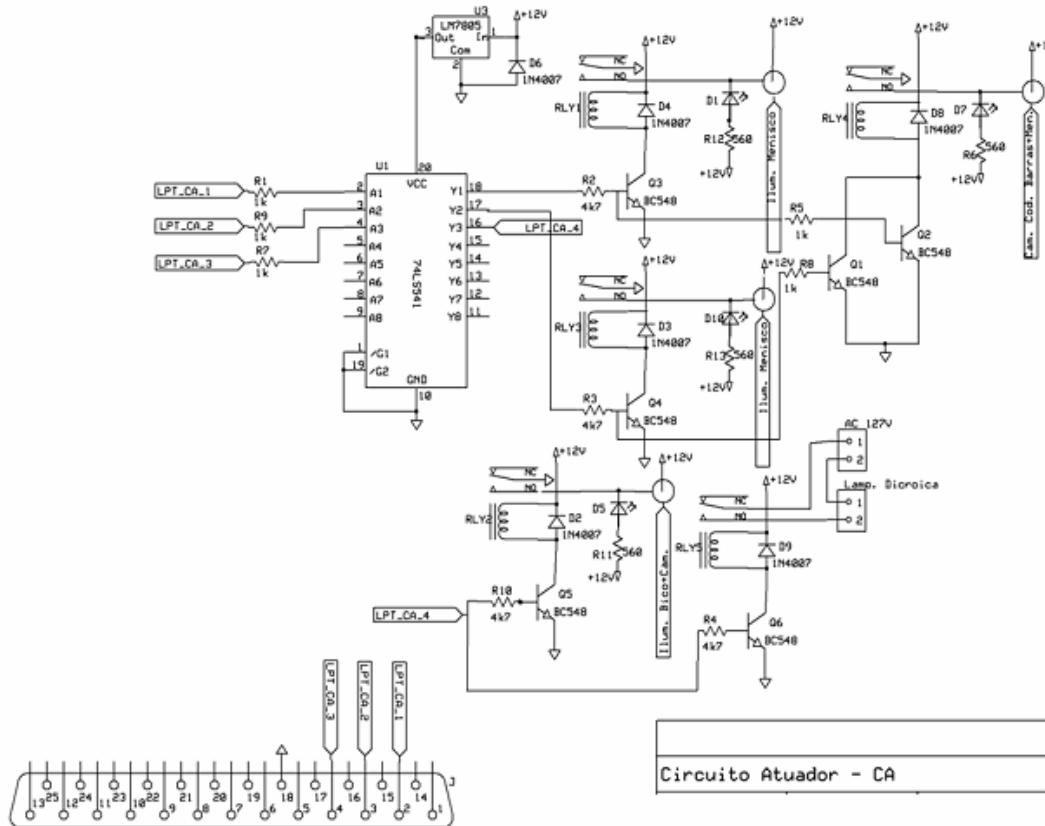


Figura A.1: Diagrama Esquemático do CA

## A.2. Diagrama do Circuito Atuador

O circuito da Figura A.2 (CAT) abaixo, tem a função de realizar o casamento dos níveis lógicos da porta paralela do computador, TTL (0V/5V), com o do CLP, 0V/24V. O CI 74LS542 tem a função de isolar a porta paralela do circuito.

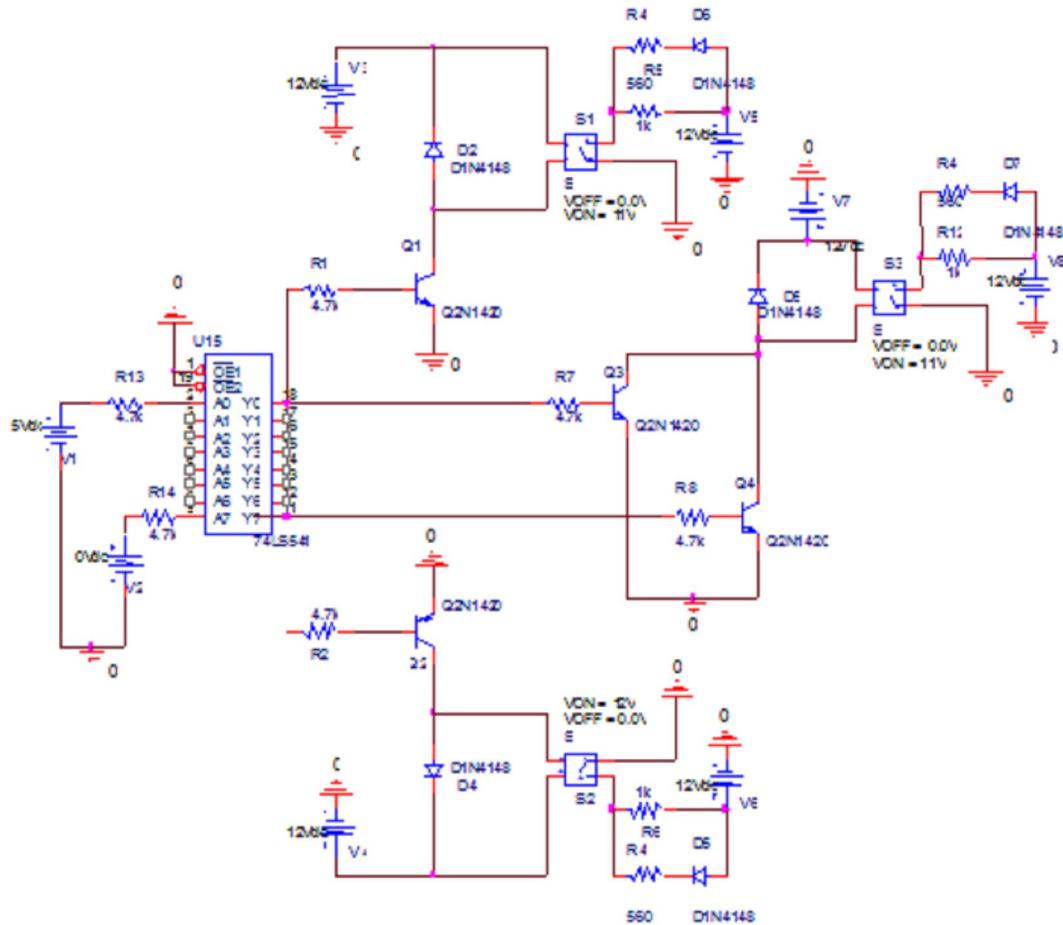


Figura A.2: Circuito CAT Simulado no ORCAD

### A.3. Diagrama do Circuito de Teste

O circuito da Figura A.3 abaixo, teve a função de auxiliar no desenvolvimento da parte de Software do Sistema. Com este circuito foi possível simular todas as entradas e saídas do sistema, via porta paralela. Como nos outros circuitos, foi utilizado um *buffer* para isolar a porta paralela do circuito.

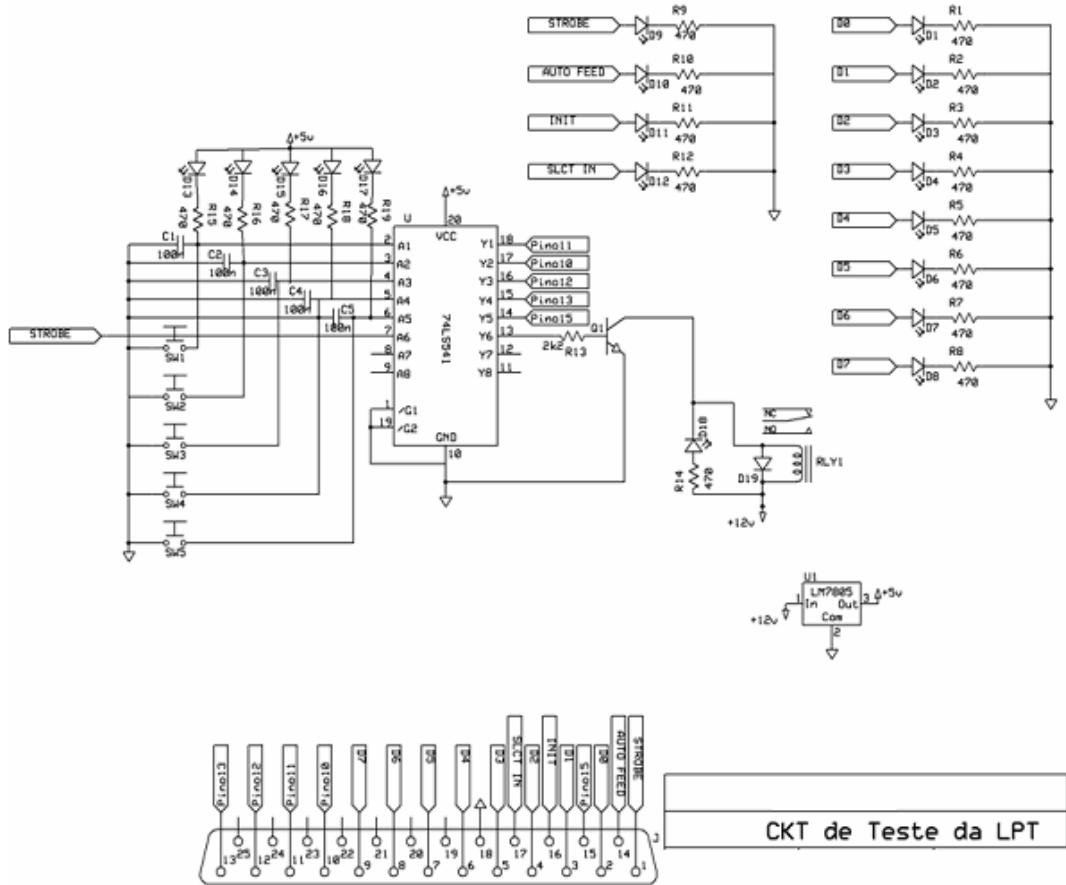


Figura A.3: Circuito de Teste

# Bibliografia

- [1] GOMES,F M. *Sistema de Monitoramento do Nível Baseado em Processamento Digital de Imagens*, Universidade Federal do Rio de Janeiro – UFRJ. Março de 2009.
- [2] BANON, G. J. F. *Formal Introduction to digital image processing*. INPE, São José dos Campos JULHO 2000.
- [3] MASCARENHAS, N. A. & VELASCO, F. R.D. *Processamento Digital de Imagens*. Ministerio da ciência e Tecnologia -MCT. Instituto de Pesquisa Espaciais - INPE. Janeiro de 1989.
- [4] BRITO, S. F. & CARVALHO, J. M. *Sistemas de Processamento digital de imagens para Fins didáticos/científicos: Estudo, seleção e implementação de algoritmos de segmentação*. Relatorio de iniciação científica. Agosto de 1998.
- [5] SCHNEIDER, B. O. *Apostila de Computação Gráfica*. Departamento de Ciência da Computação. Universidade Federal de Lavras. 9 de Agosto de 2001.
- [6] RAMOS, O. T. H. *Análise Comparativa entre os principais formatos de armazenamento de imagens*. Dissertação de Mestrado pela Universidade Presbiteriana Mackenzie. São Paulo, dezembro de 2000.
- [7] GONZALES, R. C. &WOODS, R. E. *Digital Image Processing*. University of Tennessee Perceptics Corporation, 1992.
- [8] MARIA, L. G. F. *Processamento Digital de Imagens*. INPE, Junho de 2000.
- [9] MARTA. D. S. *Algoritmos para detecção de bordas*. UFSC - dezembro de 1998.
- [10] FACON, J. *Princípios Básicos da Visão por Computador e do Processamento de Imagens*. Pontificia Universidade Católica do Paraná. Curso e Mestrado em Informatica Aplicada, 2002.