

UNIVERSIDADE FEDERAL DE SANTA CATARINA

Marcelo de Barros Leiras

**SISTEMA AUTOMATIZADO DE MONITORAMENTO E CONTROLE REMOTO DE
VARIÁVEIS HÍDRICAS**

Florianópolis

2004

Marcelo de Barros Leiras

**SISTEMA AUTOMATIZADO DE MONITORAMENTO E CONTROLE REMOTO DE
VARIÁVEIS HÍDRICAS**

Trabalho de conclusão do Curso de Bacharelado em
Ciências da Computação, da Universidade Federal de Santa
Catarina.

ORIENTADORA: Prof. Dr. Lúcia Helena Martins Pacheco

Florianópolis

2004

Marcelo de Barros Leiras

**SISTEMA AUTOMATIZADO DE MONITORAMENTO E CONTROLE REMOTO DE
VARIÁVEIS HÍDRICAS**

BANCA EXAMINADORA

Lúcia Helena Martins Pacheco – Orientadora

Roberto Willrich – Membro

Vitório Bruno Mazzola – Membro

Florianópolis

Julho de 2004

RESUMO

Nos últimos anos, o cultivo de camarões vem sendo bastante incentivado na região sul do Brasil. Um dos grandes responsáveis pelo desenvolvimento deste setor é o Laboratório de Camarões Marinhos (LCM), localizado nas dependências da UFSC, cuja principal atividade é o fornecimento de pós-larvas de camarões para fazendas de engorda.

Devido à grande sensibilidade dos camarões a variações bruscas de certos parâmetros da água como temperatura, pH, oxigênio dissolvido, amônia, entre outros, é de extrema importância que estas variáveis sejam monitoradas constantemente para que ações de correção das mesmas possam ser realizadas em tempo hábil, evitando que estes indivíduos sejam prejudicados.

Em vista disso, diariamente são realizados no LCM vários testes com reagentes químicos que utilizam uma pequena amostra de água do local analisado. Porém, a utilização em larga escala destes kits de testes é responsável por uma grande parcela nas despesas do produtor. Além disso, é necessário ainda manter um funcionário encarregado de fazer as medições periodicamente, despendendo um tempo precioso que poderia estar sendo gasto em outras atividades.

Outro grave problema deste tipo de monitoramento da qualidade da água é o tempo decorrido até que se tome alguma atitude em resposta às variações dos parâmetros. Sem um controle em tempo real, qualquer incidente ocorrido no intervalo entre duas medições capaz de alterar os parâmetros da água só vai ser percebido na próxima análise. Então, poderá ser tarde demais para uma ação corretiva, ocasionando talvez até a morte dos indivíduos.

O propósito da realização deste trabalho é apresentar uma solução para alguns problemas inerentes no ciclo produtivo do LCM através de um sistema automatizado de monitoramento e controle remoto da água dos tanques de camarões. Em contrapartida, o produto final resultado deste estudo pode ser utilizado para praticamente qualquer finalidade em que seja necessário o monitoramento da qualidade da água.

Palavras-chave: Variáveis Hídricas, Automação, Controle, Monitoramento, Acesso Remoto, Camarões.

ABSTRACT

In the last years, the culture of shrimps has been sufficiently encouraged in the South region of Brazil. One of the great responsible for the development of this sector is the Laboratório de Camarões Marinhos (Laboratory of Marine Shrimps) - LCM, belonging to the Universidade Federal de Santa Catarina (Federal University of Santa Catarina) - UFSC, whose main activity is the supply of after-larvae of shrimps for fattening farms.

Due to the great sensibility of shrimps to abrupt variations of certain water parameters as temperature, pH, dissolved oxygen, ammonia, among others, it is extremely important that these variables are constantly monitored. Then adjustments to correct of those parameters have to be carried through in the right time, preventing that these individuals are harmed.

In view of this, some tests are daily carried out with chemical reagents that use a small sample of water of the analyzed place. However, the use in wide scale of these kits of tests accounts for a meaningful part of the producer's expenses. Moreover, it is necessary to keep an employee in charge making periodical measurements.

Another serious problem of this type of monitoring of water quality is the delay until someone takes some attitude in reply to the variations of the parameters. Without a real time control, any incident that is able to modify the water parameters, which occurs in the interval between two measurements, would be detected just in the next analysis. Then, it could be excessively late for a corrective action, perhaps causing death of some individuals.

This work aims to present a solution for some inherent problems in the productive cycle of the LCM. For that, it is proposed an automatic system of remote monitoring and control of the water of the tanks of shrimps. Also such system could be used for similar applications where the monitoring of water quality is necessary.

Key-words: Water Variables, Automation, Control, Monitoring, Remote Access, Shrimps.

LISTA DE FIGURAS

Figura 1 – Metodologia de desenvolvimento do trabalho	16
Figura 2 – Máquina von Neumann (RICARTE, 1999)	20
Figura 3 – Máquina Havard (RICARTE, 1999).....	20
Figura 4 – Diagrama de Blocos do PIC 16F870 (MICROCHIP, 2004).....	21
Figura 5 – Mapeamento da memória do PIC16F870 (MICROCHIP, 2004)	23
Figura 6 – Registradores do PIC 16F870 (MICROCHIP, 2004).....	25
Figura 7 – Estrutura geral de um sistema de banco de dados (RAPCHAN, 2004).....	32
Figura 8 – Dinâmica de uma requisição a uma página JSP (DBSystems, 2004).....	38
Figura 9 – Comportamento termo-sensível dos RTDs e Termistores (GÜTHS, 1998)	43
Figura 10 – Comportamento termo-sensível do PT100 (DANTAS, 2004).....	44
Figura 11 – Ponte de Wheatstone (DANTAS, 2004).....	45
Figura 12 – RTDs de 2 e 3 fios (DANTAS, 2004).....	46
Figura 13 – Escala de pH (ROCHA, 2004).....	49
Figura 14 – Eletrodo de vidro (OLIVEIRA, 2004)	50
Figura 15 – Variação na medida do pH em função da temperatura.....	51
Figura 16 – Eletrodo Combinado (OLIVEIRA, 2004).....	52
Figura 17 – Visão geral do trabalho.....	53
Figura 18 – Modelagem do Banco de Dados	54
Figura 19 – Tipos de Alertas.....	55
Figura 20 – Rotina de funcionamento do Gerente.....	56
Figura 21 – Configurações	58
Figura 22 – Filtragem de informações para a consulta	59
Figura 23 – Visualização das informações solicitadas	59
Figura 24 – Controle	60
Figura 25 – Circuito de funcionamento básico do PIC	61
Figura 26 – Rotina Principal do PIC	62
Figura 27 – Rotina de Interrupção do PIC	63
Figura 28 – Circuito de aquisição de temperatura.....	64
Figura 29 – Circuito de Acionamento de Dispositivos	65

LISTA DE SIGLAS E ABREVIATURAS

A/D - Analog/Digital

ADCON - A/D Control Register

ADRESH - A/D Result Register High Byte

ADRESL - A/D Result Register Low Byte

ASP - Active Server Pages

CGI - Common Gateway Interface

CISC - Complex Instruction Set Computer

D/A - Digital/Analog

DDL - Data Definition Language

DHTML - Dynamic HTML

DML - Data manipulation Language

EEADR - EEPROM Address Register

EEDATA - EEPROM Data Register

EEPROM - Electrically Erasable Programmable Read-Only Memory

EPAGRI - Empresa de Pesquisa Agropecuária e Extensão Rural do Estado de Santa Catarina

EPROM - Erasable Programmable Read-Only Memory

E/S - Entrada/Saída

GUI - Grafical User Interface

HTML - Hyper Text Markup Language

HTTP - Hyper Text Transfer Protocol

JDBC - Java Data Base Connectivity

JSP - Java Server Pages

LCM - Laboratório de Camarões Marinhos

PC - Personal Computer

PIC - Peripheral Interface Controller

RAM - Random Access Memory

RISC - Reduced Instruction Set Computer

RTD - Resistance Temperature Detector

SGBD - Sistema Gerenciador de Banco de Dados

SFR - Special Function Register

SQL - Structured Query Language

UFSC - Universidade Federal de Santa Catarina

URL - *Universal Resource Location*

USART - *Universal Synchronous Asynchronous Receiver Transmitter*

SUMÁRIO

1 INTRODUÇÃO	10
2 OBJETIVOS	14
2.1 Objetivo geral	14
2.2 Objetivos específicos	14
3 METODOLOGIA	15
4 FUNDAMENTAÇÃO TEÓRICA	17
4.1 Microcontroladores	17
4.1.2 Microcontroladores x Microprocessadores	18
4.1.3 Arquitetura	19
4.1.5 Organização da Memória	22
4.1.5.1 Memória de Programa	22
4.1.5.2 Memória de Dados	24
4.1.5.3 EEPROM	26
4.1.6 Portas de E/S	26
4.1.7 Temporizadores	27
4.1.8 Interrupções	27
4.1.9 Conversores A/D	28
4.1.10 USART	29
4.2 Banco de Dados	30
4.2.1 Visão Geral	30
4.2.2 Modelo Relacional	33
4.2.3 MySQL	33
4.3 Sistema via Internet	34
4.3.1 Java	34
4.3.2 Applets	35
4.3.3 Java Beans	36
4.3.4 Servlets	37
4.3.5 JSP	37
4.4 Módulo de Aquisição de Variáveis	40
4.4.1 Temperatura	40
4.4.2 Sensores Termorresistivos	42
4.4.2.1 RTD	43

4.4.2.2 Termistores.....	47
4.4.3 pH.....	48
4.4.4 Sensores de pH.....	50
5 DESENVOLVIMENTO	53
5.1 Host	53
5.1.2 Estrutura do Banco de Dados	54
5.1.3 Gerente	55
5.1.4 Sistema de Acesso Remoto	57
5.1.4.1 Configurações.....	57
5.1.4.2 Históricos.....	58
5.1.4.3 Gráficos	60
5.1.4.4 Controle	60
5.2 Target	61
5.2.1 PIC	61
5.2.1.1 Circuito Básico de Funcionamento	61
5.2.1.2 Rotina principal	62
5.2.1.3 Rotina de Interrupção	63
5.2.2 Leitura de Variáveis.....	63
5.2.3 Acionamento de Dispositivos	64
6 ESTADO ATUAL DO SISTEMA	66
7 CONCLUSÃO.....	68
8 REFERÊNCIAS BIBLIOGRÁFICAS.....	70
APÊNDICE	72

1 INTRODUÇÃO

Nos últimos anos, o cultivo de camarões vem sendo bastante incentivado na região sul do Brasil. Um dos grandes responsáveis pelo desenvolvimento deste setor é o Laboratório de Camarões Marinhos (LCM), localizado nas dependências da UFSC, cuja principal atividade é o fornecimento de pós-larvas de camarões para fazendas de engorda.

Em 1984, a UFSC iniciou pesquisas com reprodução e cultivo de espécies nativas. Durante 17 anos dedicou-se ao desenvolvimento de tecnologia para reprodução e cultivo destas espécies que, apesar dos ótimos resultados na reprodução, em escala comercial não foram competitivos nas fazendas de produção. Durante esse período, grande parte do potencial do laboratório foi usado para programas sociais, através do repovoamento de Lagoas Costeiras (LCM, 2004).

Desde a sua criação, o LCM estabeleceu estreita relação com o setor produtivo e, além da pesquisa e do ensino, se dedicou a apoiar a indústria do cultivo com o fornecimento de pós-larvas para as unidades de produção. Isso acabou determinando sua auto-suficiência econômica.

No segundo semestre de 1998, a UFSC e a Empresa de Pesquisa Agropecuária e Extensão Rural do Estado de Santa Catarina (EPAGRI) foram responsáveis pela introdução da espécie *Litopenaeus vannamei* nas fazendas existentes no Estado. Em razão do excelente desempenho nas fazendas de cultivo, a introdução desta espécie exigiu do LCM a ampliação da produção, com capacidade instalada para produzir 60 milhões de pós-larvas por mês.

Porém, com o rápido crescimento do laboratório, verificou-se que o gerenciamento dos ciclos produtivos era falho, e deveria ser remodelado. Sendo

assim, ao longo de 2003 foi implantado no LCM um sistema informatizado de gerenciamento do processo produtivo. O próximo passo é a automatização do monitoramento e controle da água dos tanques de cultivo.

Devido à grande sensibilidade dos camarões a variações bruscas de certos parâmetros da água como temperatura, pH, oxigênio dissolvido, amônia, entre outros, é de extrema importância que estas variáveis sejam monitoradas constantemente para que ações de correção das mesmas possam ser realizadas em tempo hábil, evitando que estes indivíduos sejam prejudicados.

Em vista disso, diariamente são realizados vários testes com reagentes químicos que utilizam uma pequena amostra de água do local analisado. Porém, a utilização em larga escala destes kits de testes é responsável por uma grande parcela nas despesas do produtor. Além do custo dos kits, é necessário ainda manter um funcionário encarregado de fazer as medições periodicamente, despendendo um tempo precioso que poderia estar sendo gasto em outras atividades.

Outro grave problema deste tipo de monitoramento da qualidade da água é o tempo decorrido até que se tome alguma atitude em resposta às variações dos parâmetros. Sem um controle em tempo real, qualquer incidente ocorrido no intervalo entre duas medições capaz de alterar os parâmetros da água só vai ser percebido na próxima análise. Então, poderá ser tarde demais para uma ação corretiva, ocasionando talvez até a morte dos indivíduos.

Numa tentativa de solucionar estas questões, alguns produtores recorrem a aparelhos eletrônicos disponíveis no mercado que fazem a leitura de certos parâmetros da água através de sensores. Porém, além destes dispositivos

apresentarem um valor muito elevado (a maioria é importado), eles não eliminam o custo de um empregado para fazer as medições.

Embora a criação seja monitorada 24 horas por dia, seja por funcionários encarregados da alimentação ou até mesmo do acompanhamento da qualidade da água, não é sempre que o pessoal mais qualificado pode estar presente, entre outros motivos, devido à própria carga horária de trabalho. Sendo assim, a verificação dos parâmetros da água e possíveis ações de correção emergenciais dos mesmos certas vezes acabam sendo realizadas por funcionários sem a devida formação, podendo prejudicar o cultivo. Em vista disso, o monitoramento remoto da criação torna-se interessante.

O propósito da realização deste trabalho é, primeiramente, tentar solucionar alguns problemas inerentes no ciclo produtivo do LCM. Em contrapartida, o produto final resultado deste estudo pode ser aplicado em inúmeras outras áreas, como na criação de outros animais aquáticos (peixes, mariscos, ostras, etc.), no controle da qualidade da água de rios, piscinas, caixas d'água, enfim, pode ser utilizado para praticamente qualquer finalidade em que seja necessário monitoramento da qualidade da água.

No caso do LCM, com certeza o problema mais crítico está nos gastos excessivos com kits de testes para a análise da água, sem falar no tempo despendido por funcionários para a realização dos testes. A implantação de um sistema automatizado de medição baseado em sensores é uma solução vantajosa a longo prazo, já que os custos de instalação e aparelhagem seriam compensados ao longo do tempo pela fatia do orçamento antes destinada à compra dos kits. Além disso, não haveria mais a necessidade de um funcionário dedicado a fazer as

medições, ficando por conta do sistema o registro periódico automático destas informações.

Outra vantagem da implantação de um sistema automatizado é a diminuição do tempo entre a percepção das variações nos parâmetros hídricos e as possíveis ações corretivas. O microcontrolador pode ser configurado para ligar (ou desligar) aquecedores e bombas de gás carbônico caso os valores de temperatura e pH atinjam certos limites pré-definidos, ou simplesmente indicar através de luzes e/ou alarmes a irregularidade constatada, deixando a cargo de um funcionário capacitado a tomada da devida ação corretiva. Este é um ponto muito importante, uma vez que quanto menor for o tempo de correção destes níveis indesejáveis, menores serão os danos à criação.

Também, a possibilidade de monitorar e controlar remotamente a situação dos tanques via Internet daria muito mais segurança aos responsáveis pelos setores produtivos, permitindo que os mesmos acompanhem o andamento da produção onde quer que eles estejam.

Então, este documento apresenta os principais aspectos técnicos, metodológicos e práticos envolvidos na experiência piloto de automatização do sistema de monitoramento e controle da água dos tanques de criação do Laboratório de Camarões Marinhos.

2 OBJETIVOS

2.1 Objetivo geral

Desenvolver um sistema automatizado de monitoramento e controle remoto de variáveis hídricas.

2.2 Objetivos específicos

- Realizar um estudo aprofundado sobre os microcontroladores PIC (*Peripheral Interface Controller*);
- Definir um protocolo de comunicação entre o PIC e o PC (*Personal Computer*);
- Projetar um banco de dados para o armazenamento dos valores coletados;
- Desenvolver uma ferramenta de software que possibilite a análise e o controle das variáveis em estudo, de uma forma remota;
- Elaborar um circuito eletrônico que conecte os sensores ao microcontrolador;
- Testar e Avaliar o sistema desenvolvido.

3 METODOLOGIA

Inicialmente foi realizado um estudo aprofundado sobre os microcontroladores PIC, da empresa *Microchip*, analisando as características de cada uma das famílias de PICs disponíveis no mercado para se fazer a escolha do modelo mais adequado ao projeto. Buscou-se um microcontrolador que possuísse no mínimo uma interface de comunicação serial, possibilitando a transferência de dados para o PC, além de conversores A/D, para a leitura dos sensores de temperatura e pH. Porém, caso houvesse indisponibilidade de algum destes requisitos, o problema seria contornado via *software* ou até mesmo com o auxílio de dispositivos eletrônicos externos.

A etapa seguinte foi o desenvolvimento de um protocolo de comunicação entre o PIC e o PC, para que os dados coletados pelos sensores pudessem ser transmitidos de maneira organizada para o PC e armazenados no banco de dados. Foi definido um padrão de transferência, estabelecendo bits de controle, ordem de envio de variáveis (pH e temperatura), número de bits de cada parâmetro, entre outros.

Uma vez definido este protocolo, foi desenvolvido um banco de dados para o armazenamento das informações adquiridas pelos sensores, permitindo que as mesmas pudessem ser consultadas futuramente.

Visando tornar a manipulação dos dados mais amigável ao usuário, foi elaborado um *software* através do qual poderão ser consultadas todas as informações contidas no banco de dados, por meio de tabelas e gráficos, bastando ter acesso a um computador conectado à Internet. Pela interface gráfica do programa o usuário pode ainda acionar remotamente dispositivos eletrônicos como aquecedores e bombas de gás carbônico (CO₂), caso seja necessária alguma

correção emergencial nos parâmetros de qualidade da água, além de verificar os valores dos mesmos em tempo real. Para tornar isto possível, foi criado um circuito eletrônico que faz a “ponte” entre os sensores e o PIC, transformando os dados de temperatura e pH em níveis de tensão, compreensíveis pelos conversores A/D do microcontrolador.

O módulo construído foi fixado num tanque de camarões de um dos setores produtivos do LCM, e conectado a um computador com acesso à Internet. Os parâmetros da qualidade da água foram então acompanhados durante um determinado intervalo de tempo e, ao final, foi feita uma avaliação geral do sistema.

A Figura 1 esquematiza a metodologia adotada.

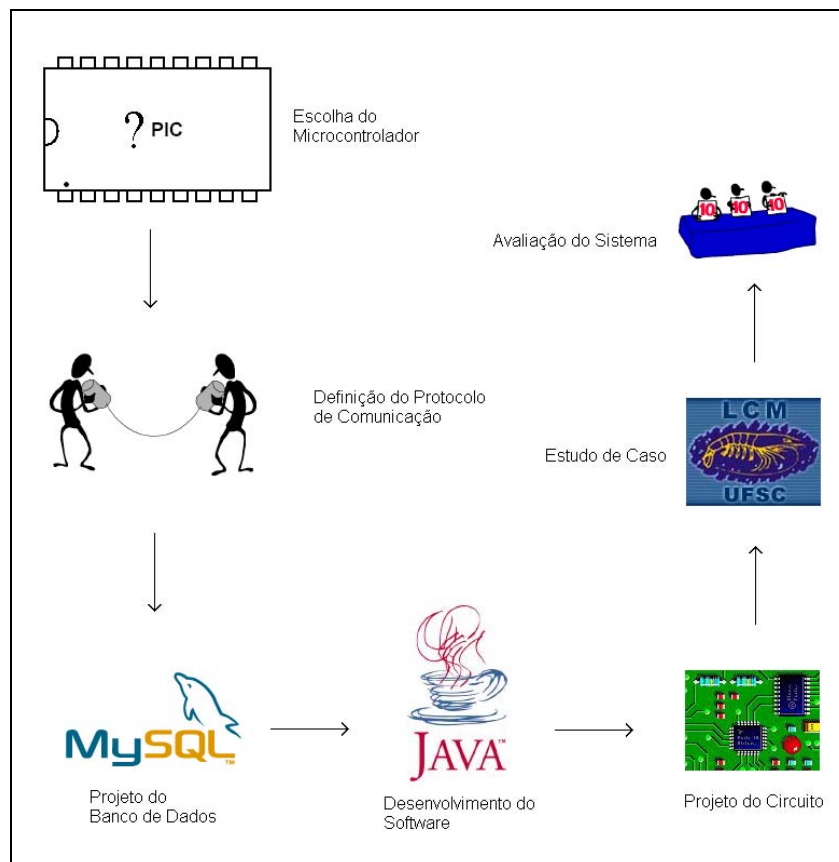


Figura 1 – Metodologia de desenvolvimento do trabalho

4 FUNDAMENTAÇÃO TEÓRICA

Aqui serão abordados alguns conceitos básicos referentes aos diferentes módulos que compõem o trabalho desenvolvido, visando facilitar a compreensão dos mesmos.

4.1 Microcontroladores

SOUZA (2003) define o microcontrolador como sendo “um pequeno componente eletrônico, dotado de uma inteligência programável, utilizado no controle de processos lógicos”.

O controle de processos deve ser entendido como o controle de periféricos, tais como: LEDs, botões, displays de segmentos, displays de cristal líquido, resistências, relês, sensores, e muitos outros.

São chamados controles lógicos, pois a operação do sistema baseia-se nas ações lógicas que devem ser executadas, dependendo do estado dos periféricos de entrada e/ou saída.

O microcontrolador é programável, pois toda a lógica de operação é estruturada na forma de um programa e gravada numa memória dentro do componente. Os PICs são uma categoria de microcontroladores e, na maioria deles, a memória interna é do tipo EPROM (*Erasable Programmable Read-Only Memory*). Tais tipos de memória permitem serem gravadas uma vez para PICs normais, ou várias vezes no caso de PICs janelados (que podem ser apagados mediante banho de luz ultravioleta, efetuado através da exposição da pastilha por uma janela

existente em seu encapsulamento). Existem ainda modelos que possuem a memória de programa do tipo FLASH, que pode ser gravada várias vezes.

Depois de gravado o programa operacional na memória interna, toda vez que o microcontrolador for ligado, o programa interno será executado.

Quanto à inteligência do componente, podemos associá-la à ALU (*Arithmetic Logic Unit*), pois é nessa unidade que todas as operações matemáticas e lógicas são executadas. Quanto mais poderosa for a ALU, maior será a capacidade de processar informações.

E finalmente, o microcontrolador ganhou ainda o adjetivo “pequeno”, pois em uma única pastilha de silício temos todos os componentes necessários ao controle de um processo.

A finalidade dos tópicos seguintes é dar uma visão geral sobre o funcionamento destes componentes, detalhando seus módulos e funcionalidades, sempre focando o PIC16F870 da empresa *Microchip*, que foi microcontrolador adotado neste trabalho.

4.1.2 Microcontroladores x Microprocessadores

Parafraseando MATIC (2004), um microcontrolador difere de um microprocessador em vários aspectos, mas principalmente devido a sua funcionalidade. Para que um microprocessador possa ser usado, outros componentes devem-lhe ser adicionados, tais como memória e componentes para receber e enviar dados. Em resumo, isso significa que o microprocessador é o verdadeiro coração do computador. Por outro lado, o microcontrolador foi projetado para reunir todos estes dispositivos numa única pastilha de silício, acabando com a

necessidade de outros componentes externos nas aplicações, uma vez que todos os periféricos necessários já estão contidos nele. Desta maneira, poupa-se tempo e espaço.

Sendo assim, os microcontroladores são indicados para aplicações dedicadas, onde haja necessidade da realização de tarefas específicas como, por exemplo, o monitoramento e controle variáveis.

4.1.3 Arquitetura

Com base no relacionamento e organização dos componentes de um sistema computacional, estes podem ser classificados entre máquinas von Neumann e máquinas não-von Neumann.

As primeiras caracterizam-se por um sistema composto a partir de três subsistemas básicos: CPU, memória principal e sistema de entrada e saída.

A CPU, por sua vez, tem três blocos principais: unidade de controle (UC), unidade lógico-aritmética (ALU) e registradores, incluindo-se aí um registrador contador de programa (PC) que indica a posição da instrução a ser executada.

São características das máquinas von Neumann a utilização do conceito de programa armazenado, a execução seqüencial de instruções e a existência de um caminho único (barramento) entre memória e unidade de controle (RICARTE, 1999), como ilustra a Figura 2. Sendo assim, dados e instruções compartilham o mesmo barramento.

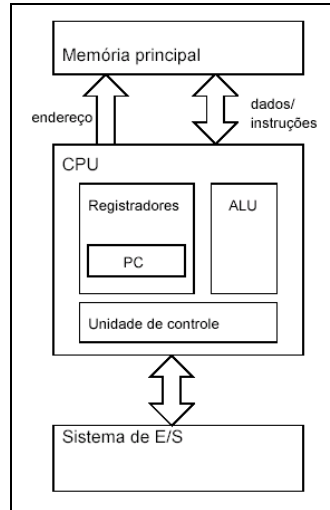


Figura 2 – Máquina von Neumann (RICARTE, 1999)

Uma variante do modelo básico de máquinas von Neumann é a arquitetura Havard (encontrada nos PICs), onde há vias separadas para dados e instruções entre memória principal e CPU (Figura 3). A origem do termo vem dos computadores Mark I a Mark IV, desenvolvidos em Harvard, com memórias de dados e instruções separadas (RICARTE, 1999).

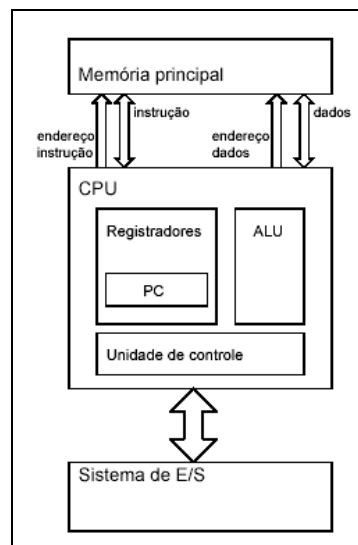


Figura 3 – Máquina Havard (RICARTE, 1999)

A Figura 4 mostra o diagrama de blocos do PIC 16F870, no qual podem ser visualizadas as diversas partes que o compõe, bem como as ligações entre as mesmas através de barramentos.

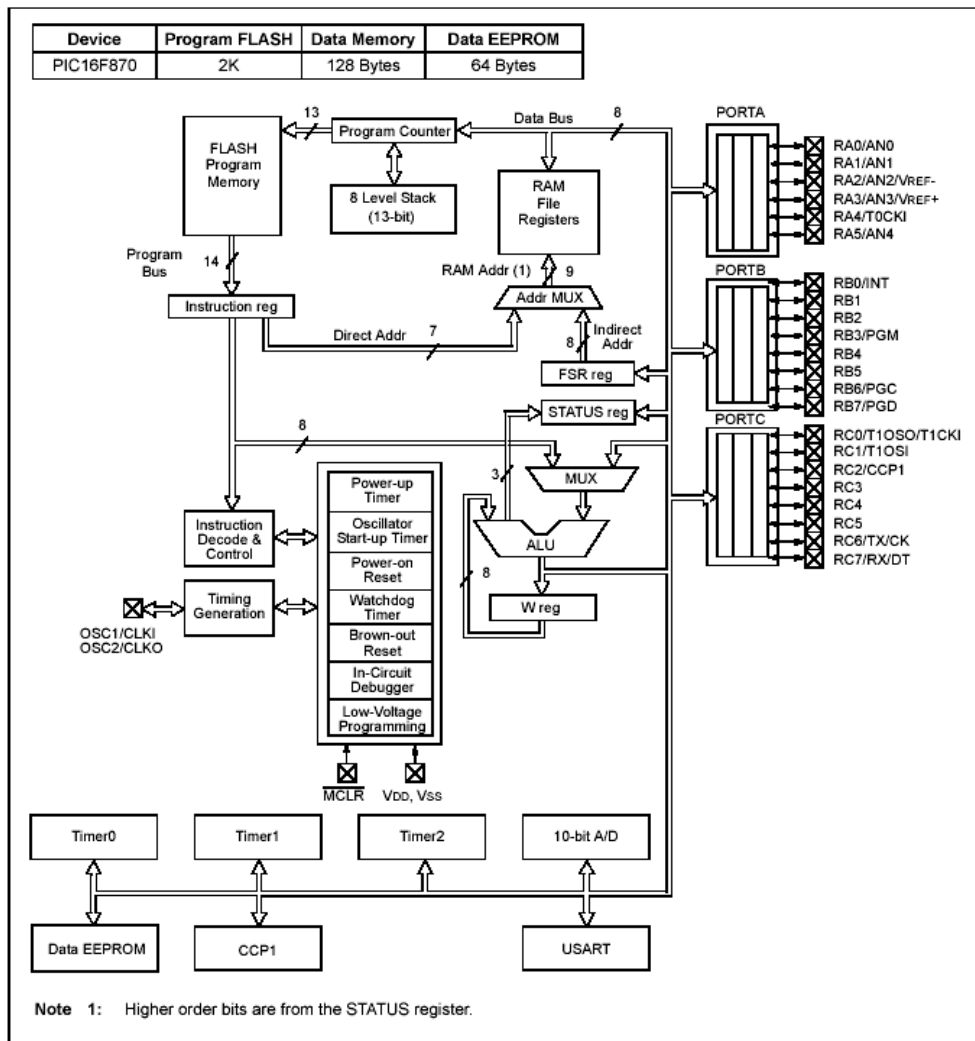


Figura 4 – Diagrama de Blocos do PIC 16F870 (MICROCHIP, 2004)

Os PICs utilizam uma tecnologia denominada RISC (*Reduced Instruction Set Computer*), que busca transferir a complexidade das operações para o software, mantendo em hardware apenas as operações primitivas. Desta forma, eles possuem um conjunto pequeno de instruções (o PIC16F870 possui 35), muito menor que o dos microcontroladores convencionais CISC (*Complex Instruction Set Computer*).

Isso torna o aprendizado mais fácil e dinâmico, mas, por outro lado, implica no fato de que muitas funções devem ser “construídas” devido à falta de uma instrução direta, exigindo maior habilidade do programador.

4.1.5 Organização da Memória

O PIC 16F870 apresenta três blocos de memória: Memória de Programa, Memória de Dados e EEPROM. Os dois primeiros blocos possuem barramentos distintos, permitindo que os mesmos sejam acessados simultaneamente.

4.1.5.1 Memória de Programa

Esta memória é implementada utilizando-se a tecnologia FLASH, o que torna possível programar o microcontrolador muitas vezes antes do mesmo ser instalado num dispositivo. Ocupa um espaço de 8192 endereços de palavras de 14 bits, divididos em 4 páginas de 2Kbytes. A Figura 5 mostra o mapeamento da memória de programa, assim como os 8 níveis da pilha.

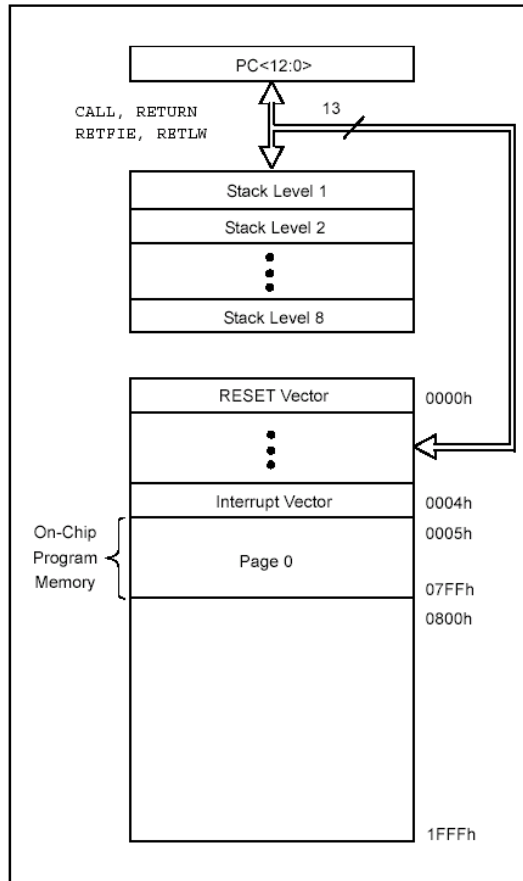


Figura 5 – Mapeamento da memória do PIC16F870 (MICROCHIP, 2004)

Vetor de Reset

Trata-se do primeiro endereço da memória executado quando o PIC começa a rodar (após a alimentação ou um *reset*). No modelo de PIC aqui utilizado, este endereço é 0x00, mas em alguns modelos mais antigos ele pode variar.

Vetor de Interrupção

As rotinas de interrupções são armazenadas na área de programa, juntamente com todo o resto do programa. No entanto, existe um endereço que é reservado para o início do tratamento de todas as interrupções. Este endereço é denominado vetor de interrupção, e encontra-se sempre na posição 0x04.

Pilha

A pilha é um local totalmente separado da memória de programa, onde são armazenados os endereços de retorno quando utiliza-se instruções de chamada de rotinas.

Quando o programa é desviado para o começo de uma rotina por meio da instrução correta, o endereço seguinte ao ponto que estava sendo rodado é armazenado na pilha para que, ao fim da rotina, o programa possa retornar. O tamanho da pilha também varia de acordo com o modelo do PIC (tem 8 posições neste modelo), determinando a quantidade de rotinas que podem ser chamadas ao mesmo tempo. Caso se tente chamar um número de rotinas maior que o tamanho da pilha, o endereço de retorno mais antigo será perdido.

4.1.5.2 Memória de Dados

A memória de dados do sistema é do tipo RAM (*Random Access Memory*), que é utilizada para guardar todas as variáveis e registradores utilizados pelo programa. Os registradores aqui mencionados consistem em unidades de armazenamento da memória RAM (Figura 6). Esta memória armazena dados de 8 bits por endereço e é volátil, sendo perdida automaticamente quando o PIC é desligado. Pode ser dividida em dois grupos principais: Registradores especiais e Registradores de uso geral.

File Address	File Address	File Address	File Address
Indirect addr. ⁽¹⁾	Indirect addr. ⁽¹⁾	Indirect addr. ⁽¹⁾	Indirect addr. ⁽¹⁾
00h	80h	100h	180h
TMR0	OPTION_REG	TMR0	OPTION_REG
01h	81h	101h	181h
PCL	PCL	PCL	PCL
02h	82h	102h	182h
STATUS	STATUS	STATUS	STATUS
03h	83h	103h	183h
FSR	FSR	FSR	FSR
04h	84h	104h	184h
PORTA	TRISA		
05h	85h	105h	185h
PORTB	TRISB	PORTB	TRISB
06h	86h	106h	186h
PORTC	TRISC		
07h	87h	107h	187h
PORTD ⁽²⁾	TRISD ⁽²⁾		
08h	88h	108h	188h
PORTE ⁽²⁾	TRISE ⁽²⁾		
09h	89h	109h	189h
PCLATH	PCLATH	PCLATH	PCLATH
0Ah	8Ah	10Ah	18Ah
INTCON	INTCON	INTCON	INTCON
0Bh	8Bh	10Bh	18Bh
PIR1	PIE1	EEDATA	EECON1
0Ch	8Ch	10Ch	18Ch
PIR2	PIE2	EEADR	EECON2
0Dh	8Dh	10Dh	18Dh
TMR1L	PCON	EEDATH	Reserved ⁽¹⁾
0Eh	8Eh	10Eh	18Eh
TMR1H		EEADRH	Reserved ⁽¹⁾
0Fh	8Fh	10Fh	18Fh
T1CON		110h	190h
10h	90h		
TMR2			
11h	91h		
T2CON	PR2		
12h	92h		
	93h		
	94h		
CCPR1L			
15h	95h		
CCPR1H			
16h	96h		
CCP1CON			
17h	97h		
RCSTA	TXSTA		
18h	98h		
TXREG	SPBRG		
19h	99h		
RCREG			
1Ah	9Ah		
	9Bh		
	9Ch		
	9Dh		
ADRESH	ADRESL		
1Eh	9Eh		
ADCON0	ADCON1		
1Fh	9Fh		
20h	A0h	120h	1A0h
General Purpose Register 96 Bytes	General Purpose Register 32 Bytes	accesses 20h-7Fh	accesses A0h - BFh
	BFh		1BFh
	C0h		1C0h
	EFh		1EFh
	F0h	16Fh	1F0h
		170h	
		accesses 70h-7Fh	accesses 70h-7Fh
7Fh	FFh	17Fh	1FFh

Unimplemented data memory locations, read as '0'.
^{*} Not a physical register.
Note 1: These registers are reserved; maintain these registers clear.
Note 2: These registers are not implemented on the PIC16F870.

Figura 6 – Registradores do PIC 16F870 (MICROCHIP, 2004)

Registradores especiais

Denominados SFRs (*Special Function Registers*), estes registradores são utilizados pelo microcontrolador para a execução do programa e processamentos da ALU. Podem ser acessados da mesma maneira que as variáveis do sistema, mudando apenas o endereço de acesso. Também podem ser escritos/lidos tanto pelo usuário quanto pelo *hardware*.

Registradores de uso geral

Trata-se de uma área destinada ao armazenamento de variáveis definidas pelo usuário para serem escritas e lidas pelo programa. Seu tamanho varia de acordo com o modelo de PIC, podendo ocupar também mais de um banco.

4.1.5.3 EEPROM

Existe ainda uma terceira memória que também pode ser utilizada pelo usuário para guardar dados. Entretanto, ao contrário da RAM esta é não volátil, sendo capaz de manter as informações mesmo sem alimentação.

A memória EEPROM (*Electrically Erasable Programmable Read-Only Memory*) consiste em 64 posições de oito bits cada. Ela não faz parte diretamente do espaço de memória, mas é acessada indiretamente através dos registradores EEADR e EEDATA.

4.1.6 Portas de E/S

Um microcontrolador não seria nada funcional se não fosse capaz de se comunicar com o mundo exterior. Para sanar esta deficiência, existem registradores interligando o barramento de dados às linhas de entrada e saída do PIC, que coincidem com os pinos do circuito integrado. A informação armazenada nestes registradores (chamados PORT) indica o estado dos pinos de E/S. Para adicionar flexibilidade e funcionalidade ao dispositivo, alguns pinos são multiplexados com funções alternativas como, por exemplo, entradas analógicas de conversores A/D.

A direção do fluxo de dados (entrada ou saída) nos pinos é controlada pelo registrador de direção de dados chamado TRIS. O valor binário “1” atribuído a determinado bit deste registrador define o pino correspondente como entrada e “0” como saída.

4.1.7 Temporizadores

Um temporizador nada mais é do que um registrador, que pode ser configurado como um contador ou como um temporizador propriamente dito.

Usado como temporizador, ele se conecta ao *clock* interno do microcontrolador e incrementa na frequência do *clock* dividido por 4. Um recurso chamado “*Prescaler*” controla a taxa de incrementos, e pode ser configurado para emitir pulsos na taxa de 1:2, 1:4, 1:8, etc. O término da contagem pode ser sinalizado por um *flag* ou via uma interrupção.

No modo de contagem, os incrementos ocorrem de acordo com pulsos aplicados a determinado pino do microcontrolador.

O PIC16F870 possui internamente 3 temporizadores, sendo 2 de 8 bits e 1 de 16 bits.

4.1.8 Interrupções

As Interrupções são ações provocadas por eventos assíncronos (podem ocorrer a qualquer momento) que causam um desvio no processamento.

Sempre que uma interrupção acontece, o valor do PC da próxima linha a ser executada é armazenado na pilha e a execução é desviada para um endereço fixo da memória de programa denominado “vetor de interrupção”. Neste endereço está a rotina escrita pelo programador que irá reconhecer e tratar a interrupção ocorrida. Depois que a mesma é executada, o programa automaticamente volta a executar a partir da linha apontada pelo PC.

Uma boa analogia para melhor entendermos o conceito de interrupção é a seguinte: você está digitando uma carta no computador quando o seu telefone toca. Neste momento, você interrompe o que está fazendo para atender ao telefone e verificar o que a pessoa do outro lado da linha está precisando. Terminada a conversa, você coloca o telefone no gancho novamente e retoma o seu trabalho do ponto onde havia parado. Observe que não há necessidade de verificar a todo instante se existe ou não alguém na linha, pois quando o telefone é chamado, ele toca avisando que existe alguém querendo falar com você.

Existe uma grande quantidade de interrupções com diferentes finalidades, dependendo do PIC. A habilitação das mesmas é gerenciada por uma chave geral (*Global Interrupt Enable*) que controla a ocorrência de todas as interrupções e por chaves específicas, que habilitam ou desabilitam cada uma delas individualmente.

4.1.9 Conversores A/D

Como os sinais dos periféricos são muitas vezes diferentes daqueles que o microcontrolador pode entender, eles devem ser convertidos num formato que possa ser compreendido pelo mesmo.

Esta tarefa é executada por intermédio de um conversor A/D, um dispositivo responsável pela conversão de uma informação de valor analógico (nível de tensão qualquer) para um número binário (“0” ou “1”) e pelo seu trajeto até a CPU, permitindo que o dado possa ser processado quando necessário.

O módulo de conversão A/D do PIC16F870 possui cinco entradas analógicas, e seu funcionamento é controlado pelos registradores ADCON0 e ADCON1.

O primeiro faz basicamente a seleção da entrada analógica a ser lida e controla o início da conversão, enquanto que o outro define se os pinos do microcontrolador serão definidos como entrada analógica ou E/S, além de configurar as tensões de referência do dispositivo.

Existem ainda dois outros registradores (ADRESH e ADRESL) que juntos armazenam o resultado da conversão A/D (um valor de 10bits).

4.1.10 USART

O PIC16F870 dispõem de um módulo de comunicação serial chamado USART (*Universal Synchronous Asynchronous Receiver Transmitter*), também conhecido como *Serial Communications Interface*, ou SCI.

A USART pode ser configurada para operar em dois modos distintos: síncrono e assíncrono, ambos implementados por duas vias de comunicação mais o terra.

No primeiro, uma das vias é utilizada como clock (via de sincronismo) enquanto a outra fica responsável pelo tráfego de dados. A comunicação ocorre de maneira bidirecional, mas não simultânea (*half-duplex*). Esta configuração é utilizada geralmente para a troca de informações entre o PIC e dispositivos periféricos como conversores A/D e D/A, memórias seriais externas, etc.

Já no modo assíncrono, não existe uma via de sincronismo, sendo uma das vias utilizadas para a transmissão (TX) e a outra para a recepção (RX) dos dados. Nesta comunicação, o sincronismo é feito pelo próprio dado através de marcas para o início (*Start bit*) e fim do dado (*Stop bit*).

4.2 Banco de Dados

No trabalho em questão, o banco de dados é responsável por armazenar os valores de temperatura e pH levantados pelo módulo de aquisição de variáveis e enviados ao PC através do microcontrolador. Ainda, ele deve manter registradas todas as anomalias ocorridas nas medições destas variáveis, além de informações referentes ao acionamento de dispositivos.

Os tópicos seguintes abordam alguns conceitos básicos relativos aos bancos de dados, numa tentativa de tornar mais clara a dinâmica de armazenamento das informações.

4.2.1 Visão Geral

A necessidade de armazenamento maciço e barato de informações com acesso direto desencadeou uma série de pesquisas na área de sistemas de banco de dados.

SILBERSCHATZ (1999) define banco de dados como uma coleção de dados inter-relacionados, representando informações sobre um domínio específico. Um sistema de banco de dados envolve os próprios dados, o *hardware* no qual eles residem e o *software* que controla o armazenamento e a recuperação de dados, chamado sistema gerenciador de banco de dados (SGBD).

Sistemas de bancos de dados são projetados para gerir grandes volumes de informações, implicando a definição das estruturas de armazenamento os mecanismos para a manipulação das mesmas. Também, devem garantir a segurança dos dados armazenados contra eventuais problemas com o sistema,

além de impedir tentativas de acesso não autorizadas. Se os dados são compartilhados por diversos usuários, o sistema deve evitar a ocorrência de resultados anômalos.

Um dos aspectos mais importantes destes sistemas é a independência de dados, isto é, os aplicativos não precisam se preocupar com a maneira como os dados são fisicamente armazenados ou acessados. Esta característica favorece a conveniência de que vários aplicativos podem ter diferentes visualizações dos mesmos dados.

Os usuários acessam um banco de dados através de linguagens de consulta especialmente projetadas referidas como linguagens nativas. Cada uma delas inclui normalmente uma sublinguagem de banco de dados voltada para as especificidades dos objetos e operações do banco de dados. Geralmente, cada sublinguagem de dados é uma combinação de duas outras linguagens: uma linguagem de definição de dados (*Data Definition Language* - DDL), que fornece recursos para definir os objetos do banco de dados, e uma linguagem de manipulação de dados (*Data manipulation Language* - DML), utilizada para especificar o processamento a ser realizado sobre objetos de banco de dados. A conhecida linguagem de consulta SQL (*Structured Query Language*) fornece tanto DDL como DML (DEITEL, 2001).

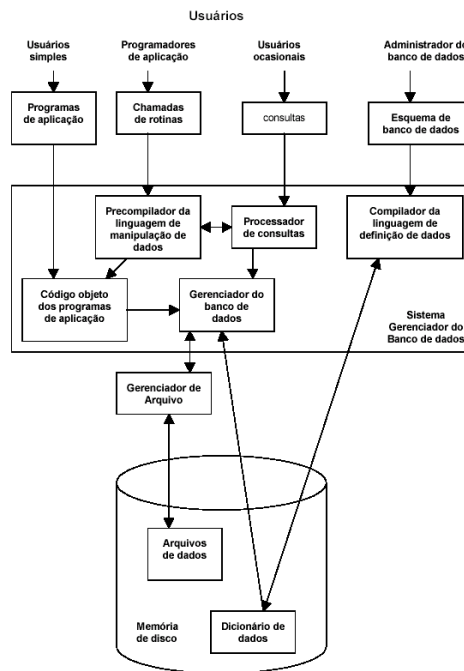


Figura 7 – Estrutura geral de um sistema de banco de dados (RAPCHAN, 2004)

Existem basicamente três tipos de modelos de dados, definidos por RAPCHAN (2004) como uma coleção de ferramentas conceituais para a descrição de dados, relacionamentos de dados, semântica de dados e restrições de integridade (consistência). São eles:

- Modelos lógicos baseados em objetos (Ex.: modelo E-R e modelo Orientado a Objetos);
- Modelos lógicos baseados em registros (Ex.: Modelo Relacional; em Redes e Hierárquico);
- Modelos Físicos de Dados. Descrevem os dados no nível mais baixo (Ex.: árvores binárias, grafos, etc.).

Como o modelo adotado para a realização deste trabalho é o relacional, apenas ele será abordado com maior detalhamento.

4.2.2 Modelo Relacional

O modelo relacional estabeleceu-se como o primeiro modelo de dados para aplicações comerciais. Os primeiros sistemas de banco de dados tiveram por base ou o modelo de rede ou o modelo hierárquico, os quais encontram-se mais próximos da implementação no nível físico que o modelo relacional.

Um banco de dados relacional consiste em uma coleção de tabelas, cada uma com um nome único. Uma linha em uma tabela representa um relacionamento entre um conjunto de valores. Uma vez que essa tabela é uma coleção de tais relacionamentos, há uma estreita correspondência entre o conceito de tabela e o conceito matemático de relação, a partir das quais se origina o nome desse modelo de dados.

4.2.3 MySQL

O MySQL é um servidor robusto de bancos de dados SQL, muito rápido, multitarefa e multiusuário. Foi desenvolvido originalmente para lidar com bancos de dados muito grandes de maneira muito mais rápida que as soluções existentes, e tem sido usado em ambientes de produção de alta demanda por diversos anos de maneira bem sucedida (MySQL AB, 2003).

Apesar de estar em constante desenvolvimento, o Servidor MySQL oferece hoje um rico e proveitoso conjunto de funções. A conectividade, velocidade, e segurança fazem com que ele seja altamente adaptável para acessar bancos de dados na Internet.

No MySQL, o conceito da estrutura que mantém os blocos (ou registros) de informações é chamado de tabela. Estes registros, por sua vez, são constituídos de objetos menores que podem ser manipulados pelos usuários, conhecidos por tipos de dados (*datatypes*). Juntos, um ou mais *datatypes*, formam um registro (*record*). Uma hierarquia de banco de dados pode ser considerada como: Banco de dados > Tabela > Registro > Tipo de dados. Os tipos de dados possuem diversas formas e tamanhos, permitindo ao programador criar tabelas específicas de acordo com suas necessidades (GONZAGA, 2000).

4.3 Sistema via Internet

O sistema que permite o acesso às informações armazenadas no banco de dados, bem como o acionamento de dispositivos, foi desenvolvido utilizando algumas técnicas e linguagens de programação que serão abordadas a seguir.

4.3.1 Java

Java é uma linguagem de programação desenvolvida pela empresa *Sun Microsystems*. Trata-se de mais um representante da nova geração de linguagens orientadas a objeto e foi projetada para resolver os problemas da área de programação cliente/servidor.

Os aplicativos em Java são compilados em um código de *bytes* independente de arquitetura chamado *bytecode*. Esse código de *bytes* pode então ser executado em qualquer plataforma que suporte um interpretador Java. A *Sun* chama esta característica de "*Write once, run anywhere*".

Hoje em dia, a maior parte do interesse pela linguagem Java está relacionado com sua habilidade em criar páginas na *Web*. Este é, hoje, o principal uso de Java, mas pode-se fazer implementações mais avançadas, como aplicativos completos de páginas interativas *Web*.

O suporte a rede e a *Web* fazem parte da linguagem. A facilidade de ler um arquivo local ou uma URL (*Universal Resource Location*) remota é a mesma. E é uma linguagem que já nasceu poliglota. O seu conjunto de caracteres de 16 bits (*Unicode*) suporta, desde o início, todos os idiomas conhecidos no mundo.

4.3.2 Applets

Um *applet* é geralmente um pequeno programa Java destinado a ser executado em um *browser* (Software utilizado para visualização de páginas da Internet, como o *Internet Explorer* ou o *Netscape Navigator*, por exemplo).

Há uma *tag* HTML (*Hyper Text Markup Language*) especial para ele (>APPLET>), que define uma "área de trabalho" semelhante à de um arquivo com extensão *.gif* ou *.jpg*. Dentro desta área o *applet* pode interagir com o usuário.

O *browser* faz o *download* do *applet* automaticamente. *Applets* pequenos são raramente notados, mas um *applet* mais complexo pode levar tempo para carregar, e infelizmente alguns *browsers* não revelam que eles estão ocupados carregando o *applet*. Então, um usuário impaciente pode interromper a exibição da página ou até mesmo tentar recarregá-la, acabando frustrado com o que está acontecendo.

O que ficou imediatamente claro para muitos desenvolvedores *Web* foi que o surgimento desta nova tecnologia trouxe a possibilidade de fazer GUIs (*Grafical User*

Interfaces) muito mais sofisticadas, permitindo a implementação de menus avançados, gráficos de negócio e efeitos em 3D.

O problema, entretanto, foi e ainda é que as máquinas virtuais não são 100% compatíveis. Uma GUI pode não ser a mesma para um programador PC e para um programador *Macintosh*. Assim os desenvolvedores se viram em um problema ao testar seus *applets* em uma grande variedade de *browsers*, versões e sistemas operacionais.

Com o *Internet Explorer 5.5* e o *Netscape 6* e suas promessas de suporte aos padrões, as pessoas começaram a usar menos *applets* e lançar mão de DHTML (*Dynamic HTML*) para manter as GUIs tão simples quanto possível (FUNDAÇÃO DA COMPUTAÇÃO, 2002).

4.3.3 Java Beans

Também conhecidos como *Beans*, os *JavaBeans* são componentes de *software* projetados para serem unidades reutilizáveis, que uma vez criadas possam ser reusadas sem modificação de código, e em qualquer propósito de aplicação.

Um modelo de componente é definido como um conjunto de classes e interfaces na forma de pacotes Java, que deve ser usado em uma forma particular para isolar e encapsular um conjunto de funcionalidades.

Para que esses componentes possam ser reconhecidos de forma geral, sua implementação deve seguir um conjunto de regras que serão usadas pelas ferramentas para introspecção da classe, ou seja, o processo de descobrir quais as funcionalidades do *Bean* e disponibilizá-la para o usuário. Cabe ao usuário fazer a

interface entre o componente e o restante da aplicação, criando assim um novo tipo de programador, o programador de componentes.

4.3.4 Servlets

Os *servlets* correspondem no lado do servidor aos *applets* no lado do cliente, sendo normalmente executados como parte de um servidor da Web (DEITEL, 2001).

São destinados a substituir o antigo CGI (*Common Gateway Interface*), uma das primeiras extensões para servidores Web simples, que tornaram possível a criação de conteúdo HTML dinâmico.

Um programa CGI primeiro obtém uma requisição de uma página HTML (geralmente através de um formulário), processa esta requisição e, finalmente, envia uma página HTML completa de volta ao *browser*.

O problema é que ele é muito ineficiente, uma vez que tem que iniciar um "processo" para cada requisição ao servidor. Servlets rodam em um único processo usando *threads*, permitindo uma melhor interação como servidor Web. E pelo fato de serem escritos em Java, são também portáteis (FUNDAÇÃO DA COMPUTAÇÃO, 2002).

4.3.5 JSP

JSP (*Java Server Pages*) é uma tecnologia para desenvolvimento de aplicações Web semelhante ao ASP (*Active Server Pages*) da *Microsoft*. Um fator que pode ser fundamental para a escolha entre estas duas ferramentas é que ASP é da *Microsoft* e só roda em ambiente *Windows*, além de todo *software* de apoio ao

desenvolvimento ser proprietário. Já o JSP, criado pela *Sun*, roda em qualquer plataforma que tenha o interpretador Java, e é amparado por vários softwares gratuitos necessários para disponibilizar as aplicações.

Ele permite ao desenvolvedor de *sites* produzir aplicações que permitam o acesso a banco de dados, o acesso a arquivos-texto, a captação de informações a partir de formulários, a captação de informações sobre o visitante e sobre o servidor, o uso de variáveis e *loops*, entre outras coisas.

O JSP não oferece nada que não se possa conseguir com os *servlets* puros, entretanto, tem a vantagem de ser facilmente codificado, facilitando assim a elaboração e manutenção de uma aplicação. Além disso, essa tecnologia permite separar a programação lógica (parte dinâmica) da programação visual (parte estática), facilitando o desenvolvimento de aplicações mais robustas, onde programador e *designer* podem trabalhar no mesmo projeto, mas de forma independente. Outra característica do JSP é produzir conteúdos dinâmicos que possam ser reutilizados.

Quando uma página JSP é requisitada pelo cliente através de um *browser*, ela é executada pelo servidor e, a partir daí, é gerada uma página HTML que é enviada de volta ao *browser* do cliente. A Figura 8 ilustra este funcionamento.

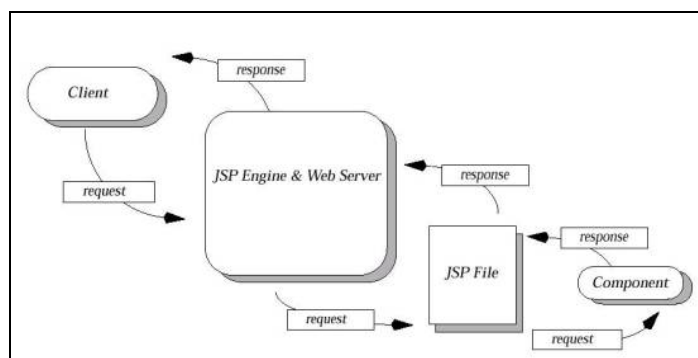


Figura 8 – Dinâmica de uma requisição a uma página JSP (DBSystems, 2004)

Quando o cliente faz a solicitação de um arquivo JSP, é enviado um *object request* para a *JSP engine*. A *JSP engine* envia a solicitação de qualquer componente (podendo ser um *JavaBeans component*, *servlet* ou *enterprise Bean*) especificado no arquivo. O componente controla a requisição possibilitando a recuperação de arquivos em banco de dados ou outro dado armazenado, em seguida, passa o objeto *response* de volta para a *JSP engine*.

A *JSP engine* e o *Web server* enviam a página JSP revisada de volta para o cliente, onde o usuário pode visualizar os resultados através do *browser*. O protocolo de comunicação usado entre o cliente e o servidor pode ser o HTTP (*Hyper Text Transfer Protocol*).

Por definição, JSP usa Java como sua linguagem de *scripts*. Por esse motivo, ele se apresenta mais flexível e mais robusto do que outras plataformas baseadas simplesmente em *JavaScripts* e *VBScripts*.

Para tirar um bom proveito do JSP é preciso entender a linguagem Java. Isso porque, a parte lógica do JSP envolve *Java Beans*, Objetos JDBC (*Java Data Base Connectivity*), *Enterprise Java Beans*, entre outros componentes que interagem com a plataforma Java. O ideal é que se conheça pelo menos um pouco de HTML.

Para conseguir ver os resultados das páginas criadas com base nesta tecnologia, é necessário um servidor JSP. Por se tratar de uma extensão ao modelo de programação Java, o JSP pode ser executado em qualquer plataforma.

Existem diversos servidores JSP espalhados na *Web*, sendo que alguns deles dispõem de versões para plataforma *Windows* e *Linux*. Dois dos mais famosos são o *Tomcat*, do projeto *Jakarta* e o *Resin* da empresa *Caucho*.

4.4 Módulo de Aquisição de Variáveis

Este módulo consiste num circuito eletrônico responsável pela aquisição dos valores de temperatura e pH da água. Ele basicamente transforma as informações captadas por sensores numa grandeza elétrica, compreensível pelos conversores A/D do microcontrolador.

Visando facilitar o entendimento deste mecanismo, serão abordados aqui alguns conceitos básicos sobre as variáveis em estudo, bem como alguns métodos para a mensuração das mesmas.

4.4.1 Temperatura

A temperatura é uma manifestação da energia cinética em um corpo. Associado ao conceito de temperatura (e com ele freqüentemente confundido) está o conceito de calor, que é a energia que aquece um corpo e eleva sua temperatura. (FRANCESCHINI, 2003).

Apesar de a temperatura ser perceptível pelo sistema nervoso humano, este se comporta como um péssimo “termômetro” pois, além de ser um instrumento diferencial, ou seja, só consegue distinguir entre o “mais quente” e o “mais frio”, ele pode induzir facilmente o observador ao erro, dado que é sensível a outras grandezas como, por exemplo, a condutibilidade térmica (uma maçaneta metálica parece mais fria ao tato do que a madeira da porta onde se encontra fixada, apesar de estarem à mesma temperatura). Sendo assim, são necessárias informações objetivas que possa medir o estado térmico de qualquer sistema.

Quando se coloca um corpo aquecido A em contato com um corpo frio B, após algum tempo os dois atingem um equilíbrio térmico, ou seja, em qualquer ponto em que se proceda a medição os dois corpos estarão à mesma temperatura.

De acordo com a lei zero da termodinâmica, se o corpo A está em equilíbrio com um corpo C e também está em equilíbrio com um corpo B, significa que os corpos B e C também estarão em equilíbrio entre si, ou seja, estarão à mesma temperatura. Neste caso, o corpo C é um termômetro, sensor utilizado para informar a temperatura após entrar em equilíbrio com o sistema analisado.

BRUSAMARELLO (2004) define sensores como elementos específicos que transformam determinada variável física de interesse em uma grandeza passível de processamento.

O termômetro é um sensor é feito de um material que varia determinada propriedade com a alteração da temperatura. Por volta de 1593, Galileu construiu o primeiro desses aparelhos. Era um tubo de vidro com um bulbo em uma das extremidades, enquanto a outra permanecia mergulhada em água. Com a variação de temperatura, a pressão do ar dentro do bulbo ficava alterada, tornando-se maior ou menor que a pressão atmosférica. A água então, subia ou descia pelo tubo. Assim, cada temperatura correspondia a uma altura da coluna de água. Mais tarde esse sistema foi aperfeiçoado pelo duque de Toscana, que encheu o bulbo e o tubo de vidro com álcool e selou a extremidade. Dessa maneira, a indicação do termômetro deixava de depender da pressão atmosférica e aumentava o seu grau de precisão.

Os primeiros termômetros apresentavam as mais diversas escalas, e a indicação de um deles nem sempre coincidia com a dos outros. Réaumur, Fahrenheit, Celsius e Linné eram algumas das escalas utilizadas em meados do

século XVIII. Era necessário estabelecer critérios universais para a construção dos termômetros.

Algumas escalas permaneceram em uso e foram convencionadas por padrões internacionais de medida. A escala centígrada (Celsius), por exemplo, é a mais utilizada em todos os países, a exceção dos daqueles de língua inglesa, onde a escala Fahrenheit é a mais freqüente. A construção de uma escala está associada a construção do termômetro. Fixam-se duas temperaturas de referência e se convencionam valores arbitrários a elas. Assim, se estabelece o ponto zero e o valor da unidade de temperatura.

Desde esta época, os sensores térmicos evoluíram muito. Atualmente existem sensores capazes de medir temperaturas com precisões bem razoáveis, como os abordados a seguir.

4.4.2 Sensores Termorresistivos

A resistência elétrica de vários tipos de materiais varia de alguma forma com a temperatura. Esta característica, muitas vezes indesejável, é o princípio básico de medida de temperatura através de elementos termorresistivos.

Estes elementos apresentam uma variação de sua resistência em função da temperatura de forma muito significativa e facilmente mensurável. Eles basicamente podem ser confeccionados de material metálico (condutores) ou semicondutor. Os materiais condutores, utilizados há muito tempo como sensores de temperatura, são tradicionalmente conhecidos pela sigla em língua inglesa RTD (*Resistance Temperature Detector*) e os semicondutores, de tecnologia mais recente, por termistores.

Nos metais a resistência aumenta quase que linearmente com a temperatura enquanto que nos semicondutores ela varia de maneira não-linear de forma positiva ou negativa (Figura 9).

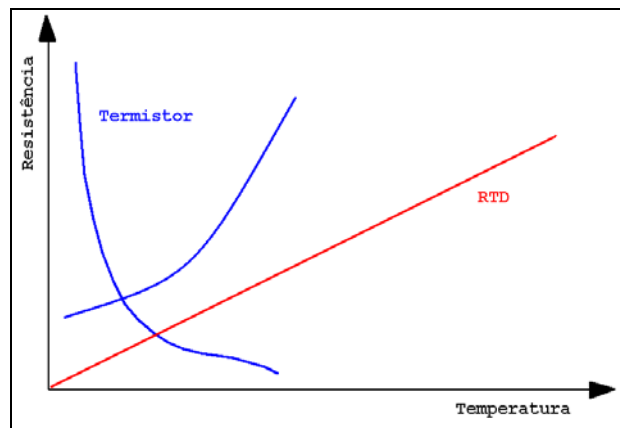


Figura 9 – Comportamento termo-sensível dos RTDs e Termistores (GÜTHS, 1998)

4.4.2.1 RTD

Todo material metálico, por melhor que sejam suas características condutivas, apresenta uma resistência elétrica que, por sua vez, varia com a temperatura (DANTAS, 2004). Esta dependência da resistência elétrica com a temperatura é o princípio fundamental utilizado pelos RTDs.

Apesar desta dependência, a relação entre estas duas grandezas (resistência e temperatura) varia de acordo com a natureza do metal em questão e não ocorre necessariamente de forma linear.

Embora todos os metais possam ser usados como sensores térmicos, alguns são mais interessantes por suas características físicas, onde é dada grande importância a aspectos como linearidade, sensibilidade, reprodutibilidade, imunidade

a influências externas, vida útil, robustez, etc. Os metais tradicionalmente utilizados são a platina, o níquel e o cobre.

Dada a alta estabilidade, imunidade a contaminações e ampla faixa de trabalho (-277 a 1064°C), a platina é o metal mais largamente utilizado na fabricação de sensores. Caso seja desejável trabalhar em sua faixa linear, esta apresenta erro de linearidade máximo de $\pm 0,4\%$ dentro do intervalo de -70 a 150°C ou $\pm 0,2\%$ dentro do intervalo de 0 a 100°C, neste último com uma exatidão de aproximadamente 0,001°C.

O termorresistor de platina mais utilizado é o PT100. Por definição, ele apresenta uma resistência elétrica de 100 Ω a 0°C com uma sensibilidade de 0,39 Ω /°C. Na Figura 10, pode-se ver a curva de calibração tabelada do PT100 para uma faixa de trabalho de -20 a 120°C.

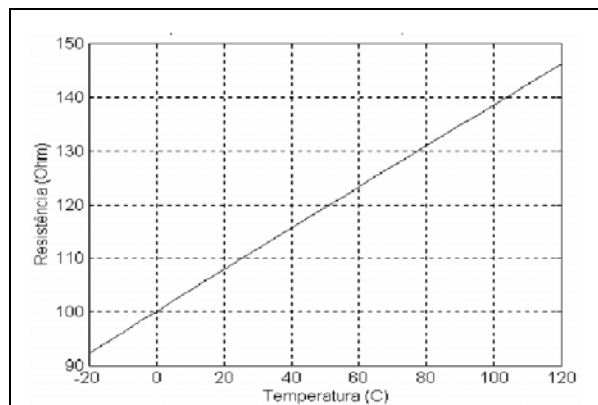


Figura 10 – Comportamento termo-sensível do PT100 (DANTAS, 2004)

O método mais difundido para se medir a resistência de um RTD é através de sua montagem em uma ponte de *Wheatstone*, como mostra a Figura 11. Ele constitui-se de dois divisores de tensão resistivos colocados em paralelo, onde a tensão de saída corresponde à diferença entre as tensões destes dois divisores. A

partir daí, qualquer variação no valor da resistência do RTD corresponderá a uma variação da tensão medida.

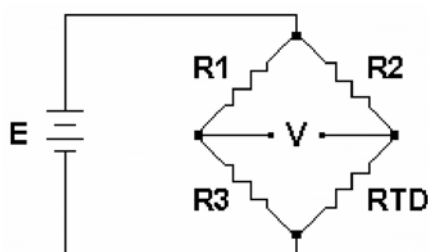


Figura 11 – Ponte de Wheatstone (DANTAS, 2004)

Apesar da simplicidade, a utilização da ponte de *Wheatstone* merece atenção especial no que concerne à alimentação, uma vez que o valor da tensão de saída é diretamente proporcional ao valor da tensão de alimentação. Portanto, a fonte deve ser estável para não comprometer a medição.

Uma ponte é dita balanceada quando a tensão de saída é nula, ou seja, os valores de resistência são tais que cada um dos braços apresentam a mesma relação de divisão de tensão.

Existem dois métodos de utilização da ponte de *Wheatstone*: método de detecção de nulo e método de deflexão.

Método de detecção de nulo

Como uma variação do valor do RTD causa um desbalanceamento da ponte, modificando a tensão de saída, este método consiste em se alterar o valor de um dos demais resistores de modo a tornar a ponte novamente balanceada. Assim, variando-se R2, quando a tensão de saída voltar a ser nula (ponte balanceada), o valor de R2 corresponderá exatamente ao valor do RTD.

Método de deflexão

Com exceção do RTD, todos os outros resistores são fixos, fazendo com que a tensão de saída varie de acordo com a temperatura, sendo esta inferida a partir de uma curva de calibração previamente determinada.

São utilizados valores elevados para os resistores, normalmente 10 vezes o valor do RTD no meio da faixa de trabalho, de modo a drenar pouca corrente da fonte. O valor de R2 é dimensionado para que a tensão de saída seja nula no início ou no meio da escala em que se está trabalhando. Quando esta engloba a temperatura de 0°C, costuma-se fazer a tensão de saída nula neste ponto.

Em muitos casos, para evitar que todo o circuito de medida seja exposto a grandes variações de temperatura, o RTD é levado até o local de medição, afastado do resto do circuito por longos fios. Isso faz com que as resistências elétricas destes fios tornem-se bastante significativas, além de também variarem com a temperatura sendo, portanto, uma provável fonte de erro.

Este problema é contornado pelo uso de RTD de três fios, conforme mostrado no circuito inferior da Figura 12.

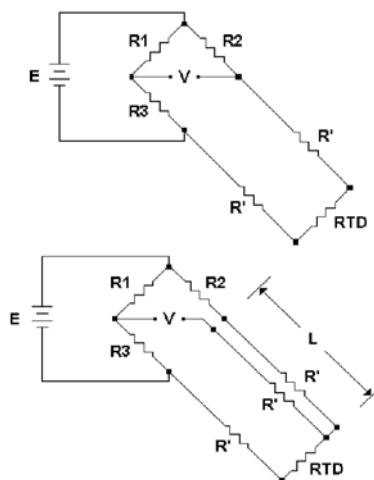


Figura 12 – RTDs de 2 e 3 fios (DANTAS, 2004)

Esta compensação de erro funciona da seguinte forma: quando os fios de conexão do RTD são de mesmo material, bitola e comprimento, suas resistências são praticamente iguais. A resistência do fio que liga um dos terminais do RTD ao circuito de medida de tensão pode ser desprezada, pois a impedância de entrada deste último deve ser bastante alta, drenando pouquíssima corrente em relação aos outros elementos da ponte. Já as resistências dos outros dois fios, em vez de fazerem parte apenas do lado inferior do braço da ponte onde se encontra o RTD, são divididas, uma ficando junto com o RTD, na parte inferior, e a outra com R2, na parte superior.

No método de medida por detecção de nulo, onde o valor de R2 é variado de modo a balancear a ponte ($V = 0$), e com os resistores R1 e R3 iguais, as resistências dos fios cancelam-se completamente.

Já no método de deflexão, os erros são diminuídos, mas não extinguidos. O dimensionamento dos resistores segue a mesma orientação dada para RTD de dois fios.

Outra forma de minimização deste tipo de erro é o uso de resistores de valores mais elevados em relação à magnitude das resistências dos fios de conexão, apesar do termorresistor de platina de 100Ω (PT100) ser tradicionalmente o mais utilizado.

4.4.2.2 Termistores

Os termistores são sensores fabricados com materiais semi-condutores como óxido de magnésio ou cobalto. Por serem construídos de material semi-condutor, possuem a grande vantagem de poderem ser fabricados em um tamanho físico

muito pequeno. O termistor de coeficiente negativo de temperatura (NTC) é um sensor muito conhecido e encontrado no mercado com uma variedade muito grande no tipo construtivo e nos valores de resistência. Ao contrário dos demais, ele diminui sua resistência elétrica com o aumento da temperatura. Já o termistor de coeficiente positivo (PTC), é mais raro de ser encontrado, dada sua complexidade no aspecto construtivo.

O termistor é um sensor muito sensível, possibilitando performances com erros de até 0,01 °C com calibração adequada. Apesar disso, obviamente, tem a desvantagem de ser não linear, o que força a utilização de um sistema para fazer o ajuste da temperatura em função da resposta do mesmo, geralmente implementado na forma de programação de um sistema de aquisição de dados.

Uma vez que a resistência dos termistores é muito alta, os erros devido a cabos e conexões é desprezível. Adicionalmente, devido a alta resistência correntes muito baixas são resultantes, o que minimiza os erros devido a autoaquecimento. Como os termistores são semicondutores, estão sujeitos a deteriorar-se em altas temperaturas, e assim limitados para medições até 300°C.

4.4.3 pH

O pH (Potencial Hidrogeniônico) é um termo utilizado para expressar a intensidade da condição ácida ou básica de um determinado meio. É definido como o logaritmo negativo da concentração hidrogeniônica $[H^+]$. Intimamente ligado a este conceito está o pOH (Potencial Oxidriônico ou Hidroxiliônico) que, por analogia, é definido como o logaritmo negativo da concentração hidroxiliônica $[OH^-]$.

A acidez ou a alcalinidade (basicidade) de uma solução é expressa em uma escala de pH que vai de 0 a 14, sendo que valores de 0 a 7 são considerados ácidos, 7 neutros e acima de 7 básicos ou alcalinos, sendo esta é baseada no número de H⁺ em uma solução e expressa em unidades químicas chamadas de moles por litro (Figura 13). A soma de pH + pOH sempre resultará 14.

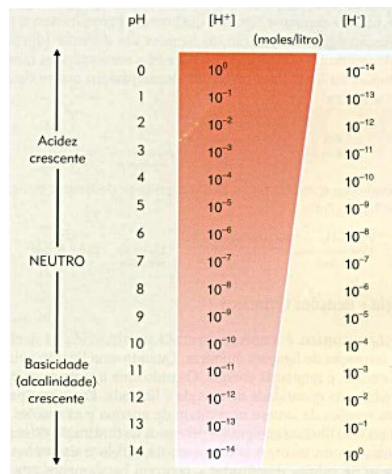


Figura 13 – Escala de pH (ROCHA, 2004)

Segundo Oliveira (2004), a determinação do pH é uma das mais comuns e importantes no contexto da química da água. No campo do abastecimento de água, o pH intervém na coagulação química, controle da corrosão, abrandamento e desinfecção. O padrão de potabilidade em vigor no Brasil preconiza uma faixa de pH entre 6,5 e 8,5.

No âmbito do tratamento de águas residuárias por processos químicos ou biológicos, o pH deve ser mantido em faixas adequadas ao desenvolvimento das reações químicas ou bioquímicas do processo. No tratamento de lodos de estações de tratamento de esgotos, especificamente através da digestão anaeróbia, o pH se constitui num dos principais fatores de controle do processo.

Em lagoas e reservatórios de estabilização de esgotos o aumento do pH, como consequência da fotossíntese de algas, desempenha importante papel na eliminação de organismos patogênicos.

4.4.4 Sensores de pH

Na prática, o pH é medido com indicadores ácido base, substâncias orgânicas que possuem a propriedade de mudar de coloração com a variação do pH do meio. A mudança de coloração se processa de maneira gradual entre valores definidos da escala de pH.

O princípio da medição eletrimétrica do pH é a determinação da atividade iônica do hidrogênio utilizando o eletrodo padrão de hidrogênio, que consiste de uma haste de platina sobre a qual o gás hidrogênio flui. O eletrodo de hidrogênio, no entanto, não é bem adaptado para uso universal, especialmente em trabalho de campo ou em soluções contendo espécies químicas contaminantes do eletrodo. Assim, um outro eletrodo, o de vidro (Figura 14), é comumente utilizado.

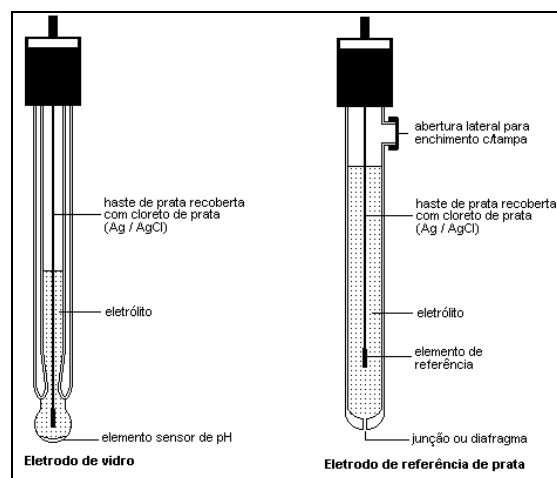


Figura 14 – Eletrodo de vidro (OLIVEIRA, 2004)

O eletrodo de vidro é um bulbo construído em vidro especial contendo uma solução de concentração fixa de ácido clorídrico, ou uma solução tamponada de cloreto em contato com o eletrodo de referência interno, normalmente constituído de prata revestida de cloreto de prata, que assegura um potencial constante na interface da superfície interna do sensor com o eletrólito.

O elemento sensor do eletrodo, situado na extremidade do bulbo, é constituído por uma membrana de vidro que, hidratada, forma uma camada de gel, externa, seletiva de íon hidrogênio. Essa seleção é, de fato, uma troca de íons sódio por íons hidrogênio os quais formam uma camada sobre a superfície do sensor. Além disso, ocorrem forças de repulsão de ânions por parte do silicato, negativamente carregado, que está fixo no sensor. Ocorre, na camada externa do sensor, a geração de um potencial que é função da atividade do íon hidrogênio na solução.

A uma Temperatura ambiente de 25° C, o eletrodo de vidro gera uma voltagem de 59 mV para cada unidade de pH. A uma temperatura de 50° C, a voltagem gerada é de 65 mV e a 100° C ela vale 74 mV. Vê-se que a representação gráfica dessa voltagem, em função do valor do pH, é uma reta cuja inclinação depende da temperatura (Figura 15).

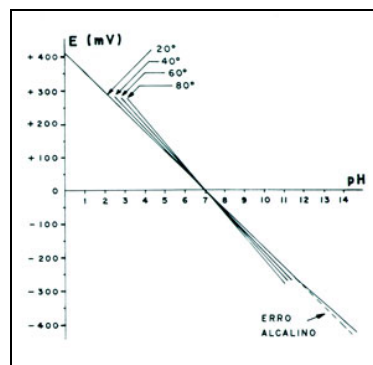


Figura 15 – Variação na medida do pH em função da temperatura

O eletrodo de referência consiste de uma meia célula de potencial constante e determinado. No interior de um bulbo (Figura 14) o elemento de referência acha-se imerso num eletrólito a qual entra em contato com a amostra através de junção (líquida) ou diafragma por onde se forma uma ponte salina a qual deve desenvolver um potencial de junção mínimo possível. O eletrodo de referência é bastante útil em determinações que se baseiam na medida de diferença de potencial entre o eletrodo específico e o eletrodo de referência. Os eletrodos de referência mais comumente usados são calomelano ($\text{Hg} / \text{Hg}_2\text{Cl}_2$) e prata/cloreto de prata.

A utilização de um par de eletrodos sempre se impõe para a determinação de íons específicos e de pH em soluções viscosas e suspensões coloidais. O eletrodo de vidro combinado ilustrado na Figura 16 é um eletrodo compacto no qual o eletrodo de vidro acha-se envolvido pelo eletrodo de referência de prata/cloreto de prata. É um eletrodo adequado para a maioria das aplicações de laboratório sendo mais fácil de manusear que o par de eletrodos separados. Os mais recentes têm também um sensor de temperatura integrado, útil na compensação automática de leituras de temperatura de diferentes amostras.

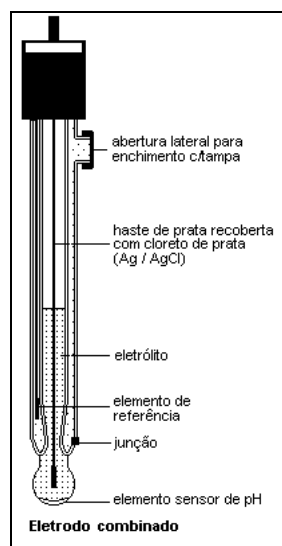


Figura 16 – Eletrodo Combinado (OLIVEIRA, 2004)

5 DESENVOLVIMENTO

Aqui serão apresentados todos os módulos que compõem o trabalho desenvolvido, abordando de uma forma detalhada o funcionamento de cada um, bem como as relações existentes entre os mesmos.

A Figura 17 dá uma visão geral do trabalho desenvolvido.

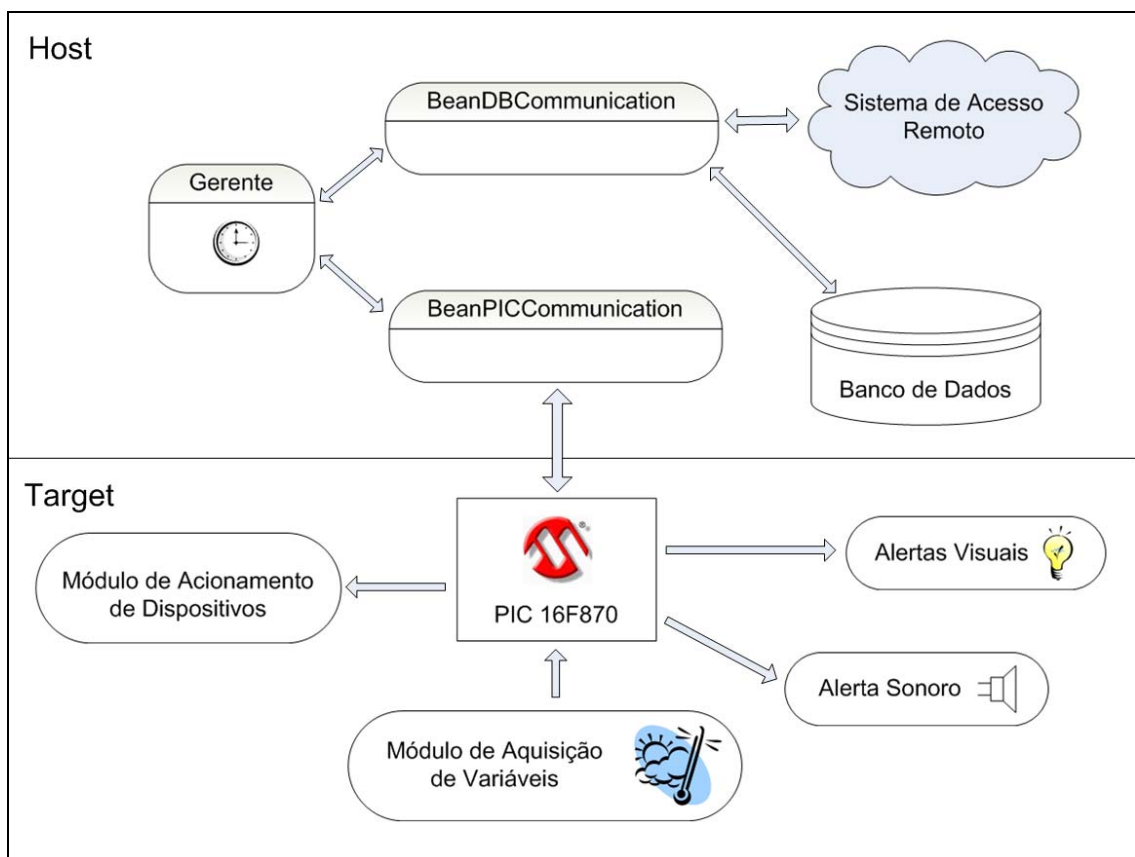


Figura 17 – Visão geral do trabalho

5.1 Host

O Host corresponde aos componentes do sistema localizados “do lado” do PC. Ele é composto pelo banco de dados, pelo sistema de acesso remoto e por um

software denominado Gerente, juntamente com duas classes que fazem a comunicação com o banco de dados e com o PIC.

5.1.2 Estrutura do Banco de Dados

O banco de dados utilizado foi o MySQL, já comentado anteriormente. Para uma melhor compreensão de sua finalidade, as tabelas construídas são descritas a seguir.

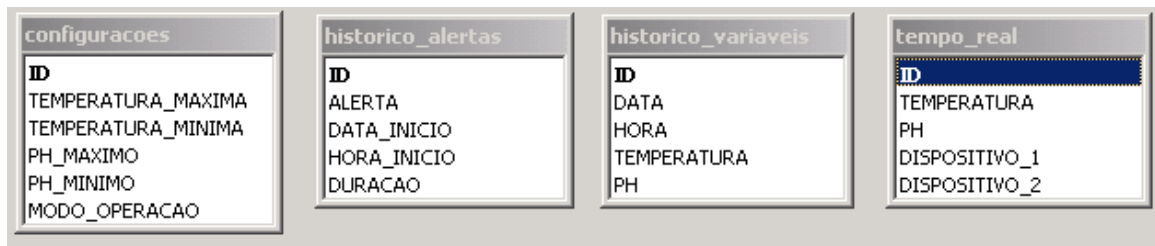


Figura 18 – Modelagem do Banco de Dados

A tabela “configuracoes” é responsável por manter as definições das faixas de temperatura e pH desejadas, bem como o modo de operação do sistema. Se o campo MODO_OPERACAO contiver o valor “Automático”, o PIC pode ligar e desligar os dispositivos de correção de variáveis (aquecedor e bomba de CO₂) de maneira automatizada, conforme a necessidade. Caso contrário (valor do campo = “Manual”), o acionamento destes dispositivos deve ser feito manualmente, através do sistema de acesso remoto.

Na tabela “histórico_variaveis” são armazenados os valores de temperatura e pH colhidos em determinados instantes pré-definidos, juntamente com a data e a hora em que foram inseridos no banco de dados.

De maneira semelhante, a tabela “histórico_alertas” mantém um histórico das anomalias ocorridas com as variáveis ao longo do tempo, registrando os momentos de início e a duração das mesmas. Existem 4 tipos de alertas definidos, como mostra a Figura 19.

1	Temperatura acima do limite máximo.
2	Temperatura abaixo do limite mínimo.
3	pH acima do limite máximo.
4	pH abaixo do limite mínimo.

Figura 19 – Tipos de Alertas

Finalmente, na tabela “tempo_real” encontram-se os valores de temperatura e pH informados a cada 1 segundo pelo PIC, possibilitando o acompanhamento via Internet destas variáveis em tempo real. Nela também estão registradas os estados (Ligado ou Desligado) do aquecedor e da bomba de CO₂.

5.1.3 Gerente

O Gerente é um componente fundamental para que as interações entre os módulos aconteçam corretamente. Trata-se de um software responsável pela troca de informações entre o PIC e o PC e por parte do armazenamento de dados. Para realizar suas tarefas ele faz uso de duas classes chamadas BeanPICCommunication e BeanDBCommunication.

Seu funcionamento básico pode ser visualizado na rotina da Figura 20.

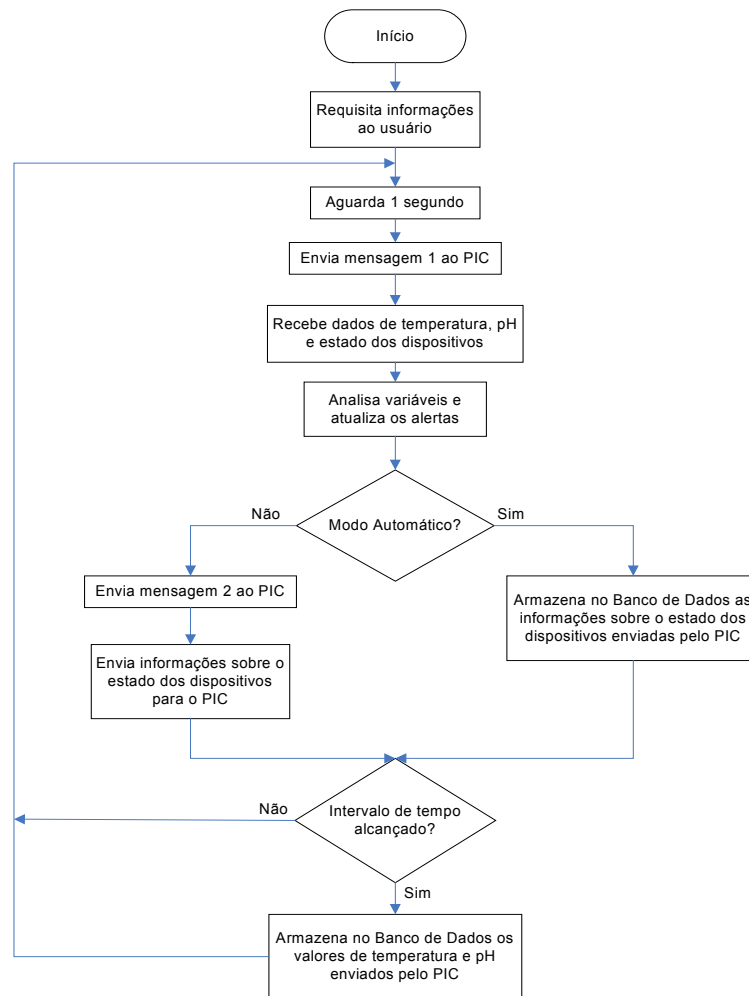


Figura 20 – Rotina de funcionamento do Gerente

Na inicialização do programa, o usuário deve informar (em minutos) com que frequência os valores de temperatura e pH serão armazenados no banco de dados. A base de tempo para a realização do controle (data e hora de início) também deve ser inserida. Quando este momento é alcançado, o Gerente entra num *loop* infinito.

Ele então aguarda 1 segundo e envia uma mensagem (msg1) ao PIC. Esta consiste numa requisição para que o microcontrolador envie as leituras de temperatura e pH, seguidas do registro contendo o estado atual dos dispositivos. Este, por sua vez, atende o pedido. É realizada então uma análise destas leituras,

seguida de uma possível atualização ou inserção de alertas da tabela “historico_alertas”.

Se o sistema estiver funcionando em modo automático, os dados referentes à situação dos dispositivos são atualizados na tabela “tempo_real”. Caso contrário, o Gerente envia outra mensagem (msg2), avisando que irá mandar as informações que definem o estado no qual estes dispositivos devem permanecer (informado no sistema de acesso remoto).

Em seguida é verificado se o intervalo de tempo decorrido entre os armazenamentos de variáveis foi alcançado. Se a verificação for negativa, o *loop* é reiniciado, caso contrário, as informações referentes às mesmas são registradas antes disso.

5.1.4 Sistema de Acesso Remoto

Este módulo, desenvolvido utilizando basicamente as linguagens de programação Java e JSP, possibilita o monitoramento e o controle via Internet das variáveis hídricas.

Encontra-se subdividido em algumas seções, que serão abordadas a seguir.

5.1.4.1 Configurações

Aqui o usuário pode configurar os valores máximos e mínimos em que deverão ser mantidas as variáveis, além de definir o funcionamento do sistema como automático ou manual.

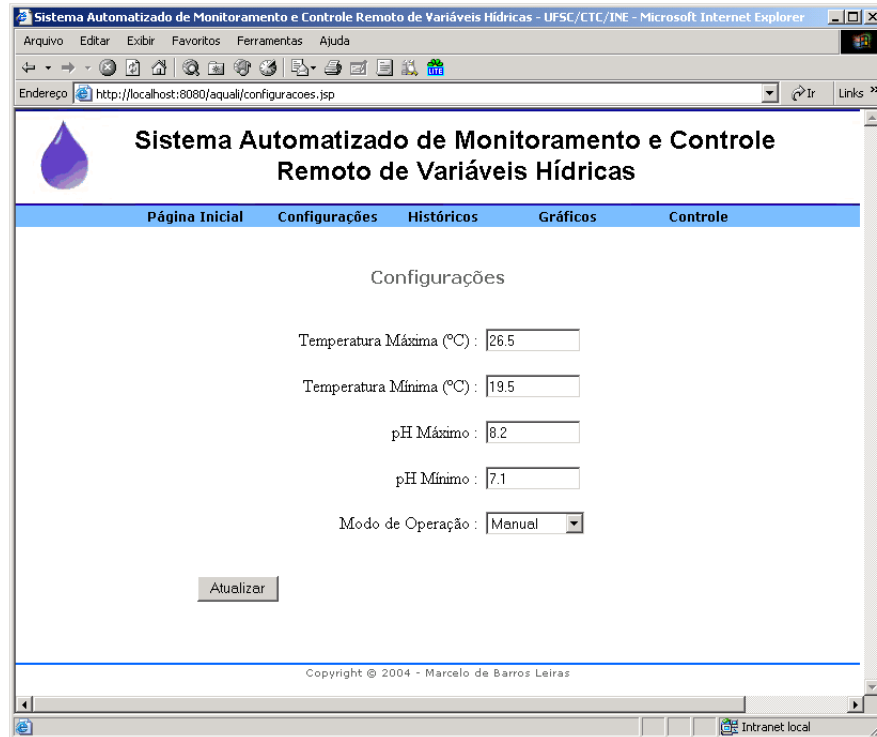


Figura 21 – Configurações

As alterações só têm efeito quando o botão “Atualizar” é pressionado (Figura 21). Então, as informações inseridas são atualizadas na tabela “configuracoes”.

5.1.4.2 Históricos

Permite que o usuário consulte o histórico de valores de temperatura, pH e alertas, armazenados na tabela “historico_variaveis” e “historico_alertas”, possibilitando que a informação desejada seja filtrada (Figura 22) de acordo com alguns critérios antes da visualização (Figura 23).

Sistema Automatizado de Monitoramento e Controle Remoto de Variáveis Hídricas

Página Inicial Configurações Históricos Gráficos Controle

Histórico de Temperatura e pH

Período (aaaa-mm-dd) : à

Hora (hh:mm) : às

Temperatura (°C) : à

pH : à

Enviar

Copyright © 2004 - Marcelo de Barros Leiras

Concluído Intranet local

Figura 22 – Filtragem de informações para a consulta

Sistema Automatizado de Monitoramento e Controle Remoto de Variáveis Hídricas

Página Inicial Configurações Históricos Gráficos Controle

Histórico de Temperatura e pH

Data	Hora	Temperatura (°C)	pH
2004-06-19	17:30:00	28	8.1
2004-06-09	16:30:00	30.2	7.6
2004-06-01	07:00:00	28.3	7.4
2004-06-01	06:00:00	29.5	7.2
2004-06-01	05:00:00	27	7

Copyright © 2004 - Marcelo de Barros Leiras

Intranet local

Figura 23 – Visualização das informações solicitadas

5.1.4.3 Gráficos

Funciona de uma forma semelhante à seção anterior. Porém, aqui as informações são apresentadas na forma de gráficos.

5.1.4.4 Controle

Aqui é possível visualizar os valores correntes de temperatura e pH. Uma vez acessada, a seção é recarregada a cada 2 segundos, permitindo um acompanhamento das variáveis em tempo real.

Permite também controlar o acionamento do aquecedor e da bomba de CO₂ quando o sistema estiver funcionando em modo manual.

A Figura 24 apresenta sua interface gráfica.

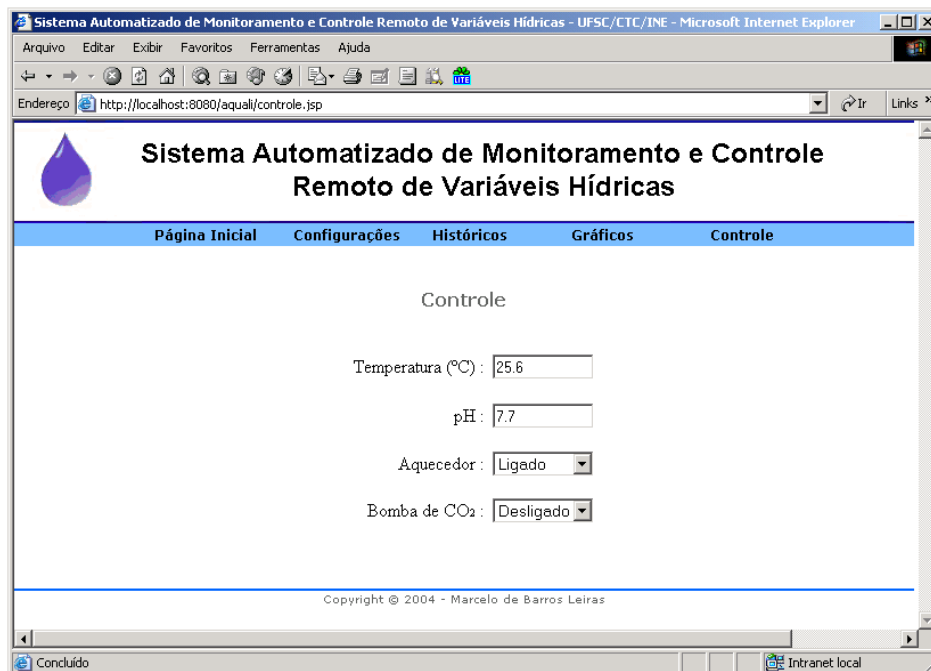


Figura 24 – Controle

5.2 Target

Ao contrário do Host, o Target corresponde aos componentes do sistema localizados “do lado” do PIC, sendo composto pelo próprio microcontrolador e pelo circuito mínimo requerido para seu funcionamento (alimentação, oscilador e reset), além dos módulos de leitura de variáveis e acionamento de dispositivos.

5.2.1 PIC

O microcontrolador é o elemento responsável pelo controle dos processos de aquisição de variáveis e acionamento de dispositivos, além de realizar toda o tipo de troca de informações com o PC.

5.2.1.1 Circuito Básico de Funcionamento

Para que o PIC possa funcionar, ele precisa de alguns requisitos básicos como alimentação e oscilador, além de um circuito de *reset* (Figura 25).

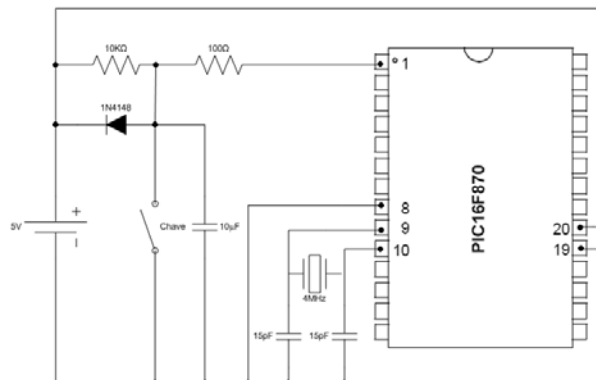


Figura 25 – Circuito básico de funcionamento do PIC

5.2.1.2 Rotina principal

Quando o PIC é inicializado, ele passa a executar um *loop* infinito, como mostra o fluxograma da Figura 26.

Esta rotina basicamente fica monitorando os valores de temperatura e pH, ativando ou desativando os alertas visuais (representados por LEDs) conforme a necessidade. Caso o sistema esteja funcionando em modo automático, ela também fica responsável por gerenciar o acionamento dos dispositivos.

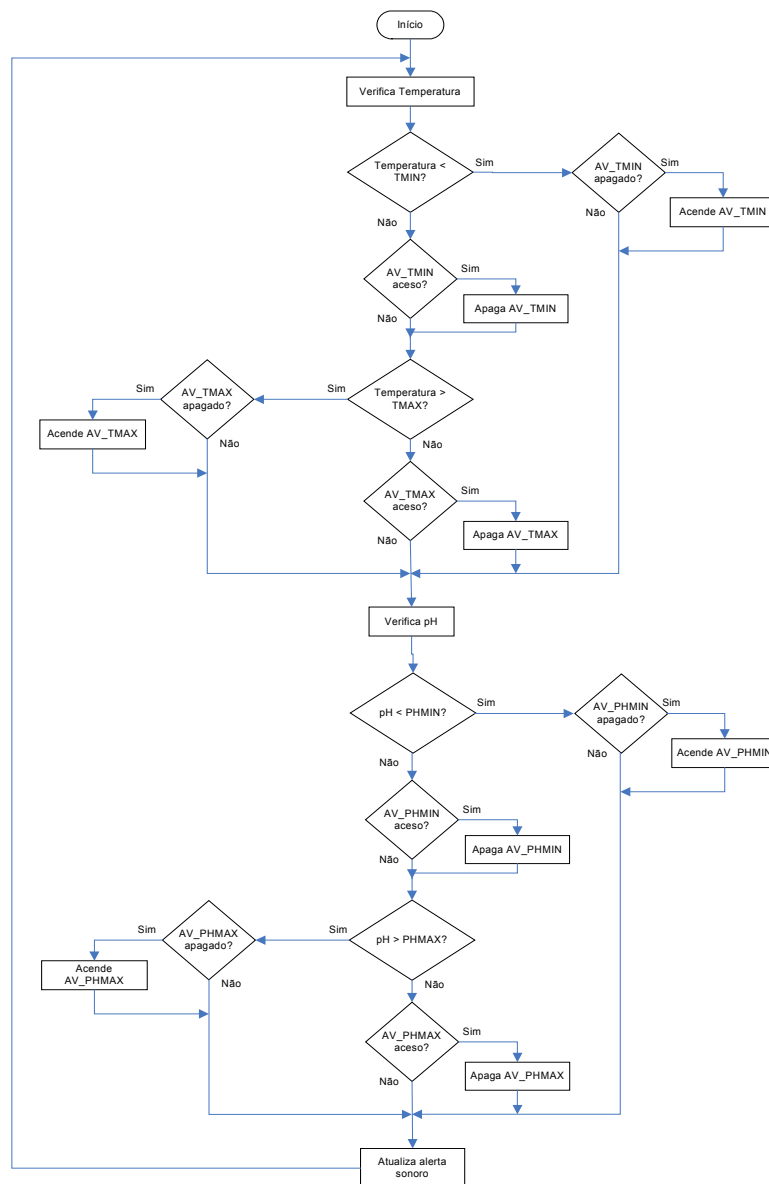


Figura 26 – Rotina Principal do PIC

5.2.1.3 Rotina de Interrupção

É acionada quando o PIC recebe alguma mensagem vinda proveniente do Gerente. Quando isto acontece, a rotina principal é interrompida, a mensagem recebida é identificada e a ação correspondente é executada. Feito isso, o programa volta a sua execução normal. Seu funcionamento mais detalhado está presente no fluxograma da Figura 27.

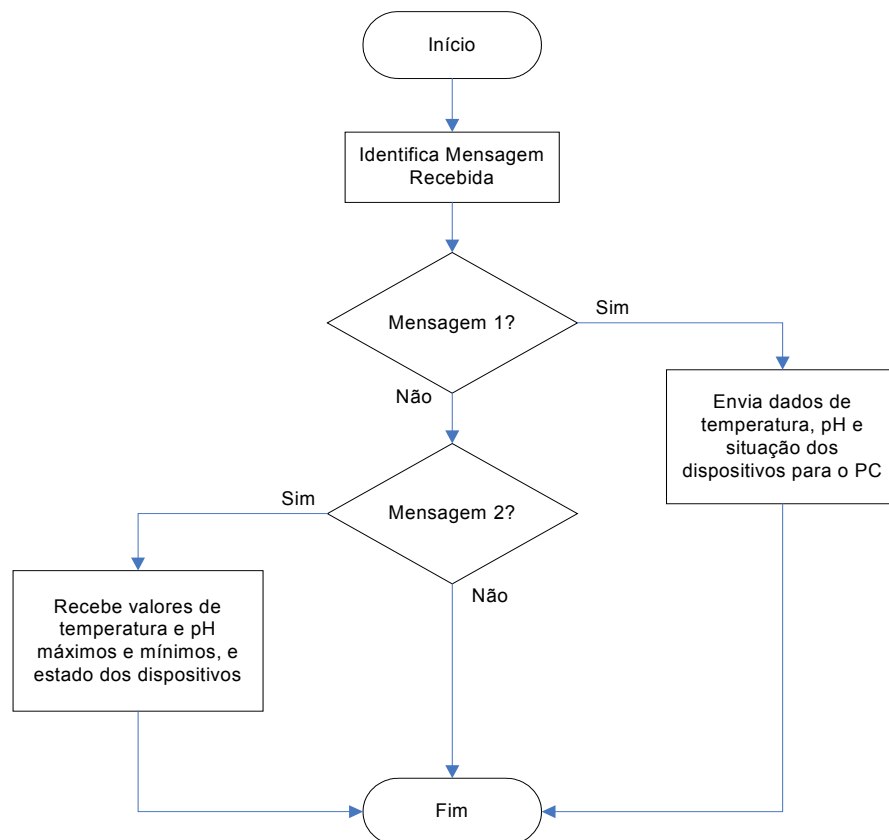


Figura 27 – Rotina de Interrupção do PIC

5.2.2 Leitura de Variáveis

Ainda não foi feita a aquisição do sensor de pH. Então, o módulo de leitura de variáveis ficou limitado a aquisição de temperatura. Mesmo assim, todo o sistema foi

desenvolvido como se este sensor estivesse presente, bastando algumas pequenas modificações para que a coleta desta variável seja agregada.

Para a aquisição da temperatura foi utilizada uma ponte de *Wheatstone* com medição a dois fios acoplada a um amplificador operacional (OPA551), além do sensor PT-100.

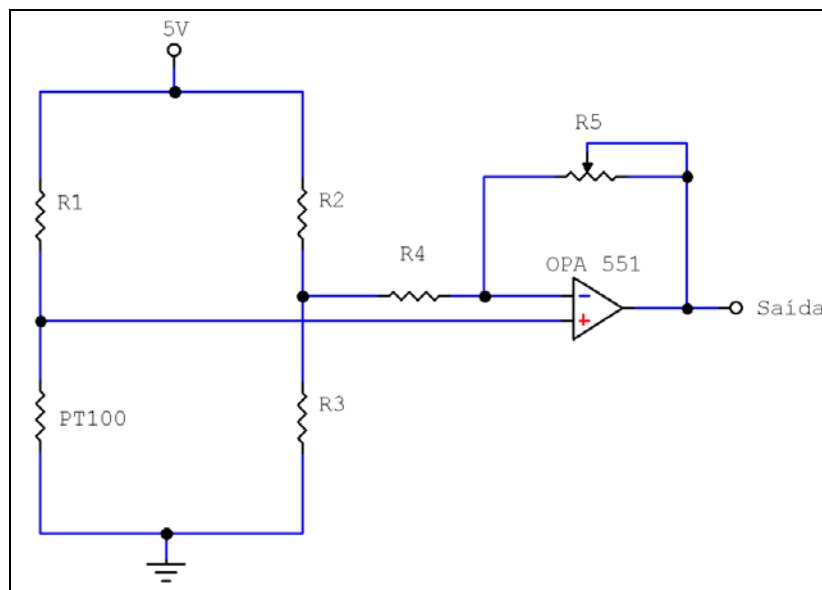


Figura 28 – Circuito de aquisição de temperatura

5.2.3 Acionamento de Dispositivos

Este módulo é responsável pelo acionamento do aquecedor e da bomba de CO₂ para o ajuste dos valores de temperatura e pH. Os pinos de saída do PIC reservadas para esta função foram conectados aos pinos 2 e 3 do *buffer* do circuito da Figura 29.

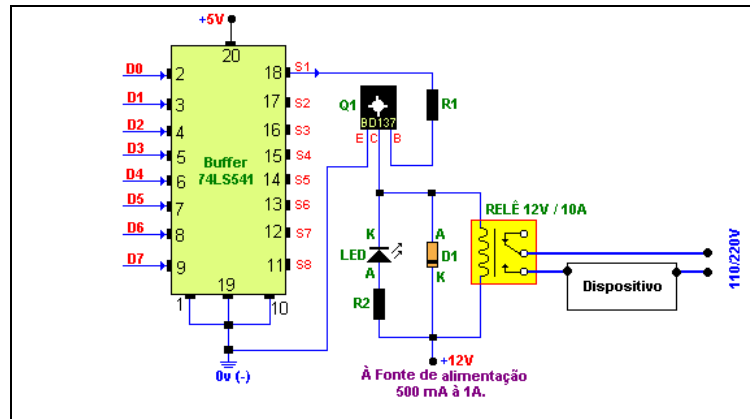


Figura 29 – Circuito de Acionamento de Dispositivos

No circuito acima, o buffer 74LS541 é usado para proteger o PIC de altas correntes. Ele é alimentado com 5V, diferente da outra parte do circuito que controla o relê, que precisa de 12V para ser acionado.

Como são dois os dispositivos a serem controlados, também foi utilizada a saída S2 do buffer, seguida de um circuito idêntico ao da S1.

6 ESTADO ATUAL DO SISTEMA

O software de acesso remoto encontra-se praticamente pronto. As seções de Configurações, Históricos e Controle funcionam perfeitamente, trocando informações corretamente com o banco de dados por meio da classe *BeanDBCommunication*, conforme o propósito de cada uma.

A única lacuna é a parte do acompanhamento das variáveis armazenadas através de gráficos. A principal dificuldade está na carência de documentação sobre o componente escolhido para a geração de gráficos, denominado *Cewolf*. Seu criador disponibiliza diferentes versões de diversos módulos que precisam se integrar para a exibição dos gráficos. Porém, nem todas as versões funcionam.

No lado do *Target*, a rotina principal do PIC já está funcionando, juntamente com o circuito de aquisição de temperatura. Sendo assim, já é possível monitorar o valor da temperatura emitindo alertas visuais e sonoros quando necessário. Porém, o circuito de acionamento de dispositivos ainda não foi montado devido à dificuldade de encontrar certos componentes nas deficitárias lojas de eletrônica de Florianópolis.

Porém, a principal dificuldade no desenrolar das atividades está na comunicação do microcontrolador com o PC, através da USART. A configuração deste dispositivo se mostrou como uma tarefa não muito trivial, formando uma barreira para a integração geral do sistema.

Talvez, ao se deparar com o conteúdo deste trabalho o leitor não se dê conta da estrutura necessária para a elaboração do mesmo. Por exemplo, o microcontrolador utilizado, juntamente com seu módulo de gravação, não são encontrados facilmente em qualquer estabelecimento especializado. Neste caso específico, ambos foram adquiridos através de fornecedores contatados via Internet.

Portanto, a entrega não é imediata. Há um certo tempo primeiro para a pesquisa de fornecedores (que não são muitos e nem sempre inspiram confiança), depois para o fechamento do negócio e envio do produto. E pra piorar, infelizmente nem sempre a encomenda recebida está em perfeitas condições. Sendo assim, acabe-se ficando à mercê da boa vontade dos fornecedores, o que muitas vezes pode atrasar o cronograma.

Então, em virtude do trabalho não estar concluído totalmente, ainda não foi possível realizar o estudo de caso no LCM previsto anteriormente. Apesar disso, os esforços continuam e a validação prática do sistema será realizada em breve.

7 CONCLUSÃO

Apesar do foco deste trabalho ser a criação de camarões marinhos, seu campo de aplicações é vasto, podendo ser utilizado, por exemplo, na criação de vários outros animais aquáticos como peixes, mariscos e ostras. Enfim, o sistema desenvolvido pode ser útil para aplicações similares onde haja a necessidade da gestão de variáveis hídricas, seja monitorando a qualidade da água de rios, lagoas, açudes, trechos de mar, ou até mesmo de piscinas, caixas d'água ou um pequeno aquário.

Além disso, a flexibilidade da ferramenta desenvolvida permite que uma pequena adaptação amplie ainda mais seu campo de atuação, utilizando-se diferentes sensores para a captação de inúmeras outras variáveis. Então, o sistema poderia, por exemplo, controlar a frequência de giros de um motor, o fluxo de fluídos num cano, a pressão no interior de um recipiente, a luminosidade de uma sala, além de várias outras aplicações.

Com relação às contribuições ao LCM, o sistema se mostra como uma ótima solução para alguns problemas encontradas no dia-a-dia do laboratório, tanto na parte financeira, eliminando os gastos com kits de testes e pagamentos de funcionários encarregados do registro das variáveis hídricas, como na parte operacional, diminuindo o tempo de resposta entre a percepção das variações nos parâmetros hídricos e as possíveis ações corretivas, reduzindo assim os danos à criação de camarões.

A elaboração do trabalho se mostrou bastante interessante, uma vez que o produto final integrou diferentes tecnologias como linguagens de programação de alto nível (JAVA, JSP, *JavaScript*, HTML), linguagem de baixo nível (*Assembly*),

microcontroladores (PIC 16F870), banco de dados (MySQL), além de uma certa noção de eletrônica. Isso exigiu conhecimentos específicos nem sempre abordados ao longo da graduação, tornando o desenvolvimento do sistema uma tarefa bem mais desafiadora.

8 REFERÊNCIAS BIBLIOGRÁFICAS

BRUSAMARELLO, Valner. **Instrumentação Aplicada à Engenharia**. Caxias do Sul: 2004.

DANTAS, Grossi Ricardo; PLAZAS, Rodrigo. **Transdutores de Temperatura**. Campinas: 2004.

DBSystems. **JAVA 2**. São Paulo: 2004.

DEITEL, H. M.; DEITEL, P. J. **JAVA Como Programar**. 3ª edição. Porto Alegre: Bookman, 2001.

FRANCESCHINI, André S., et al. **Funcionamento dos Sensores Térmicos na Medição da Temperatura em banho Quente**. Porto Alegre: 2003.

FUNDÃO DA COMPUTAÇÃO. **Introdução ao Mundo Java**. 2002.

GONZAGA, Flávio S.; BIRCKAN, Guilherme. **Curso de PHP e MySQL**. Florianópolis: 2000.

GÜTHS, Saulo; PAULO, Vicente de. **Instrumentação em Ciências Térmicas**. Florianópolis: 1998.

LCM (Laboratório de Camarões Marinhos). Disponível em <www.lcm.ufsc.br>. Acesso em: 12 fevereiro 2004.

MATIC, Nebojsa. **PIC Microcontrollers**. Disponível em: <http://www.mikroelektronika.co.yu/english/product/books/PICbook/0_Uvod.htm>. Acesso em: 12 fevereiro 2004.

MICROCHIP. **PIC16F870/871 Data Sheet**. Disponível em: <<http://www.micrchip.com>> . Acesso em: 3 junho 2004.

MySQL AB. **MySQL Reference Manual**. 2003.

OLIVEIRA, Rui de. Estudo e Determinação do pH. Disponível em: <http://www.profcupido.hpg.ig.com.br/ph_e_poh.htm> . Acesso em: 12 fevereiro 2004.

RAPCHAN, Francisco. **Banco de Dados**: Apostila de Notas de Aula. Espírito Santo: 2004.

RICARTE, Ivan Luiz Marques. **Organização de Computadores**. São Paulo: 1999.

ROCHA, Breno. **Generalidades: A Química do Corpo Humano VIII**. Disponível em: <http://www.corpohumano.hpg.ig.com.br/generalidades/quimica/quimica_08.html>. Acesso em: 12 fevereiro 2004.

SILBERSCHATZ, Abraham; KORTH, Henry F.; SUDARSHAN, S. **Sistema de Banco de Dados**. 3ª edição. São Paulo: MAKRON Books, 1999.

SOUZA, David José de. **Desbravando o PIC**: Ampliado e Atualizado para PIC 16F628A. 7ª edição. São Paulo: Érica, 2003.

APÊNDICE

SISTEMA AUTOMATIZADO DE MONITORAMENTO E CONTROLE REMOTO DE VARIÁVEIS HÍDRICAS

Marcelo de Barros Leiras

Graduando em Ciências da Computação
Universidade Federal de Santa Catarina

leiras@inf.ufsc.br

Lúcia Helena Martins Pacheco

Doutora em Engenharia de Produção
Universidade Federal de Santa Catarina

lucia@inf.ufsc.br

Resumo

Este trabalho tem o propósito de apresentar uma solução para alguns problemas inerentes no ciclo produtivo do Laboratório de Camarões Marinhos - LCM, pertencente à Universidade Federal de Santa Catarina - UFSC, através de um sistema automatizado de monitoramento e controle remoto da água dos tanques de camarões. Em contrapartida, o produto final resultante deste estudo pode ser utilizado para aplicações similares onde seja necessário o monitoramento remoto da qualidade da água.

Palavras-chave: Variáveis Hídricas, Automação, Controle, Monitoramento, Acesso Remoto, Camarões.

Abstract

This work aims to present a solution for some inherent problems in the productive cycle of the Laboratório de Camarões Marinhos (Laboratory of Marine Shrimps) - LCM, belonging to the Universidade Federal de Santa Catarina (Federal University of Santa Catarina) - UFSC. For that, it is proposed an automatic system of remote monitoring and control of the water of the tanks of shrimps. Also such system could be used for similar applications where the monitoring of water quality is necessary.

Key-words: Water Variables, Automation, Control, Monitoring, Remote Access, Shrimps.

Introdução

Nos últimos anos, o cultivo de camarões vem sendo bastante incentivado na região sul do Brasil. Um dos grandes responsáveis pelo desenvolvimento deste setor é o Laboratório de Camarões Marinhos (LCM), localizado nas dependências da UFSC, cuja principal atividade é o fornecimento de pós-larvas de camarões para fazendas de engorda.

Devido à grande sensibilidade dos camarões a variações bruscas de certos parâmetros da água como temperatura, pH, oxigênio dissolvido, amônia, entre outros, é de extrema importância que estas variáveis sejam monitoradas constantemente para que ações de correção das mesmas possam ser realizadas em tempo hábil, evitando que estes indivíduos sejam prejudicados.

Em vista disso, diariamente são realizados no LCM vários testes com reagentes químicos que utilizam uma pequena amostra de água do local analisado. Porém, a utilização em larga escala destes kits de testes é responsável por uma grande parcela nas despesas do produtor. Além disso, é necessário ainda manter um funcionário encarregado de fazer as medições periodicamente, desprendendo um tempo precioso que poderia estar sendo gasto em outras atividades.

Outro grave problema deste tipo de monitoramento da qualidade da água é o tempo decorrido até que se tome alguma atitude em resposta às variações dos parâmetros. Sem um controle em tempo real, qualquer incidente ocorrido no intervalo entre duas medições capaz de alterar os parâmetros da água só vai ser percebido na próxima análise. Então, poderá ser tarde demais para uma ação corretiva, ocasionando talvez até a morte dos indivíduos.

O propósito da realização deste trabalho é apresentar uma solução para alguns problemas inerentes no ciclo produtivo do LCM através de um sistema automatizado de monitoramento e controle remoto da água dos tanques de camarões. Em contrapartida, o produto final resultante deste estudo pode ser utilizado para praticamente qualquer finalidade em que seja necessário o monitoramento da qualidade da água.

Então, este documento apresenta os principais aspectos técnicos, metodológicos e práticos envolvidos na experiência piloto de automatização do sistema de monitoramento e controle da água dos tanques de criação do Laboratório de Camarões Marinhos. Tal experiência consistiu na elaboração de um módulo eletrônico contendo basicamente um microcontrolador, um circuito de aquisição de variáveis e um circuito de acionamento de dispositivos, além de alertas sonoros e visuais para a sinalização de possíveis anomalias nos parâmetros da água. Este módulo foi então ligado a um sistema de acesso via *Internet*, desenvolvido com a finalidade de monitorar e controlar a qualidade da água dos tanques remotamente.

Microcontroladores

SOUZA (2003) define o microcontrolador como sendo “um pequeno componente eletrônico, dotado de uma inteligência programável, utilizado no controle de processos lógicos”.

O controle de processos deve ser entendido como o controle de periféricos, tais como: LEDs, botões, displays de segmentos, displays de cristal líquido, resistências, relês, sensores, e muitos outros.

São chamados controles lógicos, pois a operação do sistema baseia-se nas ações lógicas que devem ser executadas, dependendo do estado dos periféricos de entrada e/ou saída.

O microcontrolador é programável, pois toda a lógica de operação é estruturada na forma de um programa e gravada numa memória dentro do componente. Neste trabalho foi escolhido um tipo de microcontrolador conhecido por PIC (*Peripheral Interface Controller*), fabricado pela empresa *Microchip*. A maioria dos PICs possui memória interna do tipo EPROM (*Erasable Programmable Read-Only Memory*). Tais tipos de memória permitem serem gravadas uma vez para PICs normais, ou várias vezes no caso de PICs janelados (que podem ser apagados mediante banho de luz ultravioleta, efetuado através da exposição da pastilha por uma janela existente em seu encapsulamento). Existem ainda modelos que possuem a memória de programa do tipo FLASH, que pode ser gravada várias vezes.

Depois de gravado o programa operacional na memória interna, toda vez que o microcontrolador for ligado, o programa interno será executado.

Quanto à inteligência do componente, podemos associá-la à ALU (*Arithmetic Logic Unit*), pois é nessa unidade que todas as operações matemáticas e lógicas são executadas. Quanto mais poderosa for a ALU, maior será a capacidade de processar informações.

E finalmente, o microcontrolador ganhou ainda o adjetivo “pequeno”, pois em uma única pastilha de silício temos todos os componentes necessários ao controle de um processo.

Banco de Dados

No trabalho em questão, o banco de dados é responsável por armazenar os valores de temperatura e pH levantados pelo módulo de aquisição de variáveis e enviados ao PC através do microcontrolador. Ainda, ele deve manter registradas todas as anomalias ocorridas nas medições destas variáveis, além de informações referentes ao acionamento de dispositivos.

SILBERSCHATZ (1999) define banco de dados como uma coleção de dados inter-relacionados, representando informações sobre um domínio específico. Um sistema de banco de dados envolve os próprios dados, o *hardware* no qual eles residem e o *software* que controla o armazenamento e a recuperação de dados, chamado sistema gerenciador de banco de dados (SGBD).

Existem basicamente três tipos de modelos de dados, definidos por RAPCHAN (2004) como uma coleção de ferramentas conceituais para a descrição de dados, relacionamentos de dados, semântica de dados e restrições de integridade (consistência). São eles: Modelos lógicos baseados em objetos, Modelos lógicos baseados em registros e Modelos Físicos de Dados.

O modelo adotado para a realização deste trabalho é o relacional, que consiste em uma coleção de tabelas, cada uma com um nome único. Uma linha em uma tabela representa um relacionamento entre um conjunto de valores. Uma vez que essa tabela é uma coleção de tais relacionamentos, há uma estreita correspondência entre o conceito de tabela e o conceito matemático de relação, a partir das quais se origina o nome desse modelo de dados.

Sistema via Internet

O sistema que permite o acesso remoto às informações armazenadas no banco de dados, bem como o acionamento de dispositivos, foi desenvolvido utilizando algumas técnicas e linguagens de programação como, por exemplo, Java, JSP (*Java Server Pages*), JavaBeans, JavaScript e HTML (*Hyper Text Markup Language*).

Variáveis Hídricas

As variáveis hídricas escolhidas para este estudo foram a temperatura e o pH.

A temperatura é uma manifestação da energia cinética em um corpo. Associado ao conceito de temperatura (e com ele freqüentemente confundido) está o conceito de calor, que é a energia que aquece um corpo e eleva sua temperatura. (FRANCESCHINI, 2003).

O pH (Potencial Hidrogeniônico) é um termo utilizado para expressar a intensidade da condição ácida ou básica de um determinado meio. É definido como o logaritmo negativo da concentração hidrogeniônica $[H^+]$. Intimamente ligado a este conceito está o pOH (Potencial Oxidriônico ou Hidroxiliônico) que, por analogia, é definido como o logaritmo negativo da concentração hidroxiliônica $[OH^-]$.

Metodologia e Descrição do Sistema

A Figura 1 esquematiza a metodologia adotada.

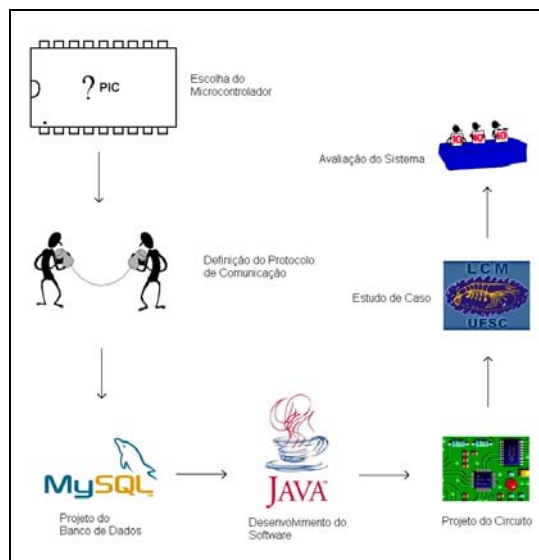


Figura 1 - Metodologia de desenvolvimento do trabalho

Inicialmente foi realizado um estudo aprofundado sobre os microcontroladores PIC, da empresa *Microchip*, analisando as características de cada uma das famílias de PICs disponíveis no mercado para se fazer a escolha do modelo mais adequado ao projeto. Buscou-se um microcontrolador que possuísse no mínimo uma interface de comunicação serial, possibilitando a transferência de dados para o PC, além de conversores A/D, para a leitura dos sensores de temperatura e pH. Porém, caso

houvesse indisponibilidade de algum destes requisitos, o problema seria contornado via *software* ou até mesmo com o auxílio de dispositivos eletrônicos externos.

A etapa seguinte foi o desenvolvimento de um protocolo de comunicação entre o PIC e o PC, para que os dados coletados pelos sensores pudessem ser transmitidos de maneira organizada para o PC e armazenados no banco de dados. Foi definido um padrão de transferência, estabelecendo bits de controle, ordem de envio de variáveis (pH e temperatura), número de bits de cada parâmetro, entre outros.

Uma vez definido este protocolo, foi desenvolvido um banco de dados para o armazenamento das informações adquiridas pelos sensores, permitindo que as mesmas pudessem ser consultadas futuramente.

Visando tornar a manipulação dos dados mais amigável ao usuário, foi elaborado um software através do qual poderão ser consultadas todas as informações contidas no banco de dados, por meio de tabelas e gráficos, bastando ter acesso a um computador conectado à Internet. Pela interface gráfica do programa o usuário pode ainda acionar remotamente dispositivos eletrônicos como aquecedores e bombas de gás carbônico (CO₂), caso seja necessária alguma correção emergencial nos parâmetros de qualidade da água, além de verificar os valores dos mesmos em tempo real. Para tornar isto possível, foi criado um circuito eletrônico que faz a “ponte” entre os sensores e o PIC, transformando os dados de temperatura e pH em níveis de tensão, compreensíveis pelos conversores A/D do microcontrolador.

O módulo construído foi fixado num tanque de camarões de um dos setores produtivos do LCM, e conectado a um computador com acesso à Internet. Os parâmetros da qualidade da água foram então acompanhados durante um determinado intervalo de tempo e, ao final, foi feita uma avaliação geral do sistema.

A Figura 2 ilustra todos os módulos que compõem o sistema desenvolvido, bem como as relações existentes entre os mesmos.

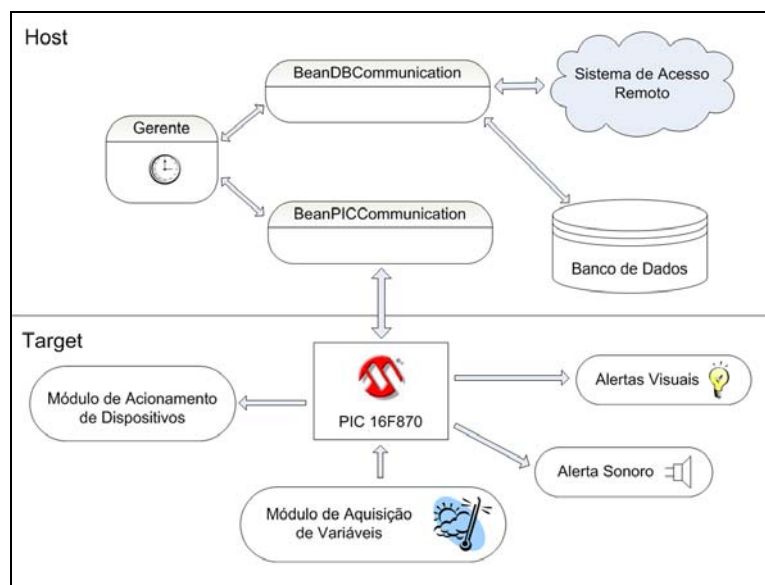


Figura 2 - Visão geral do Sistema

Host

O Host (Figura 2) corresponde aos componentes do sistema localizados “do lado” do PC. Ele é composto pelo banco de dados (MySQL), pelo sistema de acesso remoto e por um software denominado Gerente, juntamente com duas classes que fazem a comunicação com o banco de dados e com o PIC

O Gerente é um componente fundamental para que as interações entre os módulos aconteçam corretamente. Trata-se de um software responsável pela troca de informações entre o PIC e o PC e por parte do armazenamento de dados. Para realizar suas tarefas ele faz uso de duas classes chamadas BeanPICCommunication e BeanDBCommunication.

O Sistema de Acesso Remoto possibilita o monitoramento e o controle via Internet das variáveis hídricas.

Target

Ao contrário do Host, o Target (Figura 2) corresponde aos componentes do sistema localizados “do lado” do PIC, sendo composto pelo próprio microcontrolador e pelo circuito mínimo requerido para seu funcionamento (alimentação, oscilador e reset), além dos módulos de leitura de variáveis e acionamento de dispositivos.

O microcontrolador é o elemento responsável pelo controle dos processos de aquisição de variáveis e acionamento de dispositivos, além de realizar toda o tipo de troca de informações com o PC.

Como não foi feita a aquisição do sensor de pH, o módulo de leitura de variáveis ficou limitado a aquisição de temperatura. Mesmo assim, todo o sistema foi desenvolvido como se este sensor estivesse presente, bastando algumas pequenas modificações para que a coleta desta variável seja agregada.

Este módulo é responsável pelo acionamento do aquecedor e da bomba de CO₂ para o ajuste dos valores de temperatura e pH.

Conclusão

Apesar do foco deste trabalho ser a criação de camarões marinhos, seu campo de aplicações é vasto, podendo ser utilizado, por exemplo, na criação de vários outros animais aquáticos como peixes, mariscos e ostras. Enfim, o sistema desenvolvido pode ser útil para aplicações similares onde haja a necessidade da gestão de variáveis hídricas, seja monitorando a qualidade da água de rios, lagoas, açudes, trechos de mar, ou até mesmo de piscinas, caixas d’água ou um pequeno aquário.

Além disso, a flexibilidade da ferramenta desenvolvida permite que uma pequena adaptação amplie ainda mais seu campo de atuação, utilizando-se diferentes sensores para a captação de inúmeras outras variáveis. Então, o sistema poderia, por exemplo, controlar a frequência de giros de um motor, o fluxo de fluídos num cano, a pressão no interior de um recipiente, a luminosidade de uma sala, além de várias outras aplicações.

Com relação às contribuições ao LCM, o sistema se mostra como uma ótima solução para alguns problemas encontradas no dia-a-dia do laboratório, tanto na parte financeira, eliminando os gastos com kits de testes e pagamentos de funcionários encarregados do registro das variáveis hídricas, como na parte operacional, diminuindo o tempo de resposta entre a percepção das variações nos

parâmetros hídricos e as possíveis ações corretivas, reduzindo assim os danos à criação de camarões.

A elaboração do trabalho se mostrou bastante interessante, uma vez que o produto final integrou diferentes tecnologias como linguagens de programação de alto nível (JAVA, JSP, *JavaScript*, HTML), linguagem de baixo nível (*Assembly*), microcontroladores (PIC 16F870), banco de dados (MySQL), além de uma certa noção de eletrônica. Isso exigiu conhecimentos específicos nem sempre abordados ao longo da graduação, tornando o desenvolvimento do sistema uma tarefa bem mais desafiadora.

Bibliografia

FRANCESCHINI, André S., et al. Funcionamento dos Sensores Térmicos na Medição da Temperatura em banho Quente. Porto Alegre: 2003.

LCM (Laboratório de Camarões Marinhos). Disponível em <www.lcm.ufsc.br>. Acesso em: 12 fevereiro 2004.

RAPCHAN, Francisco. **Banco de Dados**: Apostila de Notas de Aula. Espírito Santo: 2004.

SILBERSCHATZ, Abraham; KORTH, Henry F.; SUDARSHAN, S. **Sistema de Banco de Dados**. 3ª edição. São Paulo: MAKRON Books, 1999.

SOUZA, David José de. **Desbravando o PIC**: Ampliado e Atualizado para PIC 16F628A. 7ª edição. São Paulo: Érica, 2003.