CENTRO UNIVERSITÁRIO DE BRASÍLIA – UNICEUB
FACULDADE DE TECNOLOGIA E CIÊNCIAS SOCIAIS APLICADAS - FATECS
CURSO DE ENGENHARIA DE COMPUTAÇÃO
PROJETO FINAL

Alex Perez dos Santos

CONTROLE DE ILUMINAÇÃO VIA COMANDO DE VOZ

Brasília - DF

Dezembro 2008

Alex Perez dos Santos

CONTROLE DE ILUMINAÇÃO VIA COMANDO DE VOZ

Monografia apresentada à banca examinadora para a conclusão do curso e obtenção do título de bacharel em Engenharia da Computação do Centro Universitário de Brasília – UniCEUB.
Orientador: Profº José Julimá Bezerra Junior

Brasília - DF

Dezembro 2008

AGRADECIMENTOS

Em primeiro lugar agradeço a Deus por estar sempre presente em minha vida, e por me mostrar o melhor caminho nos momentos difíceis.

Agradeço muito aos meus pais, irmão e namorada, por me incentivar, pela compreensão nos momentos em que estive ausente em todos esses anos e por toda a confiança creditada em mim.

Ao meu orientador, José Julimá Bezerra Junior, pelo profissionalismo, interesse e atenção na orientação desse projeto.

Aos meus colegas do TRF e do UniCEUB, pelo incentivo e pela colaboração durante o desenvolvimento do projeto.

Obrigado aos funcionários e aos professores do UniCEUB, em especial ao coordenador do curso professor Abiézer, ao professor Javier e à professora Marony, pela atenção e dedicação durante todos esses anos.

RESUMO

O projeto tem como objetivo desenvolver uma solução para automatizar o controle de iluminação residencial através do comando de voz, tendo como foco ajudar os portadores de deficiências motoras, tornar o ambiente residencial mais moderno e confortável para os usuários. Nesta solução, o usuário informa ao sistema de tratamento de dados um comando: “acende” ou “apaga” e o ambiente que será controlado “sala”,”suíte”,”cozinha” ou “banheiro”. Se o comando for encontrado na base de dados é enviado um dado para interface digital que acionará as lâmpadas desejadas.

Palavras-Chave: Automação Residencial, Comando de Voz, Linguagem de Programação Java, Porta Paralela.

ABSTRACT

The project aims to develop a solution to automate the control of residential lighting through voice command with the focus to assist those who have motor deficiencies, making the residential environment more modern and comfortable for users. The user orders the data processing system a command: “acende” or “apaga” and the environment that will be controlled “sala”, “suíte”, “cozinha” or “banheiro”. If the command is found in the database, it´s sent a data to the digital interface that will trigger the desired lamps.

Key-words: Residential Automation, Voice Command, Java Programming Language, Parallel Port.

SUMÁRIO

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ABREVIAÇÕES E SIGLAS

API	Application Programming Interface.
Aureside	Associação Brasileira de Automação Residencial.
dB	Decibel. Unidade de medida do nível de intensidade do som.
DB-25	Conector de 25 pinos da porta paralela.
IDE	Integrated Development Environment (ambiente de desenvolvimento integrado).
JDBC	Java Database Connectivity (API para acesso ao banco de dados)
JVM	Java Virtual Machine (máquina virtual java).
mA	mili Ampère. Unidade de medida de corrente elétrica.
SGBD	Sistema Gerenciador de Banco de Dados.
V	Volts. Unidade de medida de tensão.
0x	Representação do número em hexadecimal.

LISTA DE TRECHOS DE CÓDIGO

CAPÍTULO 1. INTRODUÇÃO

Neste Capítulo são apresentados os motivos que definiram o tema do projeto, uma descrição sobre onde se quer chegar com essa implementação, além dos objetivos específicos. Por fim é apresentada a visão geral do projeto e uma descrição objetiva de cada capítulo da monografia.

1.1 Motivação

A dificuldade de acionar interruptores por portadores de deficiências locomotoras, a busca pelo conforto, e a necessidade de modernização do ambiente residencial foram os grandes motivadores para a implementação do projeto.

Através da união entre software e hardware foi possível desenvolver essa solução com base em todo aprendizado no curso de Engenharia da Computação do UniCEUB.

1.1.1 Objetivos gerais

O projeto tem como objetivo desenvolver uma solução para automatizar o acionamento de lâmpadas e luminárias de uma residência por comando de voz. Com essa solução o usuário se livra da tarefa de se deslocar para acionar o interruptor e ligar a lâmpada, proporcionando conforto e comodidade.

1.1.2 Objetivos específicos

Essa solução não se aplica somente aos portadores de algum tipo de deficiência motora e usuários que buscam conforto, mas a todos que desejem um ambiente residencial mais moderno e prático, onde o simples fato de acionar uma lâmpada traga por trás de si todo um trabalho intelectual e uma implementação informatizada.

1.2 Visão geral do projeto

Através de um comando de voz o sistema irá ligar ou desligar uma lâmpada, proporcionando conforto e facilidade aos usuários. A figura 1.1 mostra a topologia do projeto.

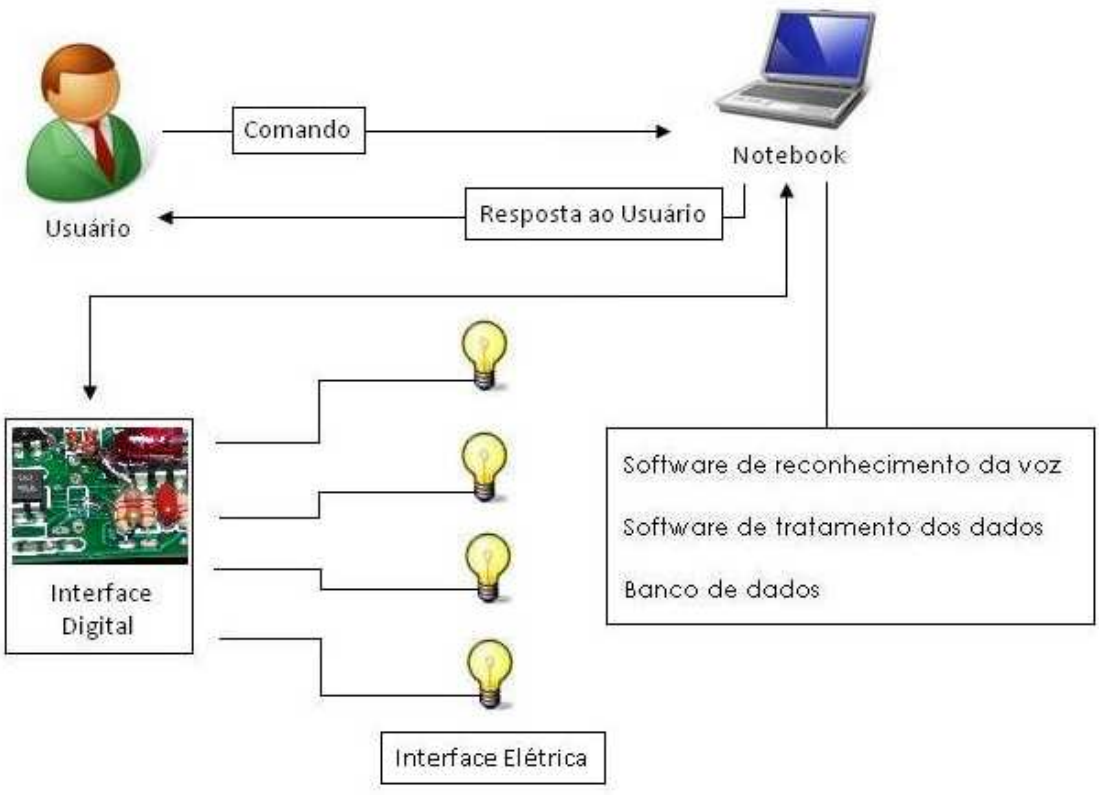

Figura 1.1 Diagrama Geral do Projeto

O microfone será ligado diretamente no servidor que por sua vez deverá ser alocado em uma parte estratégica da residência, isto é, só existirá uma central de comando para o acionamento das lâmpadas. O computador hospedará a aplicação de entrada de dados que irá fazer a comunicação com a porta paralela.

Como o projeto será implementado com mais de uma lâmpada, o sistema terá um menu visual que informará ao usuário qual comando deverá ser dito para acender ou apagar determinada lâmpada, conforme mostra a figura 1.2.

Comando	***Ação***
Acende sala	*O sistema irá acender a lâmpada da sala*

Figura 1.2 Menu de comandos

Cada comando para acionar a lâmpada de determinado cômodo será previamente cadastrado pelo administrador do sistema no software de tratamento dos dados e na base de dados de acordo com sua ligação no circuito digital, ou seja, uma vez cadastrados os comandos o usuário não terá a possibilidade de alterar a configuração do sistema**.** O sistema irá comparar esse comando com os comandos pré cadastrados no banco de dados, se for aceito a aplicação irá enviar um pacote de dados binário pela porta paralela e será enviada uma tensão de 3,3 V para o buffer e 0 V para o relé fechando o circuito em 220 V, ligando então a lâmpada do ambiente solicitado.

Após efetivar o comando, o sistema irá gerar uma mensagem informando se o comando foi bem sucedido ou não. A resposta do sistema será fornecida através da leitura da porta paralela, retornando para interface gráfica o status *aceso ou apagado*.

Todos esses passos são detalhados nos seguintes capítulos da monografia:

- Capítulo 1: Introdução, motivação para a escolha do tema, objetivos e visão geral do projeto.
- Capítulo 2: Referencial teórico sobre automação residencial, reconhecimento da voz.
- Capítulo 3: Funcionamento do sistema de tratamento dos dados, como foi desenvolvido, descrição de alguns trechos do código. O software de reconhecimento da voz.
- Capítulo 4: As interfaces do circuito físico, a comunicação com a porta paralela, os componentes utilizados, e o protótipo desenvolvido.
- Capítulo 5: Resultados obtidos
- Capítulo 6: Conclusões finais, dificuldades encontradas e sugestões para projetos futuros.
- Apêndices: Documentação do código fonte

CAPÍTULO 2. REFERENCIAL TEÓRICO

Hoje a automação residencial já é uma realidade e tem despertado cada vez mais interesse das pessoas em adquirir essas soluções. A Internet e a mídia são os grandes responsáveis por esse fato.

Nesse capítulo é apresentado na seção 2.1 a tecnologia, o processo de evolução, e o cenário atual da automação residencial. Na seção 2.2 será feita uma abordagem específica para automação por comando de voz, tendo como foco o reconhecimento da voz.

2.1 Automação Residencial

É um sistema que torna o ambiente residencial mais confortável, sofisticado, e eficiente. Dentro desse segmento estão presentes o controle de iluminação, acionamento de equipamentos elétricos, a parte de climatização do ambiente, circuito integrado de TV, sistema de irrigação, sistema de alarme entre outros.

Segundo dados da Aureside (Associação Brasileira de Automação Residencial) o processo de evolução da automação residencial começou com os eletrodomésticos nos idos de 1960. Depois, a evolução natural que nos acompanhou década a década, com novidades que hoje se mostram quase primárias, mas que de certa forma deram demonstrações do que seria possível no campo da automação residencial.

Dois fatores contribuem para a procura dessa tecnologia e para o crescimento desse mercado – o primeiro é o ambiente residencial que permanece praticamente inexplorado para a implementação de sistemas de redes e de controle

- o outro é o apelo ao novo que aflora desse tema. Para hobistas e entusiastas a Automação Residencial (AR) é sinônima de equipamentos de iluminação automatizados e home-theaters incrementados. Para os que enxergam além, a AR representa novas descobertas, desafios e oportunidades. (AURESIDE, 2008)

A figura 2.1 apresenta uma visão geral da situação atual da automação residencial.

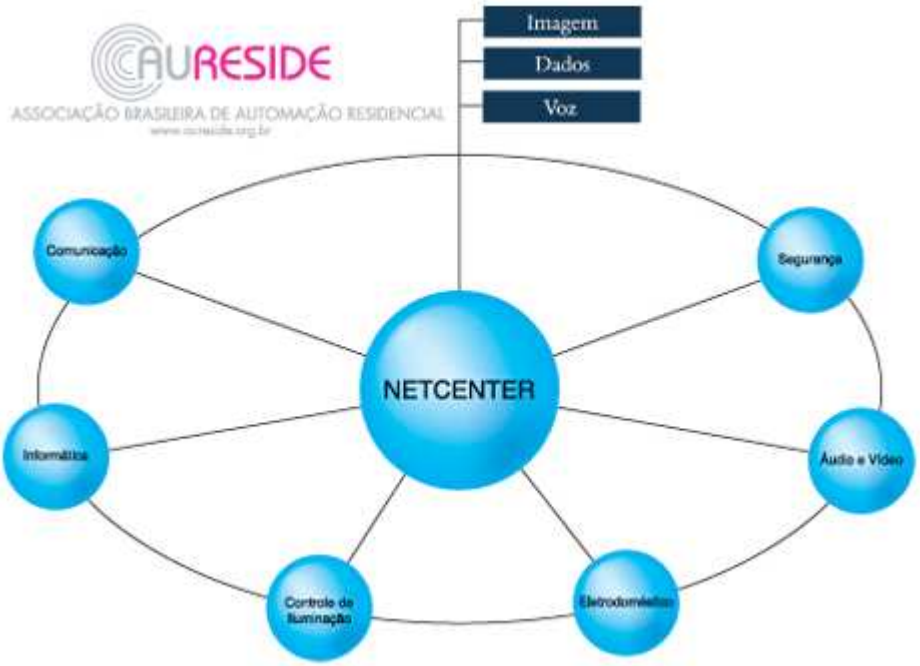

Figura 2.1 Esquema de Automação Residencial

O projeto está centralizado na parte de controle de iluminação utilizando o reconhecimento da voz. A implementação será feita a partir da união de software e hardware abordados detalhadamente no capítulo 3 e 4:

Software: Sistema de tratamento dos dados, desenvolvido em Java, banco de dados desenvolvido em MySQL e o ViaVoice da IBM para o reconhecimento da voz. Esses são citados respectivamente nas seções 3.1.4, 3.1.7, 3.1.1.

Hardware: Microfone para captar os comandos, notebook para hospedar os softwares citados e enviar e ler dados pela porta paralela, e o circuito digital para comunicação entre o microcomputador e o circuito elétrico com as lâmpadas. Esses estão nas seções 4.1, 4.3, 4.4.

2.2 Reconhecimento da Voz

A possibilidade de utilizar sistemas de reconhecimento de voz na automação residencial tem aumentado em muito pouco tempo. Nos anos passados, os esforços iniciais para utilizar o reconhecimento de voz eram inovadores e interessantes, mas lhes faltava confiabilidade e uma performance que possibilitasse ser um método viável de controle. O que aconteceu mais recentemente no mercado de computadores portáteis foi uma substancial redução de custo associada a um aumento significativo da capacidade de processamento. Foi esse o grande passo que tornou mais efetivo e viável o reconhecimento de voz.

Como resultado, os consumidores têm visto crescer o número de produtos oferecidos e melhora na qualidade dos já existentes. Boa parte destes produtos é utilizada por pessoas com problemas físicos, incapazes de acionar interruptores ou teclados ou de se deslocar livremente pela casa. Além disso, pode facilitar algumas funções que precisam ser executadas por crianças ou empregados não familiarizados com controles mais sofisticados.

Testes têm demonstrado que, embora funcionem razoavelmente bem para fazer menos e controlar um computador de mesa, todos estes produtos baseados em "ditados" necessitam que o microfone esteja bem próximo ao usuário para garantir um reconhecimento confiável. Muitos destes produtos requerem um

treinamento extensivo do usuário em relação ao vocabulário pré cadastrado no equipamento. Mesmo com todo treinamento, estes produtos nem sempre são confiáveis quando submetidos aos ruídos de som ambiente. Já existe muitos destes produtos destinados a automação residencial; no entanto, para uma operação satisfatória, a maioria deles requer um usuário de fala nítida, um ambiente silencioso e a proximidade de um microfone. Já os softwares controladores utilizados são muito confiáveis , uma vez que exaustivamente testados. (AURESIDE, 2008).

No projeto foram testados diversos softwares para o reconhecimento da voz, apenas o Via Voice 9.0 conseguiu atender as necessidades, uma vez que o sistema possui um considerável nível de confiabilidade. As configurações e o funcionamento do programa são explicados mais detalhadamente na seção 3.1.1.

CAPÍTULO 3. SOFTWARE DESENVOLVIDO

O projeto tem como proposta desenvolver uma solução de automação residencial para controlar a iluminação via comando de voz. A implementação do projeto está dividida em interface de entrada dos dados, interface digital e interface elétrica.

Neste capítulo será mostrado os softwares desenvolvidos, as tecnologias e as ferramentas utilizadas na implementação do projeto. Primeiramente será apresentado o funcionamento do sistema de tratamento dos dados, a implementação do banco de dados no MySQL, o software de reconhecimento de voz escolhido para o projeto, e alguns trechos do código fonte do sistema.

3.1 Software de reconhecimento de voz – IBM Via Voice

O projeto tem como objetivo controlar a iluminação de quatro ambientes de uma residência por comando de voz. Para o reconhecimento da voz foi utilizado o software proprietário da IBM, Via Voice Pro USB versão 9.0. Entre os programas testados este foi o que apresentou melhor desempenho no reconhecimento das palavras ditadas.

Por ser um software proprietário o seu algoritmo é desconhecido, porém, com base nas pesquisas realizadas, provavelmente o seu funcionamento se dá através de redes neurais, técnicas computacionais que apresentam um modelo matemático inspirado na estrutura neural de organismos inteligentes e que adquirem conhecimento através da experiência. (ICMC.USP, 2008).

Para o bom funcionamento do programa cada usuário deverá criar um perfil com seu padrão de voz, fornecendo ao assistente do usuário uma amostra de sua voz. Nas figuras 3.1 e 3.2 são mostradas as telas de criação, onde usuário deverá ditar no microfone o texto apresentado.

Figura 3.1 Tela de criação do padrão de voz

Figura 3.2 Tela de criação do padrão de voz

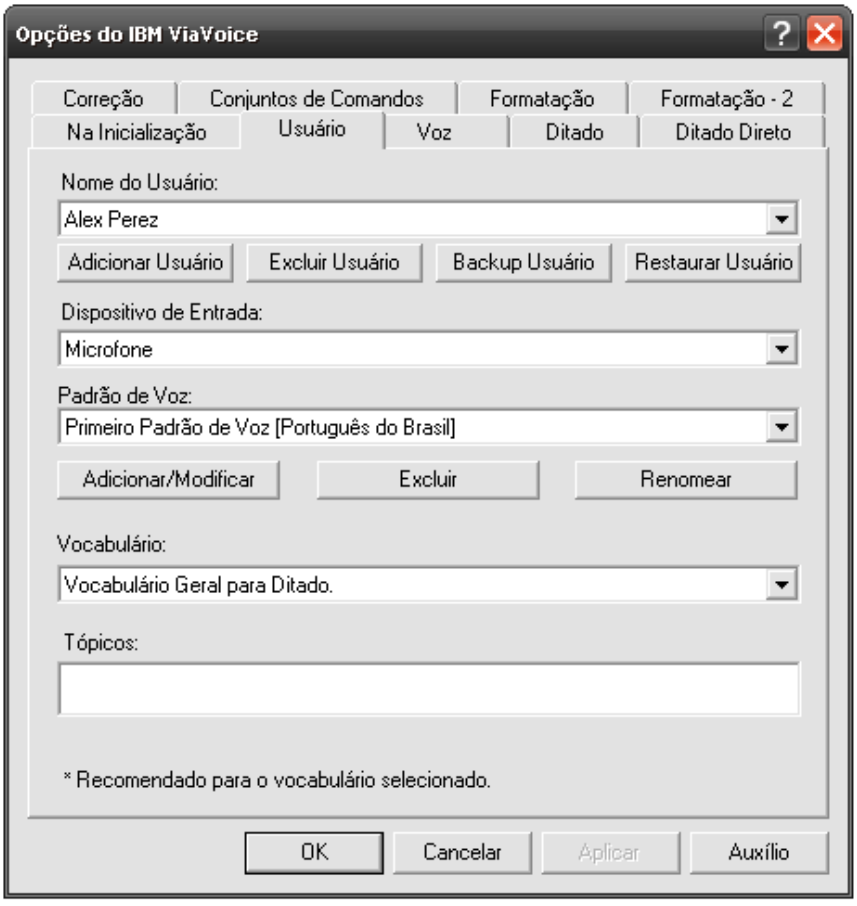

Figura 3.3 Tela do Via Voice Center

Após o processamento da voz gravada, o usuário já está apto a utilizar o Via Voice para capturar as palavras ditadas. O software possibilita ditar diretamente para o SpeakPad (processador de texto do próprio programa) ou para um aplicativo, sendo necessário apenas que este aceite entrada de texto. No projeto o sistema de tratamento dos dados possui o campo comando exclusivamente para receber a transcrição do que foi ditado para o Via Voice.

3.2 Linguagem de programação Java

A linguagem de programação Java foi criada por uma equipe da empresa Sun Microsystems e sua primeira versão foi lançada em maio de 1995. Seu foco principal foi o ambiente Web, onde se tornou popular pelas animações criadas nas páginas da Internet, conhecidas como Applets.

É uma linguagem orientada a objetos, muito semelhante ao C++ e sua idéia principal é entregar ao desenvolvedor uma plataforma simples, segura, e robusta. Uma das qualidades mais marcantes é a independência do sistema operacional instalado na máquina. Esse diferencial de execução em multi-plataformas se dá através da JVM (Java Virtual Machine) que interpreta o código bytecodes, gerado após um programa java ser compilado, permitindo assim a execução do programa.

Atualmente Java é uma das linguagens mais usadas e serve para qualquer tipo de aplicação, seja web, computadores pessoais, servidores, computadores de grande porte, jogos, aplicações móveis, chips de identificação, etc. (JavaFree, 2008).

3.3 IDE - Eclipse SDK e NetBeans

IDE é um ambiente integrado para o desenvolvimento de software, nele podemos codificar, compilar e executar programas. No projeto foram utilizados o Eclipse SDK e o NetBeans. Foram codificadas as classes TratamentoDados.java, PortaParalela.java e BancoDados.java no Eclipse. A classe Interface.java foi desenvolvida no IDE NetBeans, uma vez que esse ambiente proporciona ao desenvolvedor uma série de funcionalidades para criação da interface gráfica do sistema. (Netbeans, 2008)

Ambos os softwares são gratuitos e estão disponíveis para download nos sites: *www.eclipse.org/* e *www.netbeans.org/.*

3.4 Sistema de Tratamento dos Dados

O sistema de tratamento do dados tem como objetivo receber os dados capturados pelo programa de reconhecimento de voz, fazer a comparação com todos os comandos armazenados no banco de dados, enviar o pacote de dados através da porta paralela, e informar ao usuário se o comando foi efetivado.

A figura 3.4 mostra a tela do sistema de tratamento de dados desenvolvido em linguagem Java.

Figura 3.4 Tela do software de tratamento de dados

A interface gráfica do sistema está dividida em quatro partes que serão descritas abaixo:

1 - Menu: Apresenta ao usuário todos os comandos válidos para executar determinada ação. Está dividido em duas colunas:

- *Função*: Mostra qual palavra deve ser pronunciada para acender ou apagar determinada lâmpada.
- *Ambiente*: Mostra qual palavra deve ser pronunciada para definir qual ambiente será controlado.

2 - Status: Informa ao usuário se o comando foi efetivado ou não.

3 - Comando: O campo comando é responsável por receber as palavras pronunciadas e enviá-las para comparação com os comandos válidos. O botão “Iniciar” ativa a captura dos comandos, e estabelece conexão com o banco de dados e a porta paralela LPT1.

4 - Registros: A área de registros é destinada a informações do sistema para o usuário. O botão “Limpar Registros” limpa a campo de registros.

A figura 3.5 mostra a barra de menu, dividida em três itens:

1 - Comandos/Iniciar: Inicia a captura das palavras faladas.

2 - Comandos/Sair: O sistema é finalizado.

3 - Ajuda/Como utilizar?: Abre a tela de ajuda com instruções para o uso do sistema.

Figura 3.5 Barra de Menu

A tela da figura 3.6 mostra o sistema quando o botão “Iniciar” é acionado.

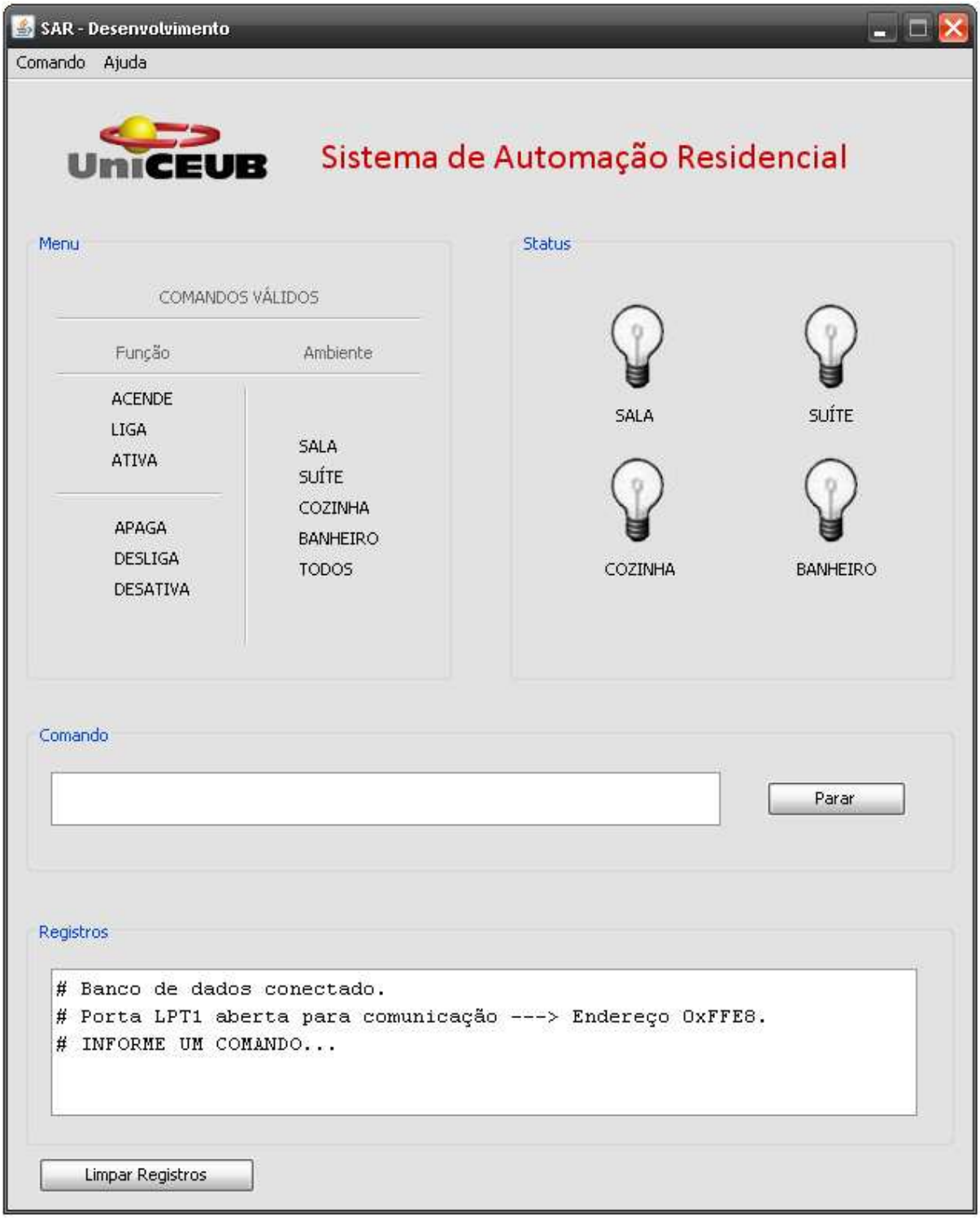

Figura 3.6 Tela após o botão Iniciar ser acionado

Quando esse evento ocorre a interface gráfica sofre as seguintes alterações:

- O botão “Iniciar” agora é “Parar”

- O campo de registros informa que as conexões com o banco de dados e com a porta paralela LPT1 foram estabelecidas.

Após a inicialização do sistema o usuário está apto a ditar os comandos no microfone. Se o comando for inválido o campo de registro fornece ao usuário a mensagem de que o comando não foi aceito, como pode-se observar na figura 3.7

Figura 3.7 Processamento de um comando inválido

Caso o comando seja encontrado na base de dados, a área de registros informará ao usuário que o comando foi aceito. Em seguida o sistema verifica o estado da porta paralela, se o sinal elétrico foi realmente enviado para a lâmpada certa, a área de status é alterada (figura 3.8).

Figura 3.8 Processamento de um comando válido

3.5 Descrição do código fonte

O objetivo desta seção é facilitar o entendimento do funcionamento do software de tratamento dos dados. Serão apresentados os trechos de códigos mais importantes, o código fonte completo do sistema pode ser localizado nos apêndices.

3.5.1 Inicialização do sistema

Nos trechos de código 3.1 e 3.2 são mostradas as rotinas necessárias para a inicialização do sistema, entre elas a verificação do estado da porta paralela para apresentar ao usuário quais lâmpadas já estão ligadas e quais estão desligadas.

```
public TratamentoDados(){

        // Configura o layout da tela do sistema
        try{

        UIManager.setLookAndFeel("com.sun.java.swing.plaf.windows.WindowsLoo
                                                                kAndFeel");

        }catch(Exception ex){
             JOptionPane.showMessageDialog(null, "Erro ao abrir o sistema!",
                                                                "Erro", 0);
             System.exit(0);
        }
             Interface layout = new Interface();
             tela=layout;
             tela.setVisible(true);

             // Verifica o estado inicial das lâmpadas
             verificaEstadoInicial();

             // Cria as funcionalidades dos botões
             ativaFuncionalidadesBotoes();

}// Fim do construtor
```

Trecho de código 3.1 Construtor da classe TratamentoDados.java

```
        // Verifica o estado da porta LPT1
        public void verificaEstadoInicial(){

               lpt1.abrePorta();
```

```
            tela.jTextArea2.setText("Aperte o botão INICIAR para ativar a
                                                captura dos comandos...");
            tela.jTextArea2.setEditable(false);

            if(lpt1.getPort()==1){
                  tela.jLabel13.setIcon(newjavax.swing.
                  ImageIcon("D:\\Desenvolvimento\\Implementação
                  \\Projeto\\lampada_acesa.jpg"));
                  l1=true;
                  valor=1;
            }

            if(lpt1.getPort()==2){

                  tela.jLabel26.setIcon(newjavax.swing
                  ImageIcon("D:\\Desenvolvimento\\Implementação
                  \\Projeto\\lampada_acesa.jpg"));
                  l2=true;
                  valor=2;
            }
      }
```

Trecho de código 3.2 Método verificaEstadoInicial( )

3.5.2 Captura dos comandos ditados

Após o botão iniciar ser acionado o método leituraDados( ) é chamado para realizar a captura das palavras ditadas e efetivar o comando desejado. Primeiramente a conexão com o banco de dados e com a porta paralela LPT1 são estabelecidas, em seguida o sistema aguarda 0,5 segundo para capturar a frase ditada no campo comando e passar como parâmetro para o método consultaBanco( ).

```
public void leituraDados() {

      //Estabelece conexão com o banco de dados
      BancodeDados banco = new BancodeDados();

      //Habilita a porta paralela LPT1 para comunicação
      lpt1.abrePorta();

      //Inicia a leitura dos comandos
      while (true) {
```

```
        try {

                // Aguarda 0,5 segundo para armazenar o comando ditado
                Thread.sleep(500);
                fraseDitada = tela.jTextField1.getText();

                //Envia a frase ditada para comparação com os comandos
                válidos
                comando=banco.consultaBanco(fraseDitada);

                tela.jTextField1.setText("");

}
```

Trecho de código 3.3 Método leituraDados( )

Ao chamar o método consultaBanco( ) a frase ditada será comparada com os comandos armazenados no banco. Através do JDBC (Ver seção 3.1.6) uma query (consulta) é executada e temporariamente os dados das tabelas são armazenados nas variáveis comando, ambiente, idcomando, e idambiente. No trecho de código 3.4 o sistema verifica a ocorrência das variáveis comando e ambiente na frase ditada, caso ocorra, a variável status do tipo inteiro é retornada para identificar qual comando deverá ser efetivado.

```
        while (consulta.next()) {

                // As variáveis armazenam as linhas das tabelas do
                //banco

                String comando = consulta.getString("COMANDO");
                String ambiente = consulta.getString("AMBIENTE");
                int idComando = consulta.getInt("TIPOCOMANDO_IDTIPO");
                int idAmbiente = consulta.getInt
                                        ("TIPOAMBIENTE_IDTIPOAMBIENTE");

                // Se a frase falada tiver palavras válidas
                // o método retorna um valor para a identificação
                // de qual comando deverá ser efetivado

                if(cmd.contains(comando) && cmd.contains(ambiente)){
                     status = idAmbiente*idAmbiente*idComando;
                }
        }
        return status;
```

Trecho de código 3.4 Acesso ao banco de dados

A variável “comando” será testada com a estrutura de seleção múltipla switch , se o valor dessa variável for igual a algum dos valores apresentados na tabela 3.1, é enviado uma seqüência de bits (Ver seção 4.3.1) pela porta LPT1 para a efetivação do comando, isto se for verificado que a lâmpada não está ligada. Em seguida o sistema precisa informar ao usuário se o comando foi realmente executado, então o estado da porta paralela é verificado, caso os dados tenham sido enviados corretamente para as lâmpadas desejadas, o usuário recebe a resposta no tela de status do sistema.

```
//Verifica a variável comando
switch(comando){

    //Acende a lâmpada da sala
    case 16:
    banco.status=0;
    if(l1==false){
          valor=valor+1;
          lpt1.setPort(valor);

          if(lpt1.getPort()==valor){
             tela.jLabel13.setIcon(new javax.swing.ImageIcon
             ("D:\\Desenvolvimento\\Implementação\\Projeto\\
             lampada_acesa.jpg"));
             tela.jTextArea2.setText("O COMANDO FOI EFETIVADO
             COM SUCESSO!");
             l1=true;
          }
          else{
             tela.jTextArea2.setText("Erro na comunicação com a
             porta paralela LPT1.");
          }
    }
    else{
       tela.jTextArea2.setText("A lâmpada da sala já está ligada!");
       tela.jTextField1.setText("");
    }
    break;

    //Acende a lâmpada da suíte
    case 25:
    banco.status=0;
    if(l2==false){
          valor=valor+2;
          lpt1.setPort(valor);
```

```
            if(lpt1.getPort()==valor){
               tela.jLabel26.setIcon(new javax.swing.ImageIcon
               ("D:\\Desenvolvimento\\Implementação\\Projeto\\
               lampada_acesa.jpg"));
               tela.jTextArea2.setText("O COMANDO FOI EFETIVADO
               COM SUCESSO!");
               l2=true;
            }
            else{
               tela.jTextArea2.setText("Erro na comunicação com a
               porta paralela LPT1.");
            }
      }
      else{
         tela.jTextArea2.setText("A lâmpada da suíte já está ligada!");
         tela.jTextField1.setText("");
      }
      break;
```

Trecho de código 3.5 Comando sendo efetivado

Caso variável comando	*Ação*
16	Liga a lâmpada da sala
25	Liga a lâmpada da suíte
36	Liga a lâmpada da cozinha
64	Liga a lâmpada do banheiro
49	Liga todas as lâmpadas
32	Desliga a lâmpada da sala
50	Desliga a lâmpada da suíte
72	Desliga a lâmpada da cozinha
128	Desliga a lâmpada do banheiro
98	Desliga todas as lâmpadas

Tabela 3.1 Detalhamento dos comandos

3.5.3 Comunicação do sistema de tratamento dos dados com a porta paralela

O sistema de tratamento dos dados necessita de um meio para se comunicar com a interface digital, isto é, enviar e receber da dados para controlar a

iluminação dos ambientes desejados. Na implementação do projeto foi escolhida a porta paralela LPT1 para fazer essa comunicação (Ver seção 4.2).

3.5.3.1 Bibliotecas Parport e NTJava

Para realizar essa comunicação foram utilizadas as bibliotecas Parport e NTPortJava, que utilizam um padrão de programação para acesso a bibliotecas construídas com o código nativo do sistema (JNI – Java Native Interface). No projeto foi utilizado o sistema operacional Windows XP, porém essa tecnologia não permite o controle direto da porta paralela para enviar e receber dados. Para solucionar esse problema foi necessário utilizar a biblioteca NTPortJava para dar permissão no endereço 0xFFE8 (Ver seção 3.2.1) e então estabelecer a comunicação com a porta paralela. (Geocities, 2008).

Esses arquivos devem estar em caminhos específicos, a pasta parport deve ser adicionada no mesmo diretório do projeto e no caminho jdk1.6.0_02\lib, o arquivo parport.dll deve ser adicionado no caminho C:\WINDOWS\system32. A pasta NTPortJava também deve estar no diretório do projeto e os arquivos NTPort.dll e NTPortJava.dll no caminho C:\WINDOWS\system32.

O trecho de código 3.6 mostra as bibliotecas sendo importadas e a porta lpt1 sendo habilitada para comunicação.

```
import parport.ParallelPort;
import NTPortJava.NTPortJava;

public class PortaParalela {

	short endPorta=(short)0xFFE8;
	int dados;
	String erro="Erro na comunicação com a porta paralela";
```

```
ParallelPort porta = new ParallelPort(endPorta);


public void abrePorta(){
    try{
    NTPortJava nt = new NTPortJava();
    nt.EnablePorts(endPorta,endPorta);
    }catch(Exception e){
    JOptionPane.showMessageDialog(null,erro,"Erro",0);
    }
}
```

Trecho de código 3.6 Classe PortaParalela.java

3.5.3.2 Enviar e receber dados pela porta paralela

O trecho de código 3.7 especifica os métodos responsáveis por enviar e receber os dados da porta paralela.

```
// Envia dados pela porta paralela
public void setPort(int dados){
      try{
      porta.write(dados);
      }catch(Exception e){
        JOptionPane.showMessageDialog(null, erro, "Erro", 0);
      }

}

// Recebe dados pela porta paralela
public int getPort(){
      try{
      dados=porta.readOneByte(endPorta);

      }catch(Exception e){
        JOptionPane.showMessageDialog(null, erro, "Erro", 0);
      }
      return dados;
}
```

Trecho de código 3.7 Métodos setPort( ) e getPort( )

3.6 API – Java Application Programming Interface

O Java contém muitas classes predefinidas que são agrupadas em categorias de classes relacionadas denominadas pacotes. Juntos, esses pacotes são referidos como Java API ou biblioteca de classes java. (Deitel, 2005). Ao utilizar uma API, o desenvolvedor não precisa se preocupar com detalhes da implementação do software, mas apenas usar seus serviços.

3.7 JDBC – Java Data Base Connectivity

O JDBC é uma API criada pela Sun que oferece ao desenvolvedor a possibilidade de acessar o banco de dados com seu programa Java. Essa API fornece mecanismos para armazenar, organizar, recuperar e modificar dados de uma determinada base. Para estabelecer essa comunicação é necessário adicionar o Driver JDBC na aplicação, disponibilizado pelo fabricante do sistema de gerenciamento de banco de dados (SGBD) utilizado.

O trecho de código 3.8 mostra as configuração dos parâmetros para a conexão com o banco de dados. São especificados o nome do Driver JDBC, o IP do servidor que hospeda a base de dados, o nome do banco de dados, usuário e senha do administrador do SGBD.

```
String nomeDriver = "com.mysql.jdbc.Driver";
    Class.forName(nomeDriver);
    String servidor = "localhost";
    String bancodedados = "db4";
    String url = "jdbc:mysql://" + servidor + ":3306/" +
    bancodedados;
    Conexao = DriverManager.getConnection(url,"root",
    "database");
```

Trecho de código 3.8 Parâmetros de configuração JDBC

Além dessas configurações é necessário adicionar o driver mysql-connector-java 5.1.2-beta-bin.jar no caminho jre1.6.0_02\lib\ext .

3.8 Banco de dados em MySQL

O MySQL é um SGBD , que utiliza SQL (Structured Query Language – Linguagem de consulta estruturada) como interface. Nasceu na Suécia e as idéias para seu desenvolvimento surgiram em 1979 de dois suecos: David Axmark, Allan Larssson, e um finlandês: Michael Widenius, também conhecido como Michael “Monty” Widenius, que trabalhavam juntos como programadores numa pequena empresa chamada TcX. Em 1995 os três amigos iniciaram o desenvolvimento do MySQL pela Empresa MySQL AB e o lançamento da primeira versão oficial ocorreu no ano de 1996. (Manzano, 2007).

Esse sistema de gerenciamento está disponível através de esquema de licenciamento duplo, tanto como um software livre quanto como comercial. A sua versão freeware pode ser encontrada no site http://dev.mysql.com/downloads.

O objetivo do banco de dados no projeto é armazenar todos os comandos válidos para a comparação com as palavras ditadas pelo usuário. A figura 3.9 mostra uma das tabelas desenvolvidas.

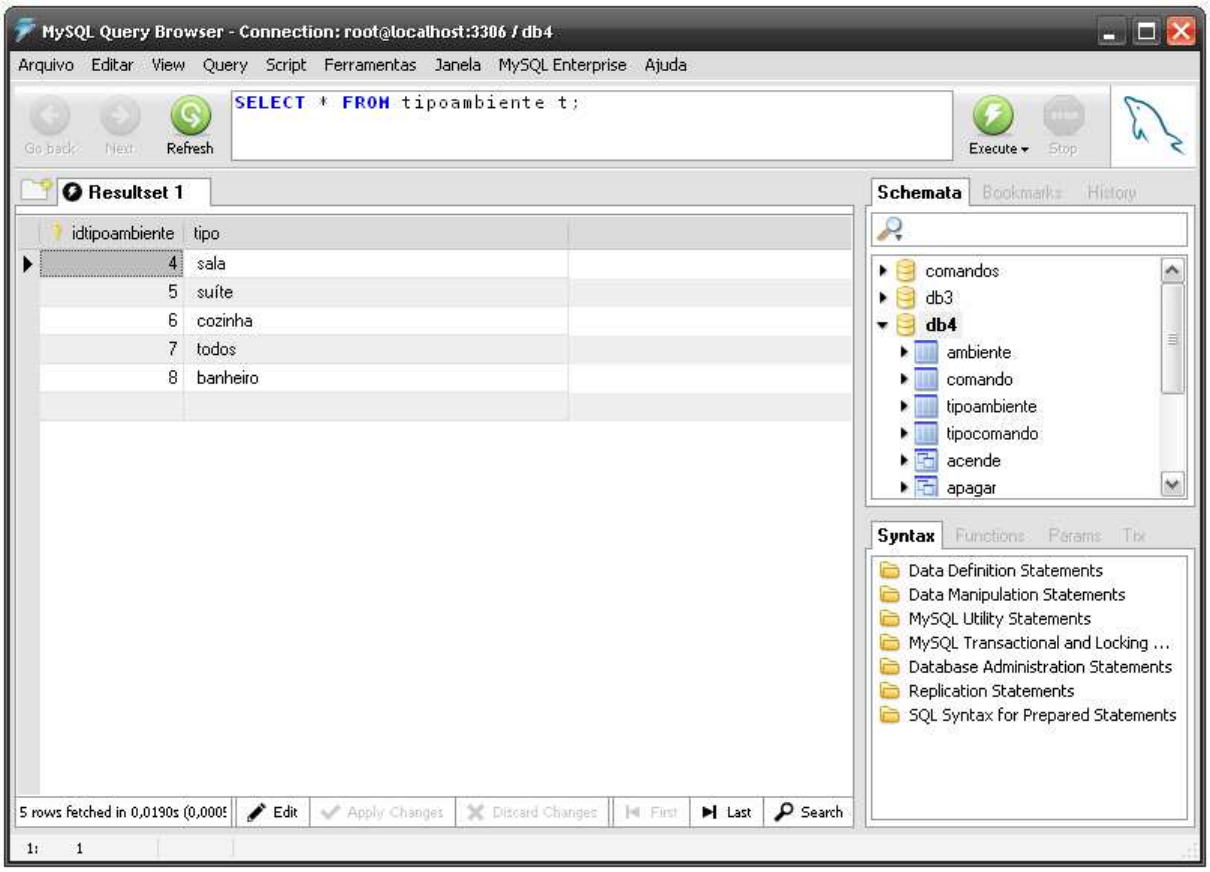

Figura 3.9 Tabela com os tipos de ambiente

Foram desenvolvidas quatro tabelas:

- Tipocomando: Armazena os dois tipos de comando acende/apaga.
- Tipoambiente: Armazena os tipos de ambiente sala, suíte, cozinha, banheiro.
- Comando: Armazena todas as palavras válidas para o tipo de comando.
- Ambiente: Armazena todas as palavras válidas para o tipo de ambiente.

Para criar a integração entre as informações das tabelas foram criados relacionamentos 1:N entre as tabelas tipocomando/comando e tipoambiente/ambiente. A figura 3.10 mostra o MER (Modelo Entidade-Relacionamento) do banco de dados criado.

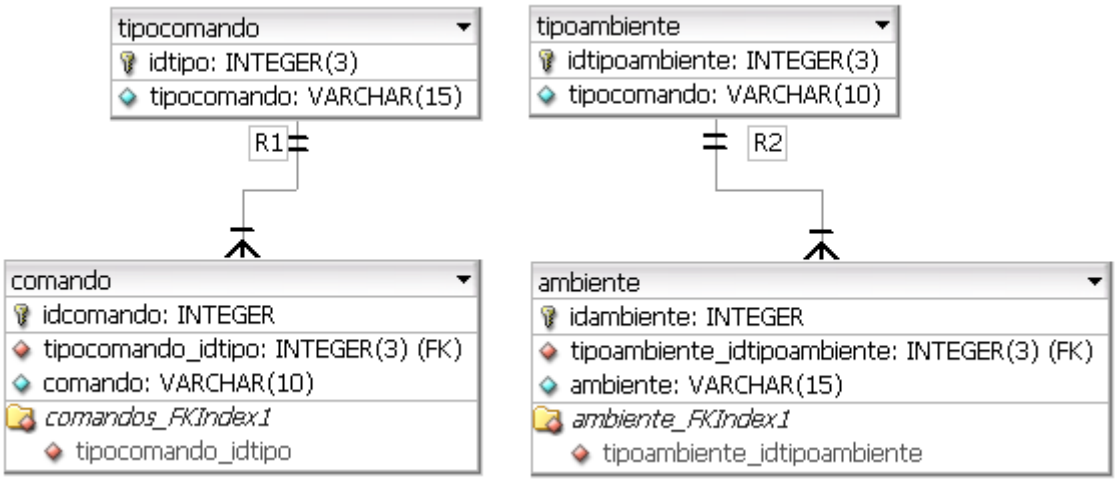

Figura 3.10 Modelo Entidade-Relacionamento

Além das palavras contidas no menu do sistema foram cadastradas palavras semelhantes as já existentes. Foram cadastradas também todas as palavras começando com letra maiúscula e minúscula. Esse procedimento garantiu uma melhora considerável no desempenho do programa (Ver seção 5.1.3). Na tabela 3.2 estão contidos todos os comandos válidos para acender ou apagar uma lâmpada de determinado ambiente.

Idcomando	Comando
1	Acende
2	Liga
3	Liga
5	Ativa
6	Aciona
8	Apaga
9	Desliga
11	Desativa
12	Acende
13	Ativa
14	Aciona
15	Desliga
17	Apaga
18	Desativa
19	Liça
20	Lica
21	Desatino
22	Desatino
23	Dês
24	Des
25	Sativa
26	Desde
27	Pagar
28	Pagar

Tabela 3.2 Tabela comando

Na tabela 3.3 são apresentadas as palavras válidas para definir qual ambiente será controlado.

Idambiente	Ambiente
14	Sala
16	Sala
22	Cozinha
24	Cozinho
27	Todo
28	Todo
36	Suíte
37	Suíte
38	Banheiro
39	Banheiro
40	Banheira
41	Banheira
42	Cozinha
43	Cozinho
44	Toda
45	Toda
50	Tudo
51	Tudo

Tabela 3.3 Tabela ambiente

O sistema de tratamento de dados executa o comando se verificar, na frase ditada pelo usuário, a existência da combinação de duas palavras das tabelas mostradas, independente da ordem em que elas são ditas.

CAPÍTULO 4. HARDWARE DESENVOLVIDO

Este Capítulo trata do hardware desenvolvido, interface digital e interface elétrica. A primeira interface é responsável por enviar os sinais de tensão para o circuito elétrico ativando ou desativando as lâmpadas. A interface elétrica é formada pelo circuito das lâmpadas e o relé. Na figura 4.1 é apresentado o circuito desenvolvido.

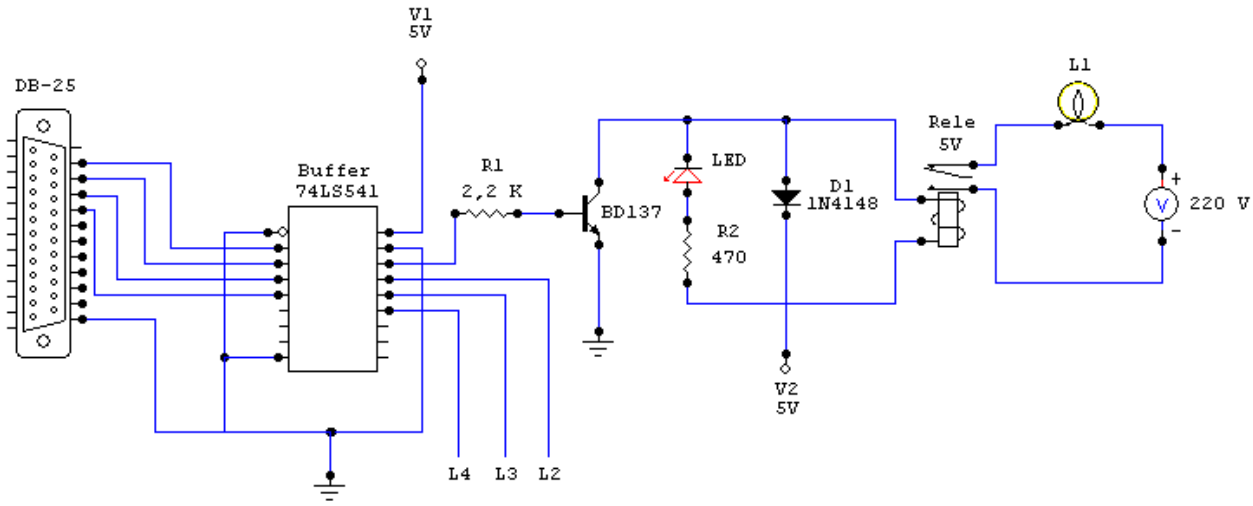

Figura 4.1 Circuito digital e elétrico

Os termos L2, L3, L4 são referentes as outra três lâmpadas que possuem os mesmos componentes do circuito de L1. Nas próximas seções esses componentes serão abordados detalhadamente.

4.1 Especificação do Microfone

O microfone será a interface de entrada do protótipo, através dele o usuário informará os comandos para controlar as lâmpadas. O software Via Voice irá reconhecer as palavras ditadas e transcrevê-las para o sistema de tratamento dos dados.

No projeto foi utilizado o microfone Clone de alta performance, a tabela 4.1 mostra suas especificações.

Caracteristicas	Valor
Sensibilidade	54 dB ± 2 dB
Impedância	2,2 K Ohms
Tensão de operação	3 Vdc
Frequência de resposta	50 Hz ~16 KHz

Tabela 4.1 Especificação do microfone

Fonte: Clone, 2008.

4.2 Porta Paralela

A porta paralela é uma interface de comunicação entre um computador e um periférico . Quando o primeiro PC (“Computador Pessoal”) foi criado pela empresa IBM, a idéia era ligar a porta a uma impressora. Hoje a porta paralela é utilizada para ligar diversos periféricos como scanners, zip drives, impressoras, adptadores de rede, entre outros. No projeto, a porta paralela é a ponte de comunicação entre o sistema de tratamento de dados e a interface digital, ela será responsável por enviar e receber dados do circuito digital. (Clube do Hardware, 2008).

O computador utiliza os termos LPT1, LPT2 e LPT3 para nomear a porta paralela, e ainda utiliza alguns endereços para enviar e receber dados. Como o notebook utilizado no projeto não possui a porta paralela foi necessário adquirir uma placa adaptadora PCMCIA\Porta Paralela, para funcionar exclusivamente como uma porta paralela, sendo a única diferença o endereço utilizado para a comunicação (figura 4.2). A tabela 4.2 mostra a porta paralela e seus endereços.

Interface	Nome da Porta	Endereço
Porta Paralela	LPT1	0x378
Porta Paralela	LPT2	0x278
PCMCIA\Porta Paralela	LPT1	0xFFE8

Tabela 4.2 Especificação da Porta Paralela

No projeto, a placa PCMCIA\Porta Paralela tem a única função de criar uma porta paralela no notebook, utilizando o nome LPT1 e o endereço 0xFFE8.

Figura 4.2 PCMCIA Parallel Card Bus

Fonte: Skycomp, 2008.

A porta paralela no computador possui o conector DB-25, representado na figura 4.3. Esse conector possui 25 pinos, 8 deles (2 até o 9) são utilizados para representar 8 bits de 1 Byte. No projeto, serão utilizados os pinos 2 a 5 para

transferir o pacote de dados binário, e o pino 18 para o GND (terra). Cada bit em nível alto “1” corresponde uma tensão de 3,3 V no pino, em nível baixo “0” a tensão é de 0 V.

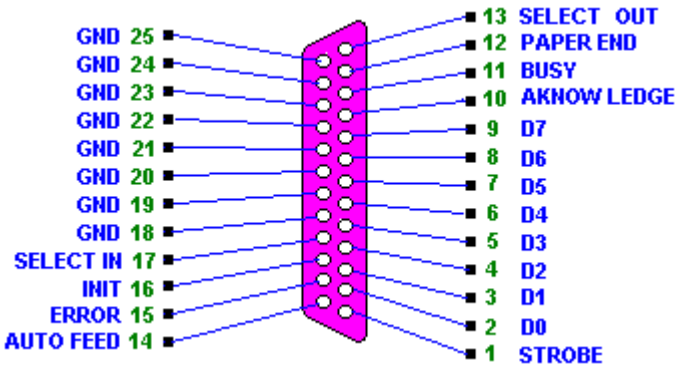

Figura 4.3 Conector DB25 (fêmea)

Fonte: Messias, 2006.

4.3 Interface digital

O objetivo da interface digital é estabelecer a comunicação entre o software de tratamento dos dados e a interface elétrica (ver seção 4.4), isto é, após o dado ser enviado pela porta paralela, o mesmo será manipulado pelo circuito digital e enviado para o circuito elétrico. A figura 4.4 mostra o funcionamento do circuito digital e os componentes utilizados para realizar essa comunicação.

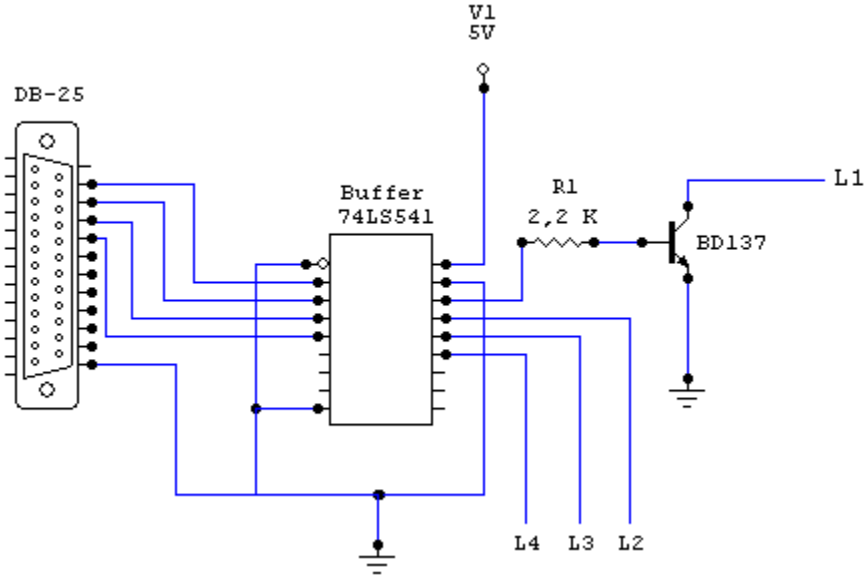

Figura 4.4 Circuito Digital

4.3.1 Buffer Octal 74LS541

O Buffer 74LS541 é o primeiro componente do circuito digital, a sua função é proteger a porta paralela de variações de corrente que possam vir do circuito elétrico onde a tensão é de 220 V. O buffer é composto de 8 entradas e saídas, no projeto serão utilizadas apenas 4 portas para enviar e receber dados da porta paralela.

A porta paralela enviará um pacote de dados de 4 bits, onde cada bit é responsável por controlar uma lâmpada, caso o valor do bit seja “1” a lâmpada é acionada, caso “0” é desligada. A tabela 4.3 mostra a tabela verdade do circuito:

Bits enviados pela porta paralela				Comando Efetivado
S_3	S_2	S_1	S_0	
0	0	0	1	Liga a lâmpada da sala
0	0	1	0	Liga a lâmpada da suíte
0	1	0	0	Liga a lâmpada da cozinha
1	0	0	0	Liga a lâmpada da banheiro
1	1	1	1	Liga todas as lâmpadas
1	1	1	0	Apaga a lâmpada da sala
1	1	0	1	Apaga a lâmpada da suíte
1	0	1	1	Apaga a lâmpada da cozinha
0	1	1	1	Apaga a lâmpada do banheiro
0	0	0	0	Apaga todas as lâmpadas

Tabela 4.3 Tabela Verdade do Circuito

4.3.2 Transistor BD137

A figura 4.5 mostra o próximo componente do circuito, o transistor BD137, no projeto o transistor funciona como um chaveador, está dividido em emissor, coletor, e base.

Em uma chave transistorizada, uma tensão positiva colocada na base do transistor faz com que o transistor entre na saturação. A tensão coletor-emissor do transistor, ou V_{ce}, cai para próximo de zero. Durante esse tempo, circula corrente através da carga. Quando a tensão é removida da base, o transistor entra no corte e a corrente cai para zero. Durante o corte, a tensão coletor-emissor é igual à tensão da fonte do coletor. (Lalond, 1999).

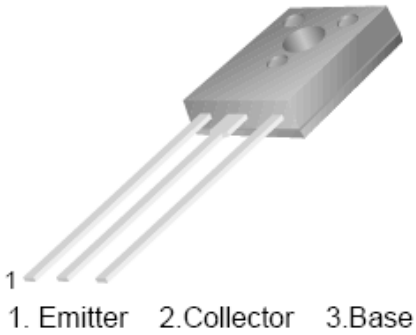

Figura 4.5 Transistor BD137
Fonte: Datasheet, 2008.

O transistor é responsável por controlar a tensão enviada para o relé, isto é, se for enviado um bit “1” para o buffer, a base do transistor receberá uma corrente de 1,21 mA, entrando na região de saturação. Nesse caso a tensão V_{ce} será aproximadamente 0 V e a corrente I_{ce} 90,5 mA, acionando o relé. Se o bit enviado for “0” o transistor entra na região de corte, nesse caso a tensão V_{ce} será

aproximadamente a mesma da fonte que alimenta o relé (5 V) e a corrente I_{ce} será 0 V, então o relé não será acionado.

4.4 Interface elétrica

A interface elétrica é composta pelos relés e as lâmpadas , quando a saída do buffer estiver em nível alto o relé fechará o circuito da lâmpada em uma tensão de 220 V. A figura 4.6 mostra o esquema do circuito elétrico.

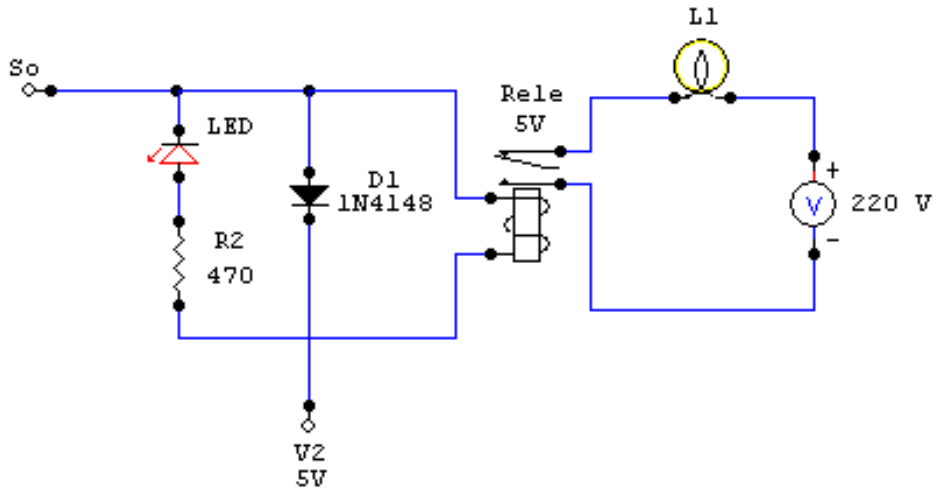

Figura 4.6 Circuito Elétrico

4.4.1 Circuito do Relé

A figura 4.7 mostra o relé, responsável por fechar e abrir o circuito da lâmpada, acendendo-a ou apagando-a.

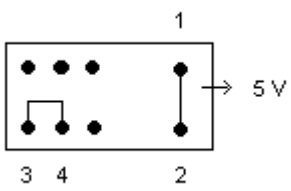

Figura 4.7 Esquema do Relé

Caso ocorra uma diferença de potencial de 5V entre os contatos 1 e 2, a chave existente entre os contatos 3 e 4 é fechada. Esses serão os pinos que ligarão as lâmpadas na tensão de 220V. A figura 4.8 mostra o circuito do relé.

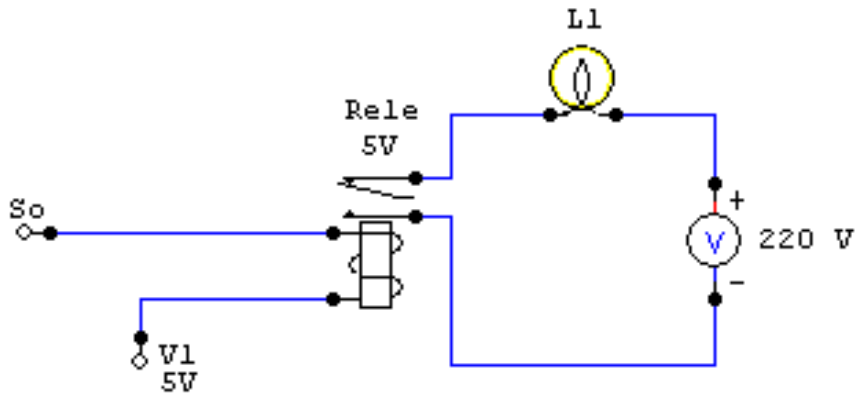

Figura 4.8 Circuito do Relé

Se a tensão na saída S_0 (proveniente do pino coletor do transistor) for de 0 V, a bobina do relé criará um campo magnético e os contatos serão ligados fechando o circuito da lâmpada, caso S_0 for aproximadamente 5 V não haverá diferença de potencial, então a chave do relé será aberta e a lâmpada será desligada.

4.4.2 Função do Diodo 1N4148

O diodo 1N4148 está inversamente polarizado em relação ao relé. Os elétrons do lado N (catodo) do diodo são atraídos para o terminal positivo do relé e as lacunas do lado P (anodo) do diodo serão atraídas pelo terminal negativo do relé. O material nesse caso apresenta características isolantes, pois devido ao aumento

da barreira de potencial, não haverá corrente, sendo sua resistência ôhmica de alto valor. (Capuano, 2002).

A função do diodo no circuito é proteger o transistor da carga indutiva do relé. Sempre que o relé é desligado, a energia armazenada é descarregada no diodo.

4.5 Protótipo Desenvolvido

A figura 4.9 mostra a foto do protótipo desenvolvido.

Figura 4.9 Foto do Protótipo

1 - Microfone

2 – Notebook com o software Via Voice 9.0, o sistema de tratamento dos dados e o banco de dados desenvolvido

3 - Interface Digital

4 - Interface Elétrica

CAPÍTULO 5. RESULTADOS

A proposta do projeto era desenvolver uma solução de automação residencial para controle de iluminação via comando de voz, integrando hardware e software. Com o sistema de tratamento dos dados e a comunicação com a porta paralela funcionando, foram realizados os primeiros testes dos comandos com leds. Este Capítulo apresentará os resultados obtidos e os procedimentos realizados para corrigir alguns desses resultados que comprometiam o bom funcionamento do projeto.

5.1 Identificação dos Ambientes

No início, a diferenciação dos ambientes da residência era feita por números, porém quando um comando era ditado com um número, no próximo comando o software Via Voice repetia o último número ditado. Por exemplo: O comando “liga um” é ditado, se o usuário quisesse informar outro comando por exemplo, “acende dois”, o Via Voice transcreveria para o sistema de tratamento dos dados “um acende dois”, inviabilizando a efetivação dos comandos. Outra maneira de identificar os ambientes foi implementada, utilizando as palavras “sala”, “quarto”, “cozinha”, e “banheiro”. O mesmo problema ocorreu com a palavra “quarto”, a solução foi trocar “quarto” por “suíte”.

5.2 Separação das Palavras Ditadas

Para armazenar as palavras ditadas, foi criado um método para capturar e separar as duas primeiras palavras da frase ditada. Em seguida essas palavras seriam comparadas com os comandos válidos do banco de dados, porém,devido aos ruídos do ambiente, nem sempre o que é ditado no microfone é fielmente transcrito, frequentemente o método de separação armazenava algumas palavras erradas. Por exemplo: “acende e a sala” , o método iria capturar P1= “acende” e P2= “e” e ignoraria o resto da frase. Esse problema foi resolvido facilmente com o método contains( ) da classe String. Este método faz uma verificação em toda a frase ditada comparando diretamente suas palavras com os comandos cadastrados na base de dados, independente da ordem em que as palavras forem ditadas. Esse procedimento foi de extrema importância para garantir o bom desempenho na efetivação dos comandos no sistema de tratamento dos dados.

5.3 Palavras Semelhantes

Durante os testes, foi observado que frequentemente algumas palavras, semelhantes as ditadas, eram transcritas para o campo comando do sistema de tratamento dos dados. Esses resultados ajudaram muito na aprimoração do reconhecimento dos comandos, como foi citado na seção 3.8, o procedimento realizado para melhorar a performance do programa foi adicionar todas essas palavras na base de comandos válidos.

5.4 Ambiente com ruído

Um fator importante na análise da confiabilidade do sistema é o nível de ruído do ambiente. Os testes realizados em um ambiente de conversação normal, com aproximadamente 60 dB de nível de intensidade do ruído, resultaram em algumas falsas detecções. A figura 5.1 mostra o nível de ruído de alguns ambientes.

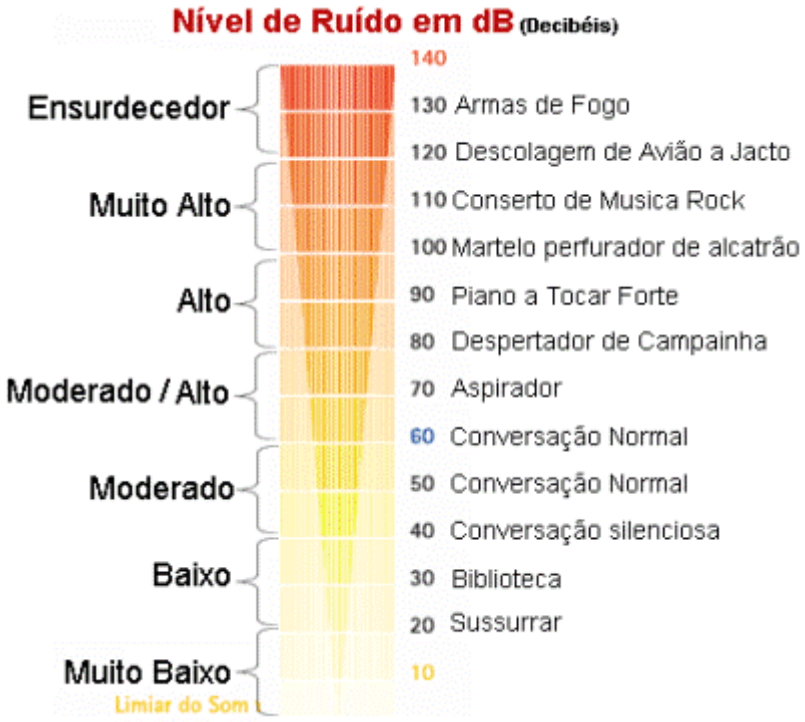

Figura 5.1 Nível de Ruído

Fonte: Megaclima, 2008.

Os dados acima e os testes em ambientes com ruído, comprovam que o ambiente de conversação silenciosa (40 dB) é o que apresenta melhor desempenho na utilização do Via Voice.

5.5 Reconhecimento da Voz

O Via Voice IBM mostrou ser um software extremamente eficiente no reconhecimento das palavras ditadas, levando em consideração que o usuário já possui seu padrão de voz cadastrado.

Foi realizado alguns testes de comando de voz utilizando o padrão de voz de outro usuário, no caso da voz ser masculina a taxa de falsos reconhecimentos não é muito alta, porém quando a voz do usuário é feminina, os resultados não são satisfatórios, comprometendo a confiabilidade do projeto. Para o perfeito funcionamento dessa solução de automação residencial é indispensável que todos os usuários da residência criem seu padrão de voz.

CAPÍTULO 6. CONSIDERAÇÕES FINAIS

A proposta do projeto foi desenvolver uma solução para acionar lâmpadas por comandos de voz, este último Capítulo apresentará os objetivos alcançados, conclusões finais, as dificuldades encontradas durante o desenvolvimento do projeto, e propostas para projetos futuros.

6.1 Dificuldades Obtidas

Durante o desenvolvimento do projeto foram encontradas algumas dificuldades que demandaram bastante tempo para serem solucionadas. O primeiro problema foi conseguir a comunicação do sistema de tratamento dos dados com a porta paralela, uma vez que os sistemas operacional Windows NT/2000/XP não permitem que o usuário controle diretamente a porta paralela. A solução foi utilizar o software UserPort para dar permissão ao usuário de controlar a porta LPT1 no endereço 0x378.

Esse problema foi solucionado, porém outro problema surgiu, o notebook utilizado não possui porta paralela. O início da solução desse problema foi adquirir uma placa conversora PCMCIA/PortaParalela. Em seguida foi observado que essa placa utilizava um endereçamento diferente (0xFFE8) para a transferência dos dados. Outro problema surgiu, o programa UserPort não tem suporte para essa faixa de endereço, a solução definitiva foi substituir esse software pela biblioteca NTPortJava, que concede essa permissão ao usuário.

6.2 Conclusões

Em síntese, os objetivos traçados foram cumpridos. A integração entre o software de reconhecimento da voz, o sistema de Tratamento dos Dados, a Interface digital e a Interface elétrica é a confirmação que os requisitos propostos foram atendidos.

Conforme citado na seção 5.1.5, a criação do padrão de voz para cada usuário no software Via Voice é extremamente necessário para o bom funcionamento da solução desenvolvida, com esse requisito sendo obedecido, o projeto implementado se mostra confiável e eficiente. Durante todos os testes finais, foi observado uma enorme praticidade em acionar as lâmpadas pelo comando de voz, proporcionando facilidade e conforto para o usuário.

O aprendizado durante o curso de Engenharia da Computação forneceu a base do conhecimento para o desenvolvimento desse projeto. Em especial as matérias Lógica Digital, Circuitos Eletrônicos, Circuitos e Máquinas Elétricas e Linguagem e Técnicas de Programação.

6.3 Sugestões para projetos futuros

Como o escopo do projeto é controlar a iluminação através do comando de voz, são propostas algumas sugestões para a continuidade do projeto:

- Acionamento de outros equipamentos elétricos por comando de voz.
- Implementar uma solução com vários microfones na residência e com a resposta dos comandos através de avisos sonoros.

- Aumentar o número de portas controladas através de demultiplexadores.
- Desenvolver um programa para o reconhecimento da voz através de redes neurais.

REFERÊNCIAS

Aureside – Associação Brasileira de Automação Residencial **[Home Page].** 2008. Disponível em: < http://www.aureside.org.br>. Acesso em 03 Set. 2008.

CAPUANO, Francisco Gabriel. **Laboratório de Eletricidade e Eletrônica.** São Paulo: Érica, 2005.

CLONE **[Home Page].** 2008. Disponível em: < http://www.clone.com.br>. Acesso em 22 Nov. 2008.

Clube do Hardware **[Home Page].** 2008. Disponível em: < http://www.clubedo hardware.com.br/artigos/1147/4 >. Acesso em 22 Ago. 2008.

DATASHEET **[Home Page].** 2008. Disponível em: < www.datasheetcatalog.net >. Acesso em 22 Nov. 2008.

DEITEL, Harvey M.; DEITEL, Paul J. **Java: Como programar.** 6. Ed. São Paulo: Pearson Prentice Hall, 2005.

Geocities **[Home Page].** 2008. Disponível em: < http://www.geocities.com/Juanga69 /parport/ >. Acesso em 20 Ago. 2008.

ICMC.USP, **[Home Page].** 2008. Disponível em: <http://www.icmc.usp.br/~andre/ research/neural/index.htm>. Acesso em 20 Ago. 2008.

Java Free **[Home Page].** 2008. Disponível em: <http://www.javafree.org /content/view.jf?idContent=84>. Acesso em 29 Ago. 2008.

LALOND, David E. ; ROSS, John A. **Princípios de Dispositivos e Circuitos Eletrônicos**. São Paulo: Makron Books, 1999.

LIMA, Leonardo Lins de Albuquerque. Monografia: Controle do aparelho de televisão via comando de voz utilizando microcontrolador. 2º semestre de 2007.

LUCENA, Gustavo Gomes. Monografia: Automação Residencial por comando de voz utilizando microcontrolador. 2º semestre de 2006.

MANZANO, José Augusto. **MySQL 5 Interativo: Guia básico de orientação e desenvolvimento.** São Paulo: Érica, 2007.

MEGACLIMA **[Home Page].** 2008. Disponível em: < http://www.megaclima.pt/informacoes_decibeis.htm >. Acesso em 22 Nov. 2008.

MESSIAS, Antônio R. **[Home Page].** 2008. Disponível em: <http://www.rogercom.com.br> . Acesso em 20 Ago. 2008.

NetBeans. **[Home Page].** 2008. Disponível em: <http://www.netbeans.org/kb/60/java/quickstart-gui_pt_BR.html>. Acesso em 18 Ago. 2008.

ROCHA, Glauber Moreira. Monografia: Uso inteligente da automatização residencial. Um estudo de caso sobre o controle de iluminação. 2º semestre de 2006.

Sky Comp Technology. **[HomePage].** 2008. Disponível em: <http://www.skycomp .com.au/>. Acesso em 29 Out. 2008.

APÊNDICE A – CÓDIGO FONTE – CLASSE TRATAMENTODADOS.JAVA

```
/*
 * PROJETO FINAL
 * CONTROLE DE ILUMINAÇÃO VIA COMANDO DE VOZ
 * Desenvolvido por Alex Perez dos Santos
 * Criação - 15/07/2008
 * Versão Final - 22/11/2008
 * Classe que irá fazer o tratamento dos comandos ditados
 */

package Projeto;

import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import javax.swing.JOptionPane;
import javax.swing.UIManager;
import NTPortJava.NTPortJava;



public class TratamentoDados extends Thread {

	// Cria a tela do sistema
	public Interface tela = new Interface();


	public TratamentoDados(){


	// Configura o layout da tela do sistema
	try{

	UIManager.setLookAndFeel("com.sun.java.swing.plaf.windows.WindowsLoo
	kAndFeel");

	}catch(Exception ex){
		JOptionPane.showMessageDialog(null, "Erro ao abrir o sistema!",
		"Erro", 0);
	System.exit(0);
	 }
			Interface layout = new Interface();
			tela=layout;
			tela.setVisible(true);

			// Verifica o estado inicial das lâmpadas
```

```
            verificaEstadoInicial();

            // Cria as funcionalidades dos botões
             ativaFuncionalidadesBotoes();

}// Fim do construtor


// Cria as variáveis de controle
int comando=0;

int valor=0;

int contCmd=0;

int contCaptura=0;

String fraseDitada="";

String msgInicial = "# Banco de dados conectado.\n" +
                                        "# Porta LPT1 aberta para
                        comunicação ---> Endereço 0xFFE8. \n" +
                                        "# INFORME UM COMANDO...";

String msgEfetivado = "O COMANDO FOI EFETIVADO COM SUCESSO!";

String erroLpt1 = "Erro na comunicação com a porta paralela LPT1!\n" +
                  "Reinicie o programa...";

// Configura as váriaveis de estado das lâmpadas
boolean l1=false,l2=false,l3=false,l4=false;


// Classe para comunicação com a porta paralela
PortaParalela lpt1 = new PortaParalela();

// Cria Thread para capturar as palavras ditadas
public Thread capturaComando = new Thread(this);


public void leituraDados() {


        //Estabelece conexão com o banco de dados
        BancoDados banco = new BancoDados();

        //Habilita a porta paralela LPT1 para comunicação
        lpt1.abrePorta();

        //Inicia a leitura dos comandos
```

```
while (true) {
        try {

        // Aguarda 0,5 segundos para armazenar o comando ditado
        Thread.sleep(500);
        fraseDitada = tela.jTextField1.getText();

        //Envia a frase ditada para comparação com os comandos
        válidos
        comando=banco.consultaBanco(fraseDitada);


        //Verifica a variável comando
        switch(comando){

                //Acende a lâmpada da sala
                case 16:
                        banco.status=0;
                        if(l1==false){
                                valor=valor+1;
                                lpt1.setPort(valor);
                                tela.jTextField1.setText("");
                                if(lpt1.getPort()==valor){
                                tela.jLabel13.setIcon(new
                                javax.swing.ImageIcon
                                ("D:\\Desenvolvimento\\Imple
                                mentação\\Projeto\\
                                lampada_acesa.jpg"));


                        tela.jTextArea2.setText(msgEfetivado);
                        l1=true;
                                }
                                else{

                                tela.jTextArea2.setText(erroLpt1);
                                }
                        }
                        else{
                                tela.jTextArea2.setText("A lâmpada
                                da sala já está ligada!");
                                tela.jTextField1.setText("");
                        }
                        break;

                //Acende a lâmpada da suíte
                case 25:
                        banco.status=0;
                        if(l2==false){
                                valor=valor+2;
```

```
                    lpt1.setPort(valor);
                    tela.jTextField1.setText("");
                    if(lpt1.getPort()==valor){
                        tela.jLabel26.setIcon(new
                         javax.swing.ImageIcon
                        ("D:\\Desenvolvimento\\Imple
                        mentação\\Projeto\\
                        lampada_acesa.jpg"));


                tela.jTextArea2.setText(msgEfetivado);
                l2=true;
                    }
                    else{

                    tela.jTextArea2.setText(erroLpt1);
                    }
                }
                else{
                    tela.jTextArea2.setText("A lâmpada
                        da suíte já está ligada!");

                    tela.jTextField1.setText("");
                }
                break;

            //Acende a lâmpada da cozinha
            case 36:
                banco.status=0;
                if(l3==false){
                    valor=valor+4;
                    lpt1.setPort(valor);
                    tela.jTextField1.setText("");
                    if(lpt1.getPort()==valor){
                        tela.jLabel27.setIcon(new
                        javax.swing.ImageIcon
                        ("D:\\Desenvolvimento\\Imple
                        mentação\\Projeto\\
                        lampada_acesa.jpg"));

                tela.jTextArea2.setText(msgEfetivado);
                l3=true;
                    }
                    else{

                    tela.jTextArea2.setText(erroLpt1);
                    }
                }
                else{
```

```
                    tela.jTextArea2.setText("A lâmpada
                        da cozinha já está ligada!");
                    tela.jTextField1.setText("");
                }
                break;

            //Acende a lâmpada do banheiro
            case 64:
                banco.status=0;
                if(l4==false){
                valor=valor+8;
                lpt1.setPort(valor);
                tela.jTextField1.setText("");
                    if(lpt1.getPort()==valor){
                        tela.jLabel28.setIcon(new
                        javax.swing.ImageIcon
                        ("D:\\Desenvolvimento\\Imple
                        mentação\\Projeto\\
                        lampada_acesa.jpg"));

            tela.jTextArea2.setText(msgEfetivado);
            l4=true;
                    }
                    else{

                    tela.jTextArea2.setText(erroLpt1);
                    }
                }
                else{
                    tela.jTextArea2.setText("A lâmpada
                        do banheiro já está ligada!");
                    tela.jTextField1.setText("");
                }
                break;


            //Acende todas as lâmpadas
            case 49:
                banco.status=0;
                tela.jTextField1.setText("");
                if(l1==true && l2==true && l3==true &&
                l4==true){
                tela.jTextArea2.setText("Todas as lâmpadas
                estão ligadas!");
                tela.jTextField1.setText("");
                }
                else{
                    valor=15;
                    lpt1.setPort(15);
```

```
                    if(lpt1.getPort()==valor){
                        tela.jLabel13.setIcon(new
                        javax.swing.ImageIcon
                        ("D:\\Desenvolvimento\\Imple
                        mentação\\Projeto\\
                        lampada_acesa.jpg"));
                        tela.jLabel26.setIcon(new

                        javax.swing.ImageIcon
                        ("D:\\Desenvolvimento\\Imple
                        mentação\\Projeto\\
                        lampada_acesa.jpg"));
                        tela.jLabel27.setIcon(new
                        javax.swing.ImageIcon
                        ("D:\\Desenvolvimento\\Imple
                        mentação\\Projeto\\
                        lampada_acesa.jpg"));
                        tela.jLabel28.setIcon(new
                        javax.swing.ImageIcon
                        ("D:\\Desenvolvimento\\Imple
                        mentação\\Projeto\\
                        lampada_acesa.jpg"));

                tela.jTextArea2.setText(msgEfetivado);
                l1=true;
                l2=true;
                l3=true;
                l4=true;
                    }
                    else{

                tela.jTextArea2.setText(erroLpt1);
                    }
                }
                break;

            //Apaga a lâmpada da sala
            case 32:
                banco.status=0;
                if(l1==true){
                    valor=valor-1;
                    lpt1.setPort(valor);
                    tela.jTextField1.setText("");
                    if(lpt1.getPort()==valor){
                        tela.jLabel13.setIcon(new
                        javax.swing.ImageIcon
                        ("D:\\Desenvolvimento\\Imple
                        mentação\\Projeto\\
                        lampada_apagada.jpg"));
```

```
                tela.jTextArea2.setText(msgEfetivado);
                l1=false;
                        }
                        else{

                tela.jTextArea2.setText(erroLpt1);
                        }
                }
                else{

                        tela.jTextArea2.setText("A lâmpada
                                da sala já está desligada!");
                        tela.jTextField1.setText("");

                }
                break;

        //Apaga a lâmpada da suíte
        case 50:
                banco.status=0;
                if(l2==true){
                valor=valor-2;
                lpt1.setPort(valor);
                tela.jTextField1.setText("");
                if(lpt1.getPort()==valor){
                        tela.jLabel26.setIcon(new
                        javax.swing.ImageIcon
                        ("D:\\Desenvolvimento\\Imple
                        mentação\\Projeto
                        \\lampada_apagada.jpg"));

                tela.jTextArea2.setText(msgEfetivado);
                l2=false;
                }
                else{
                        tela.jTextArea2.setText(erroLpt1);
                }
                }
                else{
                        tela.jTextArea2.setText("A lâmpada
                                da suíte já está desligada!");
                        tela.jTextField1.setText("");
                }
                break;

        //Apaga a lâmpada da cozinha
        case 72:
                banco.status=0;
                if(l3==true){
```

```
                valor=valor-4;
                lpt1.setPort(valor);
                tela.jTextField1.setText("");
                        if(lpt1.getPort()==valor){
                                tela.jLabel27.setIcon(new
                                javax.swing.ImageIcon
                                ("D:\\Desenvolvimento\\Imple
                                mentação\\Projeto\\
                                lampada_apagada.jpg"));

                tela.jTextArea2.setText(msgEfetivado);
                l3=false;
                        }
                        else{

                tela.jTextArea2.setText(erroLpt1);
                        }
                }
                else{

                        tela.jTextArea2.setText("A lâmpada
                        da cozinha já está desligada!");
                        tela.jTextField1.setText("");
                }

                break;

        //Apaga a lâmpada do banheiro
        case 128:
                banco.status=0;
                if(l4==true){
                        valor=valor-8;
                        lpt1.setPort(valor);
                        tela.jTextField1.setText("");
                        if(lpt1.getPort()==valor){
                                tela.jLabel28.setIcon(new
                                javax.swing.ImageIcon
                                ("D:\\Desenvolvimento\\Imple
                                mentação\\Projeto\\
                                lampada_apagada.jpg"));

                tela.jTextArea2.setText(msgEfetivado);
                l4=false;
                        }
                        else{
                        tela.jTextArea2.setText(erroLpt1);
                        }
                }
                else{
```

```
                    tela.jTextArea2.setText("A lâmpada
                    do banheiro já está desligada!");
                    tela.jTextField1.setText("");
                }
                break;

            //Apaga todas as lâmpadas
            case 98:
                banco.status=0;
                tela.jTextField1.setText("");
                if(l1==false && l2==false && l3==false &&
                l4==false){
                    if(lpt1.getPort()==0){

                    tela.jTextArea2.setText("Todas as
                    lâmpadas estão apagadas!");
                    tela.jTextField1.setText("");
                    }
                    else{

                    tela.jTextArea2.setText(erroLpt1);
                    }
                }
                else{
                    valor=0;
                    lpt1.setPort(valor);
                    if(lpt1.getPort()==valor){

                        tela.jLabel13.setIcon(new
                        javax.swing.ImageIcon
                        ("D:\\Desenvolvimento\\Imple
                        mentação\\Projeto\\
                        lampada_apagada.jpg"));
                        tela.jLabel26.setIcon(new
                        javax.swing.ImageIcon
                        ("D:\\Desenvolvimento\\Imple
                        mentação\\Projeto\\
                        lampada_apagada.jpg"));
                        tela.jLabel27.setIcon(new
                        javax.swing.ImageIcon
                        ("D:\\Desenvolvimento\\Imple
                        mentação\\Projeto\\
                        lampada_apagada.jpg"));
                        tela.jLabel28.setIcon(new
                        javax.swing.ImageIcon
                        ("D:\\Desenvolvimento\\Imple
                        mentação\\Projeto\\
                        lampada_apagada.jpg"));

                tela.jTextArea2.setText(msgEfetivado);
```

```
                                        l1=false;
                                        l2=false;
                                        l3=false;
                                        l4=false;
                                                }

                                                else{

                                                tela.jTextArea2.setText(erroLpt1);
                                                }

                                        }
                                        break;

                        // Se nenhum comando for válido retorna a mensagem
                                default:
                                        if(!fraseDitada.equals("")){
                                        tela.jTextArea2.setText("Tente outro
                                                        comando por favor.");
                                        }
                                        break;
                                }

                        } catch (Exception ex) {
                                tela.jTextArea2.setText("Não foi possível processar
                                                                o comando.");
                        }

                }

}


//Configura os eventos dos botões e menus


public void acionaBotaoLimparRegistros() {
        tela.jButton1.addActionListener(new ActionListener() {
                public void actionPerformed(ActionEvent e) {
                        tela.jTextArea2.setText("");
                        tela.jTextField1.grabFocus();
                }
        });
}

public void acionaItemIniciar(){
        tela.jMenuItem1.addActionListener(new ActionListener(){
                public void actionPerformed(ActionEvent e){
                        lpt1.abrePorta();
                        tela.jTextArea2.setText(msgInicial);
```

```
                                tela.jTextField1.grabFocus();
                                tela.jButton2.setText("Parar");
                                if(contCaptura!=0){
                                        capturaComando.resume();
                                }else{
                                        capturaComando.start();
                                        contCaptura++;
                                }
                                tela.estadobotao=false;
                        }
                });
        }

        public void acionaItemSair(){
                tela.jMenuItem2.addActionListener(new ActionListener(){
                        public void actionPerformed(ActionEvent e){
                                lpt1.fechaPorta();
                                System.exit(0);
                        }
                });
        }

        public void acionaItemConteudo(){
                tela.jMenuItem3.addActionListener(new ActionListener(){
                        public void actionPerformed(ActionEvent e){
                                JOptionPane.showMessageDialog(tela, " SISTEMA
DE AUTOMAÇÃO RESIDENCIAL\n\n " +
                                                "Para a utilização do sistema siga
os passos abaixo:\n\n " +
                                                "Passo 1 - Aperte o botão INICIAR
para ativar a captura dos comandos.\n\n" +
                                                " Passo 2 - Fale no microfone
comandos válidos para controlar a iluminação do ambiente desejado.\n", "Ajuda", 3);

                        }
                });
        }

        public void acionaBotaoIniciar() {
                tela.jButton2.addActionListener(new ActionListener() {

                        public void actionPerformed(ActionEvent e) {

                                if(tela.estadobotao==true){
                                        lpt1.abrePorta();
                                        tela.jTextArea2.setText(msgInicial);
                                        tela.jTextField1.grabFocus();
                                        tela.jButton2.setText("Parar");
                                        if(contCaptura!=0){
                                                capturaComando.resume();
```

```
                                        }else{
                                                capturaComando.start();
                                                contCaptura++;
                                        }
                                        tela.estadobotao=false;


                                } else  if(tela.estadobotao==false){

                                        tela.jTextArea2.setText("Aperte o botão
                                        INICIAR para ativar a captura dos
                                        comandos.");

                                        tela.jButton2.setText("Iniciar");
                                        tela.jTextField1.grabFocus();
                                        tela.jTextField1.setText("");
                                        tela.estadobotao=true;
                                        lpt1.fechaPorta();
                                        capturaComando.suspend();
                                        tela.estadobotao=true;
                                }

                        }
                });
        }


        public void ativaFuncionalidadesBotoes(){
                acionaBotaoIniciar();
                acionaBotaoLimparRegistros();
                acionaItemSair();
                acionaItemConteudo();
                acionaItemIniciar();

        }

        public void run(){
                leituraDados();
        }

        // Verifica o estado da porta LPT1
        public void verificaEstadoInicial(){

                lpt1.abrePorta();
                tela.jTextArea2.setText("Aperte o botão INICIAR para ativar a
                                                        captura dos comandos...");
                tela.jTextArea2.setEditable(false);

                if(lpt1.getPort()==1){
                        tela.jLabel13.setIcon(new javax.swing.ImageIcon
```

```
            ("D:\\Desenvolvimento\\Imple
            mentação\\Projeto\\
            lampada_acesa.jpg"));
            l1=true;
            valor=1;
}
if(lpt1.getPort()==2){
            tela.jLabel26.setIcon(new javax.swing.ImageIcon
            ("D:\\Desenvolvimento\\Imple
            mentação\\Projeto\\
            lampada_acesa.jpg"));
            l2=true;
            valor=2;
}
if(lpt1.getPort()==4){
            tela.jLabel27.setIcon(new javax.swing.ImageIcon
            ("D:\\Desenvolvimento\\Imple
            mentação\\Projeto\\
            lampada_acesa.jpg"));
            l3=true;
            valor=4;
}
if(lpt1.getPort()==8){
            tela.jLabel28.setIcon(new javax.swing.ImageIcon
            ("D:\\Desenvolvimento\\Imple
            mentação\\Projeto\\
            lampada_acesa.jpg"));
            l4=true;
            valor=8;
}

if(lpt1.getPort()==3){
            tela.jLabel13.setIcon(new javax.swing.ImageIcon
            ("D:\\Desenvolvimento\\Imple
            mentação\\Projeto\\
            lampada_acesa.jpg"));
            l1=true;
            tela.jLabel26.setIcon(new javax.swing.ImageIcon
            ("D:\\Desenvolvimento\\Imple
            mentação\\Projeto\\
            lampada_acesa.jpg"));
            l2=true;
            valor=3;
}

if(lpt1.getPort()==5){
            tela.jLabel13.setIcon(new javax.swing.ImageIcon
            ("D:\\Desenvolvimento\\Imple
            mentação\\Projeto\\
            lampada_acesa.jpg"));
```

```
                l1=true;
                tela.jLabel27.setIcon(new javax.swing.ImageIcon
                ("D:\\Desenvolvimento\\Imple
                mentação\\Projeto\\
                lampada_acesa.jpg"));
                l3=true;
                valor=5;
        }
        if(lpt1.getPort()==7){
                tela.jLabel13.setIcon(new javax.swing.ImageIcon
                ("D:\\Desenvolvimento\\Imple
                mentação\\Projeto\\
                lampada_acesa.jpg"));
                l1=true;
                tela.jLabel26.setIcon(new javax.swing.ImageIcon
                ("D:\\Desenvolvimento\\Imple
                mentação\\Projeto\\
                lampada_acesa.jpg"));
                l2=true;
                tela.jLabel27.setIcon(new javax.swing.ImageIcon
                ("D:\\Desenvolvimento\\Imple
                mentação\\Projeto\\
                lampada_acesa.jpg"));
                l3=true;
                valor=7;
        }
        if(lpt1.getPort()==11){
                tela.jLabel13.setIcon(new javax.swing.ImageIcon
                ("D:\\Desenvolvimento\\Imple
                mentação\\Projeto\\
                lampada_acesa.jpg"));
                l1=true;
                tela.jLabel26.setIcon(new javax.swing.ImageIcon
                ("D:\\Desenvolvimento\\Imple
                mentação\\Projeto\\
                lampada_acesa.jpg"));
                l2=true;
                tela.jLabel28.setIcon(new javax.swing.ImageIcon
                ("D:\\Desenvolvimento\\Imple
                mentação\\Projeto\\
                lampada_acesa.jpg"));
                l4=true;
                valor=11;
        }
        if(lpt1.getPort()==13){
                tela.jLabel13.setIcon(new javax.swing.ImageIcon
                ("D:\\Desenvolvimento\\Imple
                mentação\\Projeto\\
                lampada_acesa.jpg"));
                l1=true;
```

```
            tela.jLabel27.setIcon(new javax.swing.ImageIcon
            ("D:\\Desenvolvimento\\Imple
            mentação\\Projeto\\
            lampada_acesa.jpg"));
            l3=true;
            tela.jLabel28.setIcon(new javax.swing.ImageIcon
            ("D:\\Desenvolvimento\\Imple
            mentação\\Projeto\\
            lampada_acesa.jpg"));
            l4=true;
            valor=13;
}
if(lpt1.getPort()==14){
            tela.jLabel26.setIcon(new javax.swing.ImageIcon
            ("D:\\Desenvolvimento\\Imple
            mentação\\Projeto\\
            lampada_acesa.jpg"));
            l2=true;
            tela.jLabel27.setIcon(new javax.swing.ImageIcon
            ("D:\\Desenvolvimento\\Imple
            mentação\\Projeto\\
            lampada_acesa.jpg"));
            l3=true;
            tela.jLabel28.setIcon(new javax.swing.ImageIcon
            ("D:\\Desenvolvimento\\Imple
            mentação\\Projeto\\
            lampada_acesa.jpg"));
            l4=true;
            valor=14;
}

if(lpt1.getPort()==9){
            tela.jLabel13.setIcon(new javax.swing.ImageIcon
            ("D:\\Desenvolvimento\\Imple
            mentação\\Projeto\\
            lampada_acesa.jpg"));
            l1=true;
            tela.jLabel28.setIcon(new javax.swing.ImageIcon
            ("D:\\Desenvolvimento\\Imple
            mentação\\Projeto\\
            lampada_acesa.jpg"));
            l4=true;
            valor=9;
}
if(lpt1.getPort()==6){
            tela.jLabel26.setIcon(new javax.swing.ImageIcon
            ("D:\\Desenvolvimento\\Imple
            mentação\\Projeto\\
            lampada_acesa.jpg"));
            l2=true;
```

```
                tela.jLabel27.setIcon(new javax.swing.ImageIcon
                ("D:\\Desenvolvimento\\Imple
                mentação\\Projeto\\
                lampada_acesa.jpg"));
                l3=true;
                valor=6;
        }
        if(lpt1.getPort()==10){
                tela.jLabel26.setIcon(new javax.swing.ImageIcon
                ("D:\\Desenvolvimento\\Imple
                mentação\\Projeto\\
                lampada_acesa.jpg"));
                l2=true;
                tela.jLabel28.setIcon(new javax.swing.ImageIcon
                ("D:\\Desenvolvimento\\Imple
                mentação\\Projeto\\
                lampada_acesa.jpg"));
                l4=true;
                valor=10;
        }
        if(lpt1.getPort()==12){
                tela.jLabel27.setIcon(new javax.swing.ImageIcon
                ("D:\\Desenvolvimento\\Imple
                mentação\\Projeto\\
                lampada_acesa.jpg"));
                l3=true;
                tela.jLabel28.setIcon(new javax.swing.ImageIcon
                ("D:\\Desenvolvimento\\Imple
                mentação\\Projeto\\
                lampada_acesa.jpg"));
                l4=true;
                valor=12;
        }
        if(lpt1.getPort()==15){
                tela.jLabel13.setIcon(new javax.swing.ImageIcon
                ("D:\\Desenvolvimento\\Imple
                mentação\\Projeto\\
                lampada_acesa.jpg"));
                tela.jLabel26.setIcon(new javax.swing.ImageIcon
                ("D:\\Desenvolvimento\\Imple
                mentação\\Projeto\\
                lampada_acesa.jpg"));
                tela.jLabel27.setIcon(new javax.swing.ImageIcon
                ("D:\\Desenvolvimento\\Imple
                mentação\\Projeto\\
                lampada_acesa.jpg"));
                tela.jLabel28.setIcon(new javax.swing.ImageIcon
                ("D:\\Desenvolvimento\\Imple
                mentação\\Projeto\\
                lampada_acesa.jpg"));
```

```
                        l1=true;
                        l2=true;
                        l3=true;
                        l4=true;
                        valor=15;
                }

        }



        public static void main(String args[]) {
                TratamentoDados sistema = new TratamentoDados();
        }

}//Fim da classe
```

APÊNDICE B – CÓDIGO FONTE – CLASSE BANCODADOS.JAVA

```
/*
 * PROJETO FINAL
 * CONTROLE DE ILUMINAÇÃO VIA COMANDO DE VOZ
 * Desenvolvido por Alex Perez dos Santos
 * Criação - 15/07/2008
 * Versão Final - 22/11/2008
 * Classe para a comunicação com o banco de dados
 */

package Projeto;

import java.sql.*;
import javax.swing.JOptionPane;


public class BancoDados {

	// Estabelece a conexão com o banco de dados
	Connection conexao = null;
	Statement st = null;
	int status=0;

	public int consultaBanco(String cmd) {

		try {
			String nomeDriver = "com.mysql.jdbc.Driver";
			Class.forName(nomeDriver);
			String servidor = "localhost";
			String bancodedados = "db4";
			String url = "jdbc:mysql://" + servidor + ":3306/" +
			bancodedados;
			conexao = DriverManager.getConnection(url, "root",
			"database");
			st = conexao.createStatement();

			// Executa a view criada no mysql
			ResultSet consulta = st.executeQuery("SELECT * FROM
			TODOSOSCOMANDOS");


			while (consulta.next()) {

				// As variáveis armazenam as linhas das tabelas
				do banco
				String comando = consulta.getString("COMANDO");
```

```
                                String ambiente =
                                consulta.getString("AMBIENTE");
                                int idComando =
                                consulta.getInt("TIPOCOMANDO_IDTIPO");
                                int idAmbiente =
                                consulta.getInt("TIPOAMBIENTE_
                                IDTIPOAMBIENTE");


                                // Se a frase falada tiver palavras válidas
                                // o método retorna um valor para a
                                // identificação
                                // de qual comando deverá ser efetivado

                    if(cmd.contains(comando) && cmd.contains(ambiente)){

                            status = idAmbiente*idAmbiente*idComando;

                                    }
                            }

                            st.close();
                            conexao.close();

                    } catch (ClassNotFoundException e) {
                            JOptionPane.showMessageDialog(null, "O driver
                                        especificado não foi encontrado.",
                                                "ClassNotFoundException", 0);

                    } catch (SQLException e) {
                            JOptionPane.showMessageDialog(null, "Não foi possível
                                    estabelecer conexão com o banco de dados.",
                                                    "SQLException", 0);
                    }
                    return status;

            }// Fim do método consulta banco

}// Fim da classe
```

APÊNDICE C – CÓDIGO FONTE – CLASSE INTERFACE.JAVA

```
/*
 * PROJETO FINAL
 * CONTROLE DE ILUMINAÇÃO VIA COMANDO DE VOZ
 * Desenvolvido por Alex Perez dos Santos
 * Criação - 15/07/2008
 * Versão Final - 22/11/2008
 * Classe para criação da interface gráfica do sistema
 */

package Projeto;

import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.Color;
import javax.swing.*;



public class Interface extends javax.swing.JFrame {


    public Interface() {
        inicializaComponentes();

    }


    private void inicializaComponentes() {
        jLabel7 = new javax.swing.JLabel();
        jLabel25 = new javax.swing.JLabel();
        jPanel1 = new javax.swing.JPanel();
        jTextField1 = new javax.swing.JTextField();
        jButton2 = new javax.swing.JButton();
        jLabel2 = new javax.swing.JLabel();
        jPanel2 = new javax.swing.JPanel();
        jLabel4 = new javax.swing.JLabel();
        jLabel12 = new javax.swing.JLabel();
        jLabel14 = new javax.swing.JLabel();
        jSeparator1 = new javax.swing.JSeparator();
        jSeparator2 = new javax.swing.JSeparator();
        jSeparator3 = new javax.swing.JSeparator();
        jLabel15 = new javax.swing.JLabel();
        jLabel16 = new javax.swing.JLabel();
        jLabel17 = new javax.swing.JLabel();
```

```
        jLabel18 = new javax.swing.JLabel();
        jLabel19 = new javax.swing.JLabel();
        jLabel20 = new javax.swing.JLabel();
        jLabel21 = new javax.swing.JLabel();
        jLabel22 = new javax.swing.JLabel();
        jLabel23 = new javax.swing.JLabel();
        jLabel24 = new javax.swing.JLabel();
        jSeparator4 = new javax.swing.JSeparator();
        jPanel7 = new javax.swing.JPanel();
        jScrollPane2 = new javax.swing.JScrollPane();
        jTextArea2 = new javax.swing.JTextArea();
        jPanel9 = new javax.swing.JPanel();
        jLabel9 = new javax.swing.JLabel();
        jLabel10 = new javax.swing.JLabel();
        jLabel11 = new javax.swing.JLabel();
        jLabel6 = new javax.swing.JLabel();
        jLabel13 = new javax.swing.JLabel();
        jLabel26 = new javax.swing.JLabel();
        jLabel27 = new javax.swing.JLabel();
        jLabel28 = new javax.swing.JLabel();
        jButton1 = new javax.swing.JButton();
        jLabel1 = new javax.swing.JLabel();
        jMenuBar1 = new javax.swing.JMenuBar();
        jMenu1 = new javax.swing.JMenu();
        jMenuItem1 = new javax.swing.JMenuItem();
        jMenuItem2 = new javax.swing.JMenuItem();
        jMenu2 = new javax.swing.JMenu();
        jMenuItem3 = new javax.swing.JMenuItem();
        jLabel30 = new javax.swing.JLabel();


        jTextArea2.setSelectionColor(Color.white);
        jLabel7.setIcon(new
javax.swing.ImageIcon("D:\\Desenvolvimento\\Implementação\\Projeto\\logo_ceub.jp
g"));
        jLabel25.setIcon(new
javax.swing.ImageIcon("D:\\Desenvolvimento\\Implementação\\Projeto\\lampada_ap
agada.jpg"));
        jLabel25.setIcon(new
javax.swing.ImageIcon("D:\\Desenvolvimento\\Implementação\\Projeto\\lampada_ac
esa.jpg"));

        setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
        setTitle("SAR - Desenvolvimento");
        setIconImage(getIconImage());
        setResizable(false);
        jPanel1.setBorder(javax.swing.BorderFactory.createTitledBorder("Comando"));

        jButton2.setText("Iniciar");
```

```
        javax.swing.GroupLayout jPanel1Layout = new
javax.swing.GroupLayout(jPanel1);
        jPanel1.setLayout(jPanel1Layout);
        jPanel1Layout.setHorizontalGroup(

jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(jPanel1Layout.createSequentialGroup()
                .addContainerGap()
                .addComponent(jTextField1,
javax.swing.GroupLayout.PREFERRED_SIZE, 436,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 30,
Short.MAX_VALUE)
                .addComponent(jButton2, javax.swing.GroupLayout.PREFERRED_SIZE,
91, javax.swing.GroupLayout.PREFERRED_SIZE)
                .addGap(25, 25, 25))
        );
        jPanel1Layout.setVerticalGroup(

jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
jPanel1Layout.createSequentialGroup()
                .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.
BASELINE)
                    .addComponent(jTextField1,
javax.swing.GroupLayout.PREFERRED_SIZE, 35,
javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addComponent(jButton2))
                .addGap(81, 81, 81))
        );

        jLabel2.setFont(new java.awt.Font("Calibri", 0, 24));
        jLabel2.setForeground(new java.awt.Color(204, 0, 0));
        jLabel2.setText("Sistema de Automação Residencial");


jPanel2.setBorder(javax.swing.BorderFactory.createTitledBorder(javax.swing.Border
Factory.createTitledBorder("Menu")));
        jLabel4.setBackground(new java.awt.Color(51, 51, 51));
        jLabel4.setForeground(new java.awt.Color(102, 102, 102));
        jLabel4.setText("COMANDOS VÁLIDOS");

        jLabel12.setBackground(new java.awt.Color(102, 102, 102));
        jLabel12.setForeground(new java.awt.Color(102, 102, 102));
        jLabel12.setText("    Função");
```

```
        jLabel14.setBackground(new java.awt.Color(102, 102, 102));
        jLabel14.setForeground(new java.awt.Color(102, 102, 102));
        jLabel14.setText("Ambiente");

        jSeparator3.setOrientation(javax.swing.SwingConstants.VERTICAL);

        jLabel15.setText("ACENDE");

        jLabel16.setText("APAGA");

        jLabel17.setText("LIGA");

        jLabel18.setText("ATIVA");

        jLabel19.setText("DESLIGA");

        jLabel20.setText("DESATIVA");

        jLabel21.setText("SALA");

        jLabel22.setText("SUÍTE");

        jLabel23.setText("COZINHA");

        jLabel24.setText("BANHEIRO");

        jLabel30.setText("TODOS");


        javax.swing.GroupLayout jPanel2Layout = new
javax.swing.GroupLayout(jPanel2);
        jPanel2.setLayout(jPanel2Layout);
        jPanel2Layout.setHorizontalGroup(

jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(jPanel2Layout.createSequentialGroup()
               .addContainerGap()

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.L
EADING)
                  .addComponent(jSeparator1,
javax.swing.GroupLayout.DEFAULT_SIZE, 237, Short.MAX_VALUE)
                  .addComponent(jSeparator2,
javax.swing.GroupLayout.DEFAULT_SIZE, 237, Short.MAX_VALUE)
                  .addGroup(jPanel2Layout.createSequentialGroup()

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.L
EADING)
                        .addGroup(jPanel2Layout.createSequentialGroup()
```

```
.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.L
EADING)
                        .addComponent(jSeparator4,
javax.swing.GroupLayout.PREFERRED_SIZE, 107,
javax.swing.GroupLayout.PREFERRED_SIZE)
                        .addGroup(jPanel2Layout.createSequentialGroup()
                            .addGap(36, 36, 36)

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.L
EADING)
                                .addComponent(jLabel15)
                                .addComponent(jLabel17)
                                .addComponent(jLabel18))))
                    .addGap(15, 15, 15))
                .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
jPanel2Layout.createSequentialGroup()

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.L
EADING)
                        .addComponent(jLabel16)
                        .addComponent(jLabel19)
                        .addComponent(jLabel20))
                    .addGap(35, 35, 35)))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                .addComponent(jSeparator3,
javax.swing.GroupLayout.PREFERRED_SIZE, 15,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addGap(21, 21, 21)

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.L
EADING)
                    .addComponent(jLabel30)
                    .addComponent(jLabel24)
                    .addComponent(jLabel23)
                    .addComponent(jLabel22)
                    .addComponent(jLabel21)))
            .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
jPanel2Layout.createSequentialGroup()
                .addGap(27, 27, 27)
                .addComponent(jLabel12,
javax.swing.GroupLayout.PREFERRED_SIZE, 62,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 72,
Short.MAX_VALUE)
                .addComponent(jLabel14)
                .addGap(31, 31, 31))
```

```
                .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
jPanel2Layout.createSequentialGroup()

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 67,
javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addComponent(jLabel4,
javax.swing.GroupLayout.PREFERRED_SIZE, 118,
javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addGap(52, 52, 52)))
            .addContainerGap())
        );
        jPanel2Layout.setVerticalGroup(

jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(jPanel2Layout.createSequentialGroup()
                .addContainerGap()
                .addComponent(jLabel4)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                .addComponent(jSeparator1,
javax.swing.GroupLayout.PREFERRED_SIZE, 10,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.
BASELINE)
                    .addComponent(jLabel14)
                    .addComponent(jLabel12))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                .addComponent(jSeparator2,
javax.swing.GroupLayout.PREFERRED_SIZE, 10,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.L
EADING)

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.L
EADING)
                    .addGroup(jPanel2Layout.createSequentialGroup()
                        .addComponent(jLabel15)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                        .addComponent(jLabel17)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                        .addComponent(jLabel18)
                        .addGap(14, 14, 14)
```

```
                        .addComponent(jSeparator4,
javax.swing.GroupLayout.PREFERRED_SIZE, 10,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                        .addComponent(jLabel16)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                        .addComponent(jLabel19)
                        .addGap(6, 6, 6)
                        .addComponent(jLabel20))
                    .addComponent(jSeparator3,
javax.swing.GroupLayout.DEFAULT_SIZE, 140, Short.MAX_VALUE))
                .addGroup(jPanel2Layout.createSequentialGroup()
                    .addGap(31, 31, 31)
                    .addComponent(jLabel21)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                    .addComponent(jLabel22)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                    .addComponent(jLabel23)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                    .addComponent(jLabel24)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                    .addComponent(jLabel30)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 35,
javax.swing.GroupLayout.PREFERRED_SIZE)))
                .addGap(14, 14, 14))
        );

        jPanel7.setBorder(javax.swing.BorderFactory.createTitledBorder("Registros"));
        jTextArea2.setColumns(20);
        jTextArea2.setRows(5);
        jScrollPane2.setViewportView(jTextArea2);

        javax.swing.GroupLayout jPanel7Layout = new
javax.swing.GroupLayout(jPanel7);
        jPanel7.setLayout(jPanel7Layout);
        jPanel7Layout.setHorizontalGroup(

jPanel7Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(jPanel7Layout.createSequentialGroup()
                .addContainerGap()
                .addComponent(jScrollPane2,
javax.swing.GroupLayout.PREFERRED_SIZE, 564,
javax.swing.GroupLayout.PREFERRED_SIZE)
```

```
                .addContainerGap(18, Short.MAX_VALUE))
        );
        jPanel7Layout.setVerticalGroup(

jPanel7Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(jPanel7Layout.createSequentialGroup()
                .addContainerGap()
                .addComponent(jScrollPane2,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addContainerGap(13, Short.MAX_VALUE))
        );

        jPanel9.setBorder(javax.swing.BorderFactory.createTitledBorder("Status"));
        jLabel9.setText("SALA");

        jLabel10.setText(" SUÍTE");

        jLabel11.setText(" COZINHA");

        jLabel6.setText("  BANHEIRO");
        jLabel13.setIcon(new
javax.swing.ImageIcon("D:\\Desenvolvimento\\Implementação\\Projeto\\lampada_ap
agada.jpg"));

        jLabel26.setIcon(new
javax.swing.ImageIcon("D:\\Desenvolvimento\\Implementação\\Projeto\\lampada_ap
agada.jpg"));

        jLabel27.setIcon(new
javax.swing.ImageIcon("D:\\Desenvolvimento\\Implementação\\Projeto\\lampada_ap
agada.jpg"));

        jLabel28.setIcon(new
javax.swing.ImageIcon("D:\\Desenvolvimento\\Implementação\\Projeto\\lampada_ap
agada.jpg"));

        javax.swing.GroupLayout jPanel9Layout = new
javax.swing.GroupLayout(jPanel9);
        jPanel9.setLayout(jPanel9Layout);
        jPanel9Layout.setHorizontalGroup(

jPanel9Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
jPanel9Layout.createSequentialGroup()
                .addGap(53, 53, 53)

.addGroup(jPanel9Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.L
EADING)
```

```
                .addGroup(jPanel9Layout.createSequentialGroup()
                    .addGap(10, 10, 10)
                    .addComponent(jLabel9,
javax.swing.GroupLayout.PREFERRED_SIZE, 30,
javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addGap(93, 93, 93)
                    .addComponent(jLabel10)
                    .addGap(15, 15, 15))
                .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
jPanel9Layout.createSequentialGroup()
                    .addComponent(jLabel27)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 76,
Short.MAX_VALUE)
                    .addComponent(jLabel28)
                    .addGap(4, 4, 4))
                .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
jPanel9Layout.createSequentialGroup()
                    .addComponent(jLabel11,
javax.swing.GroupLayout.PREFERRED_SIZE, 82,
javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addGap(40, 40, 40)
                    .addComponent(jLabel6, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
                .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
jPanel9Layout.createSequentialGroup()
                    .addComponent(jLabel13)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 76,
Short.MAX_VALUE)
                    .addComponent(jLabel26)
                    .addGap(4, 4, 4)))
            .addGap(44, 44, 44))
        );
        jPanel9Layout.setVerticalGroup(

jPanel9Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(jPanel9Layout.createSequentialGroup()
                .addGap(24, 24, 24)

.addGroup(jPanel9Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.
TRAILING)
                .addComponent(jLabel13)
                .addComponent(jLabel26))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(jPanel9Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.
BASELINE)
                .addComponent(jLabel9)
```

```
                .addComponent(jLabel10))
            .addGap(18, 18, 18)

.addGroup(jPanel9Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.
TRAILING)
                .addComponent(jLabel28)
                .addComponent(jLabel27))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)

.addGroup(jPanel9Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.
BASELINE)
                .addComponent(jLabel11)
                .addComponent(jLabel6))
            .addGap(59, 59, 59))
        );

        jButton1.setText("Limpar Registros");

        jLabel1.setIcon(new
javax.swing.ImageIcon("D:\\Desenvolvimento\\Implementação\\Projeto\\logo_ceub.jp
g"));

        jMenu1.setText("Comando");
        jMenuItem1.setText("Iniciar");
        jMenu1.add(jMenuItem1);

        jMenuItem2.setText("Sair");
        jMenu1.add(jMenuItem2);

        jMenuBar1.add(jMenu1);

        jMenu2.setText("Ajuda");
        jMenuItem3.setText("Como utilizar?");
        jMenu2.add(jMenuItem3);

        jMenuBar1.add(jMenu2);

        setJMenuBar(jMenuBar1);

        javax.swing.GroupLayout layout = new
javax.swing.GroupLayout(getContentPane());
        getContentPane().setLayout(layout);
        layout.setHorizontalGroup(
            layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(layout.createSequentialGroup()
                .addGap(36, 36, 36)
                .addComponent(jLabel1)
                .addGap(26, 26, 26)
```

```
            .addComponent(jLabel2, javax.swing.GroupLayout.PREFERRED_SIZE,
424, javax.swing.GroupLayout.PREFERRED_SIZE))
         .addGroup(layout.createSequentialGroup()
            .addContainerGap()
            .addComponent(jPanel2, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addGap(34, 34, 34)
            .addComponent(jPanel9, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
            .addContainerGap())
         .addGroup(layout.createSequentialGroup()
            .addGap(20, 20, 20)
            .addComponent(jButton1, javax.swing.GroupLayout.PREFERRED_SIZE,
141, javax.swing.GroupLayout.PREFERRED_SIZE)
            .addContainerGap(467, Short.MAX_VALUE))
         .addGroup(layout.createSequentialGroup()
            .addContainerGap()
            .addComponent(jPanel7, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
            .addContainerGap())
         .addGroup(layout.createSequentialGroup()
            .addContainerGap()
            .addComponent(jPanel1, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
            .addContainerGap())
      );
      layout.setVerticalGroup(
         layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
         .addGroup(layout.createSequentialGroup()

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADIN
G)
               .addGroup(layout.createSequentialGroup()
                  .addGap(36, 36, 36)
                  .addComponent(jLabel2))
               .addGroup(layout.createSequentialGroup()
                  .addGap(21, 21, 21)
                  .addComponent(jLabel1)))
            .addGap(26, 26, 26)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADIN
G, false)
               .addComponent(jPanel2, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
               .addComponent(jPanel9, javax.swing.GroupLayout.DEFAULT_SIZE,
269, Short.MAX_VALUE))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 25,
Short.MAX_VALUE)
```

```
            .addComponent(jPanel1, javax.swing.GroupLayout.PREFERRED_SIZE,
100, javax.swing.GroupLayout.PREFERRED_SIZE)
            .addGap(28, 28, 28)
            .addComponent(jPanel7, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
            .addComponent(jButton1)
            .addContainerGap())
        );
        pack();
    }



    // Declaração de variáveis
    public javax.swing.JButton jButton1;
    public javax.swing.JButton jButton2;
    private javax.swing.JLabel jLabel1;
    private javax.swing.JLabel jLabel10;
    private javax.swing.JLabel jLabel30;
    private javax.swing.JLabel jLabel11;
    private javax.swing.JLabel jLabel12;
    public javax.swing.JLabel jLabel13;
    private javax.swing.JLabel jLabel14;
    private javax.swing.JLabel jLabel15;
    private javax.swing.JLabel jLabel16;
    public javax.swing.JLabel jLabel17;
    private javax.swing.JLabel jLabel18;
    private javax.swing.JLabel jLabel19;
    private javax.swing.JLabel jLabel2;
    private javax.swing.JLabel jLabel20;
    private javax.swing.JLabel jLabel21;
    private javax.swing.JLabel jLabel22;
    private javax.swing.JLabel jLabel23;
    private javax.swing.JLabel jLabel24;
    public javax.swing.JLabel jLabel25;
    public javax.swing.JLabel jLabel26;
    public javax.swing.JLabel jLabel27;
    public javax.swing.JLabel jLabel28;
    private javax.swing.JLabel jLabel4;
    private javax.swing.JLabel jLabel6;
    private javax.swing.JLabel jLabel7;
    private javax.swing.JLabel jLabel9;
    public javax.swing.JMenu jMenu1;
    public javax.swing.JMenu jMenu2;
    private javax.swing.JMenuBar jMenuBar1;
    public javax.swing.JMenuItem jMenuItem1;
    public javax.swing.JMenuItem jMenuItem2;
```

```
    public javax.swing.JMenuItem jMenuItem3;
    private javax.swing.JPanel jPanel1;
    private javax.swing.JPanel jPanel2;
    private javax.swing.JPanel jPanel7;
    private javax.swing.JPanel jPanel9;
    private javax.swing.JScrollPane jScrollPane2;
    private javax.swing.JSeparator jSeparator1;
    private javax.swing.JSeparator jSeparator2;
    private javax.swing.JSeparator jSeparator3;
    private javax.swing.JSeparator jSeparator4;
    public javax.swing.JTextArea jTextArea2;
    public javax.swing.JTextField jTextField1;
    public javax.swing.JLabel jLabelicon;
    public javax.swing.ImageIcon icone;
    public boolean estadobotao=true;

    // Fim da declaração de variáveis

}// Fim da classe
```

APÊNDICE D – CÓDIGO FONTE – CLASSE PORTAPARALELA.JAVA

```
/*
 * PROJETO FINAL
 * CONTROLE DE ILUMINAÇÃO VIA COMANDO DE VOZ
 * Desenvolvido por Alex Perez dos Santos
 * Criação - 15/07/2008
 * Versão Final - 22/11/2008
 * Classe para a comunicação com a porta paralela
 */

package Projeto;

import javax.swing.JOptionPane;
import parport.ParallelPort;
import NTPortJava.NTPortJava;

public class PortaParalela {

	short endPorta=(short)0xFFE8;
	int dados;
	String erro="Erro na comunicação com a porta paralela";


	ParallelPort porta = new ParallelPort(endPorta);

	// Abre a porta para comunicação
	public void abrePorta(){
		try{
		NTPortJava nt = new NTPortJava();
		nt.EnablePorts(endPorta,endPorta);
		}catch(Exception e){
		    JOptionPane.showMessageDialog(null, erro, "Erro", 0);
		}
	}
	// Fecha a porta para comunicação
	public void fechaPorta(){
		try{
		NTPortJava nt = new NTPortJava();
		nt.DisablePorts(endPorta, endPorta);
		}catch(Exception e){
		    JOptionPane.showMessageDialog(null, erro, "Erro", 0);
		}
	}
	// Envia dados para porta paralela
	public void setPort(int dados){
		try{
```

```
                porta.write(dados);
                }catch(Exception e){
                     JOptionPane.showMessageDialog(null, erro, "Erro", 0);
                }

        }
        // Recebe dados pela porta paralela
        public int getPort(){
                try{
                dados=porta.readOneByte(endPorta);

                }catch(Exception e){
                        JOptionPane.showMessageDialog(null, erro, "Erro", 0);
                }
                return dados;
        }

}// Fim da classe
```