



Escola Superior de Tecnologia e Gestão
Instituto Politécnico da Guarda

RELATÓRIO DE PROJETO

Licenciatura em Engenharia Informática

António Jorge de Almeida Fernandes
novembro | 2012



Instituto Politécnico da Guarda
Escola Superior de Tecnologia e Gestão

**Sistema de Monitorização Remota de Temperatura
e Humidade de Câmaras de Frio**

António Jorge de Almeida Fernandes

Projeto de Engenharia Informática

12 de Novembro de 2012



Instituto Politécnico da Guarda
Escola Superior de Tecnologia e Gestão

**Sistema de Monitorização Remota de Temperatura
e Humidade de Câmaras de Frio**

António Jorge de Almeida Fernandes

Projeto de Engenharia Informática

Orientador: Professor Doutor Carlos Alberto Carreto, Professor da
Unidade Técnico-Científica de Informática da ESTG.

12 de Novembro de 2012

Agradecimentos

O desempenho no desenvolvimento deste projeto não teria sido o mesmo sem a ajuda e apoio de algumas pessoas, às quais gostaria de aqui expressar o meu reconhecimento.

Assim, começo por agradecer ao meu orientador, Prof. Doutor Carlos Alberto Carreto por toda a ajuda, aconselhamento e motivação prestados durante o desenvolvimento do meu trabalho.

À minha família, em especial à minha mãe e meu pai, pela forma como sempre me apoiaram e ajudaram a ultrapassar as barreiras que foram surgindo.

À direção da ESTG pela autorização da utilização da máquina de prototipagem rápida.

Ao Professor Miguel Lourenço pelo empenho na elaboração da caixa em ABS para o alojamento do protótipo desenvolvido no projeto.

Ao Técnico Superior Joaquim Abreu pelo trabalho realizado na produção da caixa em ABS.

Finalmente, um agradecimento muito especial à minha namorada pela compreensão, carinho, paciência e incondicional apoio que sempre demonstrou.

A todos os que estão e não estão aqui indicados, o meu obrigado.

Resumo

Neste trabalho foi desenvolvido um dispositivo eletrônico com capacidade de ler a temperatura e a humidade do local onde estiver instalado, enviar essa informação para um computador onde vai ser guardada para posterior análise por parte do utilizador. Além do envio de dados para uma base de dados num servidor Web, o sistema apresenta os dados num LCD e emite um alerta sonoro quando há uma falha de corrente na fonte de alimentação principal.

Do dispositivo proposto, são descritos todos os componentes e módulos incluídos na solução, as características do software e hardware necessários para a implementação do projeto, assim como o seu desempenho baseado em testes realizados em ambiente de laboratório e industrial.

Palavras Chave

Telemetria, Arduino, Sensores, Wi-Fi, Sistemas Embebidos.

Abstract

In this work an electronic device was developed with the ability to read the temperature and the humidity of the local where it is installed and send the data to a computer where is going to be saved for posterior analysis by users. Besides sending data, the system presents data on an LCD and emits a sound when there is a power failure in the main power supply.

All the components and modules included in the solution are presented, the characteristics of the hardware and software necessary to implement of the project, as well its performance based on tests conducted in laboratory and industrial environment.

Key words

Telemetry, Arduino, Sensors, Wi-Fi.

Conteúdo

1	Introdução	12
1.1	Motivação	12
1.2	Solução	12
1.3	Contribuição	13
1.4	Estrutura do Documento	13
2	Definição do Problema e Objetivos Previstos	14
2.1	Definição do Problema	14
2.2	Objetivos Previstos	14
3	Tarefas e Metodologia	16
3.1	Descrição das tarefas	16
3.2	Metodologia	17
4	Estado da Arte	18
4.1	Introdução	18
5	Implementação da Solução	20
5.1	Introdução	20
5.2	Desenvolvimento do Hardware do Dispositivo	20
5.2.1	Microcontrolador	21
5.2.2	Sensor Temperatura e Humidade	22
5.2.3	Alarme Sonoro	25
5.2.4	LCD	26
5.2.5	Relé	26
5.2.6	Dispositivo Wi-Fi	27
5.2.7	Fonte de Alimentação	28
5.2.8	Relógio de Tempo Real	29
5.2.9	Montagem	29
5.2.10	Caixa de Alojamento	30
5.3	Desenvolvimento do Software do Dispositivo	31
5.3.1	Programa do Microcontrolador	31
5.3.2	Configuração do RTC	33
5.3.3	Configuração do Wi-Fi	34
5.4	Implementação da Base de Dados	36
5.5	Configuração do Servidor	37
5.6	Web Page	38

6	Conclusões e Trabalho Futuro	44
6.1	Conclusões	44
6.1.1	Verificação e Desempenho do Dispositivo	44
6.2	Trabalho Futuro	46
A	Listagem de programas	48
A.1	Programa Microcontrolador	48
A.2	Configurar módulo Wifly utilizador	52
A.3	Programa configurar RTC	65
A.4	Aplicação Servidor	66
A.5	WEB page index	69
A.6	WEB page valida Utilizador	71
A.7	WEB page Adiciona Dispositivo	72
A.8	WEB page Consultar Dispositivo	74
A.9	WEB page altera Dispositivo	78
A.10	WEB page alterar Dispositivo	80
A.11	WEB page Adiciona Conta	83
A.12	WEB page Adicionar Conta	85

Lista de Figuras

3.1	Mapa de Gantt.	17
4.1	Equipamentos para leitura da temperatura e envio sem fios, adaptado de [5][4].	18
5.1	Esquema geral de funcionamento.	20
5.2	Diagrama de blocos do hardware do dispositivo eletrônico.	21
5.3	Microcontrolador e função de cada pino, adaptado de [2].	21
5.4	Sensor RHT03, adaptado de [1].	23
5.5	Esquema ligação entre microcontrolador e sensor temperatura/humidade, adaptado de [1].	24
5.6	Etapas de comunicação entre microcontrolador e sensor RHT03, adaptado de [1].	24
5.7	Diagrama esquemático do circuito do buzzer.	25
5.8	LCD DOGM128, adaptado de [3].	26
5.9	Esquema de ligação entre o microcontrolador e o LCD.	26
5.10	Diagrama esquemático do circuito para ativação do relé.	27
5.11	Modulo Wi-Fi, adaptado de [7].	27
5.12	Modo funcionamento do Wifly, adaptado de [7].	28
5.13	Bateria e fonte de alimentação do dispositivo.	29
5.14	Diagrama esquemático do circuito RTC e conexão com microcontrolador, adaptado de [6].	29
5.15	Placas de circuito impresso com componentes.	30
5.16	Desenhos das placas de circuito impresso.	30
5.17	Caixa de alojamento.	31
5.18	Ambiente de programação, adaptado de [2].	31
5.19	Fluxograma do programa do microcontrolador.	32
5.20	Formato pacote dados que o microcontrolador envia para o servidor.	33
5.21	Programa de configuração do RTC.	33
5.22	Modo de funcionamento do módulo Wifly, adaptado de [7].	34
5.23	Configuração através modo comando via <i>Wireless</i>	34
5.24	Comandos necessários para configurar o módulo Wifly para aceder à rede.	35
5.25	Extrato de código para conexão através de sockets com o módulo Wifly.	35
5.26	Interface da aplicação com o utilizador.	36
5.27	Diagrama classes da base de dados do servidor.	37
5.28	Servidor MiniServX.	38
5.29	Diagrama de casos de uso utilizador.	38

5.30	Diagrama de atividades do utilizador.	39
5.31	Diagrama de sequência para o utilizador fazer o login.	40
5.32	Web page, realizar login.	41
5.33	Diagrama de sequência para inserir nova conta.	41
5.34	Diagrama de sequência para inserir novo dispositivo.	42
5.35	Web page para adicionar um novo dispositivo.	42
5.36	Diagrama de sequência para consultar dados.	43
5.37	Registo da temperatura e humidade.	43
6.1	Taxa de sucesso de entrega dados.	45

Lista de Tabelas

4.1	Principais características dos equipamentos.	19
5.1	Principais características do microcontrolador.	22
5.2	Pinos usados do microcontrolador.	22
6.1	Desempenho do dispositivo.	44
6.2	Comparação do dispositivo desenvolvido com os diapositivos apresentados no capítulo 4.	45

Glossário

ADC — Analog to Digital Converter. Conversor de tensão contínua para um valor digital em formato binário.

API — Application Program Interface. Conjunto de rotinas, funções.

GND — Ground. Massa da fonte de alimentação.

Firmware — Software que está instalado em um dispositivo de hardware.

HACCP — Hazard Analysis and Critical Control Points (Análise de Perigos e Controlo de Pontos Críticos). Sistema que identifica, avalia e controla perigos que sejam significativos para a segurança alimentar.

IDE — Integrated Development Environment. Ambiente integrado para desenvolvimento de software.

IEEE 802.11 — Institute of Electric and Electronics Engineers. Organização internacional que, entre outras atividades, efetua normas no âmbito da engenharia eletrotécnica e eletrónica. A norma 802.11 refere-se a comunicações wireless existindo diversas variantes identificadas por uma letra no final.

IP — Internet Protocol. Principal protocolo de comunicação da Internet.

I2C — Inter-Integrated Circuit bus. Barramento série desenvolvido pela Philips que é usado para a comunicação de dados entre módulos de circuitos eletrónicos.

JSP — Java Server Pages. Páginas HTML que contêm marcadores especiais, dentro dos quais é inserido código Java.

LCD — Liquid Crystal Display. Dispositivo electro-óptico para apresentar imagens ou dígitos.

LED — Light-Emitting Diode. Díodo semicondutor que quando é energizado emite luz visível.

PC — Personal Computer. Computador pessoal.

PHP — Hypertext Preprocessor (originalmente Personal Home Page). Linguagem livre e utilizada para gerar conteúdo dinâmico na World Wide Web.

MCU — Microcontrolador. Abreviatura para o microcontrolador, também pode ser usado uC.

Rede Adhoc — Rede em que não é necessário um Access Point ou ponto de acesso.

RTC — Real-Time Clock. Circuito eletrônico capaz de gerar uma contagem de tempo precisa a partir da oscilação de um cristal.

SDK — Software Developers Kit. Pacote que inclui ferramentas (APIs, linguagens de scripting e interface gráfica de utilizador) necessários para o desenvolvimento de software.

SPI — Serial Peripheral Interface Bus. Protocolo que permite a comunicação do microcontrolador com diversos componentes.

SQL — Structured Query Language. Linguagem de pesquisa declarativa para uma base de dados relacional.

USB — Universal Serial Bus. Interface padrão, que oferece uma plataforma para estabelecer uma conexão plug-and-play entre diversos dispositivos digitais.

Web Page — Página web, isto é, de hipertextos acessíveis geralmente pelo protocolo HTTP na Internet.

WEP — Wired Equivalent Privacy. Wi-Fi Protected Access, modo de acesso seguro a redes wireless.

WPA2 — Wi-Fi Protected Access. Modo de acesso seguro a redes wireless, sucessor do vulnerável WEP.

Wi-Fi — Wireless Fidelity. Consorcio de redes wireless que define normas e regulamentos universais de modo a fornecer interoperabilidade entre redes wireless.

Servidor Web — Programa de computador responsável por aceitar pedidos HTTP de clientes, geralmente os navegadores, e servi-los com respostas HTTP, incluindo opcionalmente dados, que geralmente são páginas web, tais como documentos HTML com objetos embutidos.

Capítulo 1

Introdução

Este documento descreve o trabalho realizado no âmbito da unidade Curricular Projeto de Informática do 3ºano do curso de Engenharia Informática do Instituto Politécnico da Guarda.

1.1 Motivação

A transmissão de dados remota é um recurso fundamental para empresas que necessitam de informações em tempo real como fator estratégico para tomadas de decisões. No sector industrial e agro-industrial cada vez mais há necessidade de informação para tomar decisões de forma rápida e eficiente ou até simplesmente satisfazer os requisitos legais. Atualmente na industria e estabelecimentos que usam câmaras de frio é exigido o registo da temperatura que tipicamente é feito por um funcionário, numa folha, em intervalos de tempo pré definidos que pode ir de 2 ou até 4 vezes ao dia, passando por todas as câmaras ou locais onde os sensores estão instalados. Surgiu então a necessidade de desenvolver um sistema capaz de monitorizar, de forma continua e em tempo real, a temperatura e humidade em câmaras de frio, eliminando assim a falha nos controlos dos registos e alertar os intervenientes caso apareça alguma anomalia, reduzindo os problemas causados por falta destas informações.

1.2 Solução

A solução consiste em desenvolver um dispositivo com capacidade de leitura da temperatura e humidade, amostragem dos valores lidos num LCD e envio dos dados, através de uma rede sem fios, para um PC, cada vez que seja registada uma alteração de um dos valores. Este sistema deve ser facilmente configurado pelo utilizador final de modo a facilitar a sua instalação sem a intervenção de um técnico da especialidade.

Foi desenvolvida uma aplicação para o PC com a funcionalidade de guardar os valores numa base de dados onde o utilizador pode consultar os dados através da escolha do dispositivo pretendido.

1.3 Contribuição

O objetivo é a concepção e implementação de um dispositivo de baixo custo com base em componentes de uso geral, fácil implementação, capaz de fazer a leitura da temperatura e humidade, ter a capacidade de mostrar esses valores em tempo real num LCD e enviar os dados para o servidor via Wi-Fi através de uma rede sem fios que use a norma IEEE 802.11.

1.4 Estrutura do Documento

O documento compreende seis capítulos, para além da presente introdução e de um capítulo de conclusões.

O segundo capítulo incide sobre a definição do problema e objetivos previstos.

No terceiro capítulo é descrita a metodologia utilizada para desenvolver, implementar e testar o sistema.

No quarto capítulo é apresentado o estado da arte.

No capítulo cinco descreve-se a implementação da solução proposta.

Finalmente, no capítulo seis, são apresentadas as conclusões mais relevantes do trabalho, e as perspetivas de desenvolvimento futuro.

Capítulo 2

Definição do Problema e Objetivos Previstos

2.1 Definição do Problema

Os alimentos como peixes, carnes, produtos lácteos e confeccionados, que não são para consumo imediato, deverão ser armazenados a temperaturas entre 1°C e 5°C. Se ocorrer a congelação dos produtos esta deve ser mantida a uma temperatura de -18°C ou inferior em frigoríficos, câmaras frigoríficas ou outro equipamento de frio. Estes limites de temperatura são muito importantes e fazem parte dos requisitos de qualquer plano HACCP que obriga a monitorizações periódicas da temperatura e a conservação dos registos exatos que permitam provar que o plano HACCP se encontra efetivamente implementado. Todas estas etapas exigem registos precisos que podem ser utilizados para avaliar o funcionamento eficaz do sistema, assim como, serem utilizados para corroborar decisões importantes, tais como a frequência das monitorizações, requisitos para a verificação ou para identificar as alterações que visem melhorar o plano HACCP.

Atualmente o registo da temperatura das câmaras de frio é feito tipicamente por um funcionário, num formulário em papel em intervalos de tempo pré definidos que pode ir de 2 ou até 4 vezes ao dia, passando por todas as câmaras ou locais onde os sensores estão instalados.

Este método além de ser dispendioso e obrigar a intervenção humana, não garante que os registos sejam efetuados no tempo indicado. Se houver variações bruscas na temperatura resultante de uma anomalia nas câmaras de frio, só são detetadas quando for efetuada a próxima leitura, o que já pode vir a ser muito tarde em alguns casos, provocando graves prejuízos económicos para o proprietário.

2.2 Objetivos Previstos

Os objetivos a atingir é criar um dispositivo com as seguintes características:

- Leitura automática da temperatura e humidade.

- Mostrar as leituras em tempo real num LCD.
- Emitir um sinal sonoro se houver uma falha na alimentação principal da câmara de frio.
- Enviar as leituras para um servidor via Wi-Fi.
- Fácil instalação.
- Baixo custo.
- Consultar via Web as leituras.

Capítulo 3

Tarefas e Metodologia

3.1 Descrição das tarefas

As principais tarefas são:

- Tarefa 1 — Análise dos requisitos do hardware e software.
- Tarefa 2 — Definição dos módulos de hardware e software que constituem o sistema.
- Tarefa 3 — Escolha dos componentes para os módulos de hardware.
- Tarefa 4 — Desenho dos circuitos eletrónicos dos vários módulos de hardware.
- Tarefa 5 — Definição da arquitetura da aplicação para o microcontrolador.
- Tarefa 6 — Fabrico das placas de circuito impresso e montagem dos componentes dos vários módulos de hardware.
- Tarefa 7 — Teste do dispositivo.
- Tarefa 8 — Construção da caixa para alojar o módulo eletrónico.
- Tarefa 9 — Definição dos requisitos e construir da base de dados.
- Tarefa 10 — Definição da arquitetura da aplicação para o servidor.
- Tarefa 11 — Implementação e testes da aplicação do servidor.
- Tarefa 12 — Definição da arquitetura da aplicação para o utilizador.
- Tarefa 13 — Implementação e testes da aplicação do utilizador.
- Tarefa 14 — Elaboração do relatório do projeto.

O agendamento das tarefas é apresentado na figura 3.1.

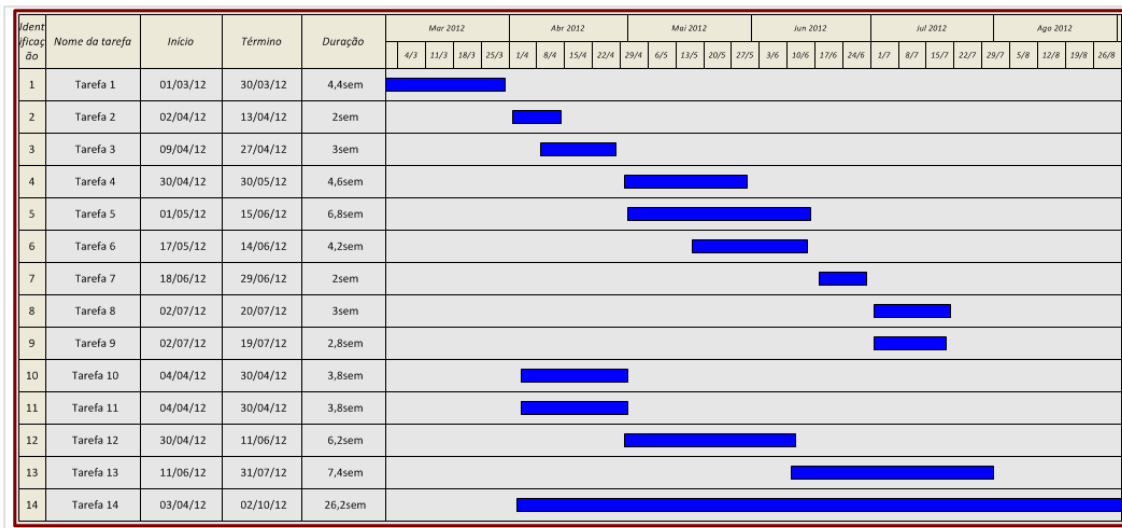


Figura 3.1: Mapa de Gantt.

3.2 Metodologia

A metodologia utilizada para desenhar, implementar o dispositivo foi a seguinte:

- Usar UML e diagramas de caso se uso para fazer a análise do sistema.
- Usar a linguagem de programação Java através do IDE NetBeans para desenvolver as aplicações e o SDK do arduino para programar o microcontrolador.
- Usar o MYSQL para construir a base de dados e o MiniServX para o servidor.
- Usar o Diptrace para desenhar as placas de circuito impresso e para as fabricar usar o processo fotográfico.

Capítulo 4

Estado da Arte

4.1 Introdução

Atualmente, a concorrência económica torna a inovação, pesquisa, desenvolvimento e lançamento de novos produtos, um meio essencial de conservação e conquista de faixas de mercado. Os mercados encontram-se cada vez mais saturados e a procura acaba conseqüentemente por incidir sobre aqueles produtos que apresentem diferenças inovadoras.



Figura 4.1: Equipamentos para leitura da temperatura e envio sem fios, adaptado de [5][4].

Na figura 4.1 são apresentados 3 equipamentos dos muitos que existem no mercado que permitem leituras de temperatura e humidade. A escolha incidu no preço, sendo apresentado um para cada segmento, barato, intermédio e caro.

Um dos módulos muito utilizado em redes de sensores sem fios é o XBee. Estes módulos comunicam entre si numa frequência de 2.4Ghz usando o protocolo IEEE 802.15.4. Fabricados pela MAXTREAM, incluem todo hardware e lógica necessários para implementar uma rede, podendo comunicar até uma distância de 1600 metros se usarmos a versão Pro.

Um dos equipamentos mais baratos que se encontra no mercado é o Digi XBee [5]. Usa tecnologia XBee e permite obter leituras de temperatura e humidade. Pode envia-las até cerca 120m, se encontrar em linha de vista (Outdoor) ou 40m, se encontrar em ambiente fechado com paredes entre o equipamento e o recetor (Indoor). Este equipamento só permite fazer leituras no local onde estiver instalado,

para ler a temperatura dentro de uma câmara frigorífica tinha que ser instalado no seu interior.

O Temperature@lertWifi [8], vem incluído com um sensor externo de temperatura e usa o protocolo 802.11 b/g para comunicar com o recetor ou através de cabo de Ethernet. Depois de fazer a instalação as leituras realizadas podem ser observadas numa Web page.

O SureCross DX80 [4] representa um dos equipamentos mais caros. É comercializado pela prestigiada marca Banner, sendo um sensor desenvolvido para aplicações industriais. É alimentado com tensões de 10V a 30V dc. Não vem equipado com sensores mas permite ligar diversos tipos de sensores. Comunica com o recetor por rádio frequência 900MHz ou 2.4GHz. Vem equipado com um pequeno LCD para mostrar as leituras e possui leds para indicar o status em que se encontra (ligado, erro na leitura, erro a enviar dados).

Na tabela 4.1 são apresentadas as principais características dos equipamentos mencionados anteriormente.

Equipamento	DigiXbee	Temperature@lertWifi	SureCross DX80
Alimentação	3 x AA alkaline 1.5V	110/240V	10 to 30V dc
Consumo	35 mA / 45 mA	desconhecido	60 mA
Frequência	2.4 GHz	2.4 GHz	900 MHz ou 2.4GHz
Norma	802.15.4	802.11b/g	desconhecido
Potência transmissão	1.25 mW (+1 dBm)	desconhecido	18 dBm
Performance Indoor	40 m	desconhecido	3.2 Km
Performance Outdoor	120 m	desconhecido	3.2 Km
Sensor tensão	não	não	não
Sensor temperatura	-18° C to 55° C	-55° C to 125° C	-50 to 1,000 C
Sensor Humidade	0 to 100%	desconhecido	desconhecido
LCD	não	não	sim
Configuração leitura	desconhecido	sim	sim
Software	desconhecido	sim	sim
Alerta sonoro	não	não	não
Preço	100€	300€	1438€

Tabela 4.1: Principais características dos equipamentos.

Pretende-se agora desenvolver um dispositivo que agrupe algumas das funcionalidades dos equipamentos apresentados anteriormente usando protocolo 802.11 b/g para poder comunicar diretamente com um PC, com capacidade de leitura de diferentes tipos de sensores de temperatura e que esteja equipado com um alerta sonoro.

Capítulo 5

Implementação da Solução

5.1 Introdução

Como já foi referido, o objetivo do projeto é a implementação de um dispositivo capaz de monitorizar a temperatura e a humidade. O sistema deve ter a capacidade de mostrar esses parâmetros em tempo real num LCD e enviar os dados via Wi-Fi para um servidor quando um desses parâmetros é alterado. Os parâmetros a registar é a temperatura e humidade, para isso recorreu-se a um sensor que tem a capacidade de medir a temperatura e humidade ao mesmo tempo, seguidamente essa leitura é enviada a um microcontrolador que prepara as leituras e as envia via Wi-Fi para um servidor como se pode ver na figura 5.1. As leituras podem ser consultadas numa Web Page por vários utilizadores.

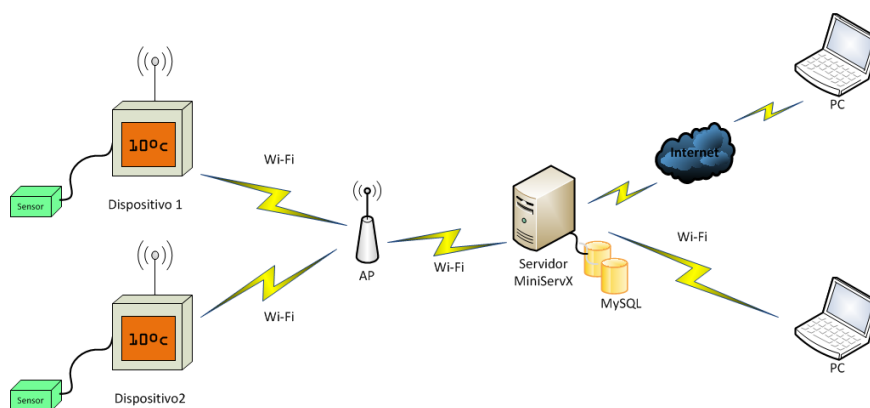


Figura 5.1: Esquema geral de funcionamento.

Assim temos duas componentes neste projeto a componente de hardware e a componente do software que é constituída por vários módulos, programação do microcontrolador, servidor e aplicação final para se poderem observar os dados.

5.2 Desenvolvimento do Hardware do Dispositivo

O hardware refere-se à parte física do dispositivo, ou seja, o conjunto de módulos

que constituem todo o sistema, sendo composto por diversas partes eletrônicas e mecânicas.

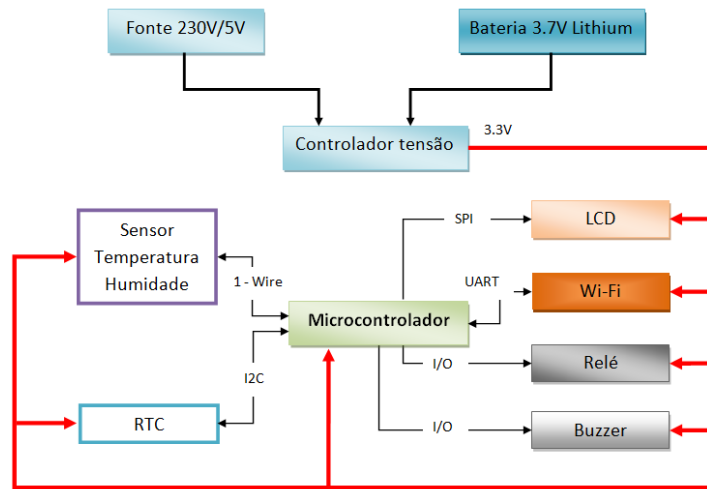


Figura 5.2: Diagrama de blocos do hardware do dispositivo eletrônico.

Através do diagrama de blocos da figura 5.2, podemos identificar os principais componentes e as ligações entre eles. O dispositivo é alimentado com uma tensão de 230V (tensão da rede elétrica) e por uma bateria de lítio de 3.7V 1700mA, que é usada para manter o seu funcionamento por pequenos intervalos de tempo, quando há uma falha de corrente na linha principal, possibilitando assim alertar com um aviso sonoro a falha de energia. A seguir é feita uma descrição individual de cada módulo.

5.2.1 Microcontrolador

A figura 5.3 apresenta o microcontrolador Atmega328 [2] que vem com um *bootloader* que permite carregar uma aplicação no microcontrolador usando uma placa Arduino UNO. O ambiente de desenvolvimento do Arduino pode ser utilizado para escrever e compilar o programa de acordo com a aplicação antes de o enviar para a placa via USB.

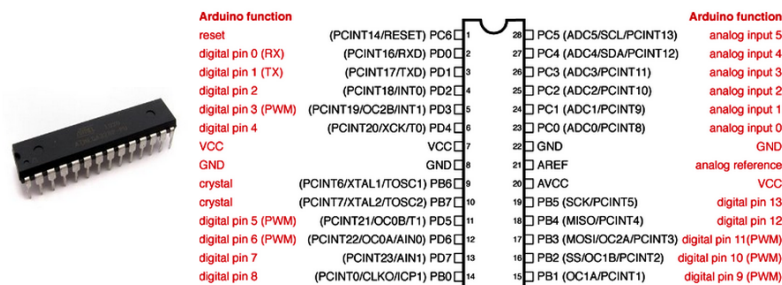


Figura 5.3: Microcontrolador e função de cada pino, adaptado de [2].

As aplicações são escritas numa linguagem muito parecida com a linguagem C usando um ambiente de programação gratuito. Devido à sua popularidade este pos-

sui funções para quase tudo o que possa ser necessário para uma aplicação, funções com entradas e saídas analógicas e digitais, comunicações série, síncronas (SPI) e assíncronas (UART). Os fabricantes de alguns dispositivos também disponibilizam APIs com as funções necessárias para estabelecer ligação e comunicar com o Arduino, como é o caso do sensor de temperatura e o módulo LCD usados no projeto.

Na tabela 5.1 podemos observar as principais características deste microcontrolador.

Características:	
Microcontrolador	Atmega328
Tensão funcionamento	5V
Pinos E/S digitais	14 (dos quais 6 podem ser saídas PWM)
Pinos de entrada analógica	6
Corrente CC por pino E/S	40mA
Corrente CC para o pino 3,3V	50mA
Flash Memory	32 KB (Atmega328) dos quais 0,5KB são utilizados pelo bootloader
SRAM	2KB (Atmega328)
EEPROM	1 KB (Atmega328)
Velocidade de Clock	16 MHz

Tabela 5.1: Principais características do microcontrolador.

O carregamento do programa é feito usando a placa Arduino UNO, depois o microcontrolador é transferido para a placa de circuito impresso construída para alojar todos os componentes deste dispositivo. A tabela 5.2 mostra os pinos usados do microcontrolador para cada módulo.

Pino microcontrolador	Função
13	Sensor humidade/Temperatura
4	Wifi, pino 2
5	Wifi, pino 3
15	Wifi, pino 8 (enable modo Adhoc)
14	Wifi, pino 5 (não usado)
13	Buzzer
14	Relé
26	Detetor falta tensão principal
27	SCL do DS1307 (RTC)
28	SDA do DS1307 (RTC)
19	SCL do LCD
17	SI do LCD
16	CS1B do LCD
15	A0 do LCD
18	Iluminação LCD

Tabela 5.2: Pinos usados do microcontrolador.

5.2.2 Sensor Temperatura e Humidade

Os sensores de temperatura e humidade estão divididos em dois grupos, analógicos e digitais.

Os analógicos como é o caso do sensor de temperatura LM35, é um sensor que apresenta uma saída de tensão linear relativa à temperatura do local onde este se encontra no momento em que for alimentado, tendo em sua saída um sinal de 10mV para cada grau Célsius de temperatura. Não necessita de calibração. Tem a vantagem de ser calibrado diretamente em graus Célsius e opera na faixa de -55°C até 150°C . É alimentado de 4-20V e consome cerca de 60mA.

Os sensores digitais normalmente usam a ligação *1-Wire* para comunicar com o microcontrolador, que é um tipo de comunicação projetado pela Dallas Semiconductor Corp que permite enviar dados a baixa velocidade e a grandes distâncias. em comparação com outras normas similares (I2C), usando apenas dois cabos de comunicação e alimentação, um para os dados e outro para o GND.

Este tipo de comunicação permite uma maior precisão na leitura dos dados que são lidos e convertidos em binário no local onde se encontra o sensor e enviados de forma digital para o microcontrolador eliminando assim problemas que podem surgir com a resistividade do fio condutor usado. Problemas que ocorrem quando são usados termopares ou outro tipo de sensor similar que use uma saída analógica em função da temperatura ou humidade medida. Os digitais operam numa faixa de temperatura de -55°C a 125°C e têm uma precisão de $\pm 0,5^{\circ}\text{C}$ na faixa de -10°C a $+85^{\circ}\text{C}$.

O DS18B20 é um exemplo de um sensor digital de temperatura que tem a particularidade de ter gravado um número de série de 64bits permitindo assim ligar diversos dispositivos em série usando a conexão 1-Wire. Recentemente apareceram no mercado sensores digitais que incorporam num só dispositivo um sensor de temperatura e humidade como é o caso do SHT1 ou SHT15, mas que tem um custo bastante elevado.



Figura 5.4: Sensor RHT03, adaptado de [1].

O RHT03 [1], apresentado na figura 5.4 é um sensor de temperatura e humidade que opera na faixa de -40°C a $+80^{\circ}\text{C}$ e 0-100%RH (humidade relativa) respetivamente, possui uma precisão de $\pm 1\%$ para a temperatura e $\pm 0,1\%$ para a humidade. Constituído por um sensor de humidade capacitivo e um termistor para medir a temperatura não necessita de componentes externos, visto que possui internamente um conversor A/D e uma memória na qual os dados são convertidos.

Não necessita de calibração para funcionar. Este dispositivo leva no máximo 2s para realizar as leituras e fazer a conversão em formato digital, este tempo não representa problemas para aplicações de medição de temperatura e humidade porque não ocorrem grandes variações neste intervalo de tempo. A tensão de alimentação é de 3.3V a 6 V consumindo apenas 1mA quando realiza a leitura e 40uA em *Stand-by*, quando alimentado a 3.3V. A comunicação com o microcontrolador é feita via 1-Wire, permitindo estabelecer conexões até 100m não necessitando componentes adicionais.

Neste projeto foi usado o sensor RHT03 devido ao seu baixo custo, à tensão de alimentação 3.3V e à gama de temperatura pretendida.

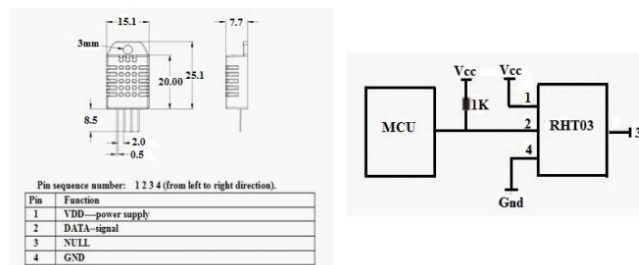


Figura 5.5: Esquema ligação entre microcontrolador e sensor temperatura/humidade, adaptado de [1].

A ligação do microcontrolador com o sensor RHT03, figura 5.5 é feita pela porta do microcontrolador (pino 13), que se encontra-se no seu estado inicial com um nível lógico 1, quando pretende pedir os dados ao sensor coloca a porta a 0 durante 1ms, retorna ao nível 1 durante 20-40us e aguarda pela resposta do sensor, passo A da figura 5.6. Quando o RHT03 deteta o sinal de inicio coloca a porta no nível baixo e tem cerca de 80us para preparar os dados a enviar, depois começa o envio dos dados como se pode observar no passo B.

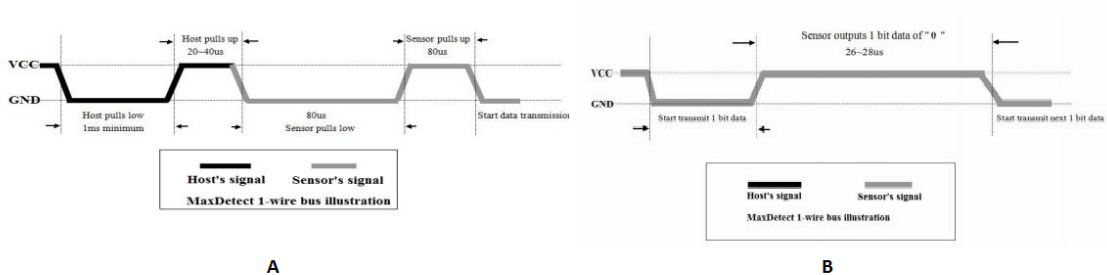


Figura 5.6: Etapas de comunicação entre microcontrolador e sensor RHT03, adaptado de [1].

Os dados enviados são constituídos por duas partes, uma integral e outra decimal, e existe um checksum para confirmar a veracidade dos dados. O exemplo

seguinte apresenta como os dados são enviados:

Exemplo:

0000 0010 1000 1100 0000 0001 0101 1111 1110 1110

0000 0010 1000 1100 - valor da humidade em binário que representa 652 em decimal logo temos 65.2% da humidade relativa.

0000 0001 0101 1111 - valor da temperatura em binário que representa 351 em decimal logo temos 35.1°C.

1110 1110 são os 8 bits que representam o checksum que é usado para verificar a veracidade dos dados através da seguinte fórmula:

$$\text{Checksum sum} = 0000\ 0010 + 1000\ 1100 + 0000\ 0001 + 0101\ 1111 = 1110\ 1110$$

Quando o bit mais significativo da temperatura é 1, significa que a temperatura é negativa.

Por exemplo: 1000 0000 0110 0101 corresponde a -10.1°C.

5.2.3 Alarme Sonoro

Na figura 5.7 é apresentado o alarme sonoro que é constituído por um dispositivo tipo buzzer que emite um som quando falta a tensão principal ao dispositivo. Foi usado um buzzer que emite um som com uma frequência de 2048Hz e uma intensidade de 85dBa. O consumo é de 35mA a 3.3V. Para isso o microcontrolador gera uma onda quadrada com a frequência pretendida através do pino 12 e aplica o sinal a um transístor que tem como função amplificar o sinal e alimentar os pinos do buzzer.

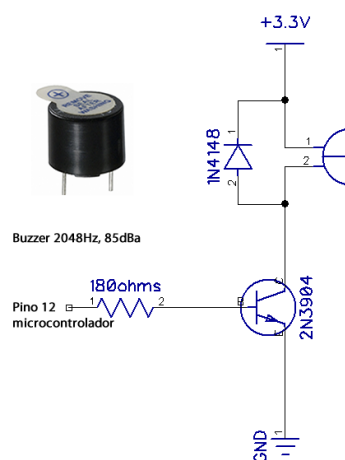


Figura 5.7: Diagrama esquemático do circuito do buzzer.

5.2.4 LCD

Na figura 5.8 é apresentado o LCD EA DOGM128W-6 fabricado pela Electronic Assembly [3] permite ver em tempo real a temperatura e a humidade medida pelo sensor, usa um display do tipo gráfico de 128x64 pixels, e dimensão de 55x46mm.

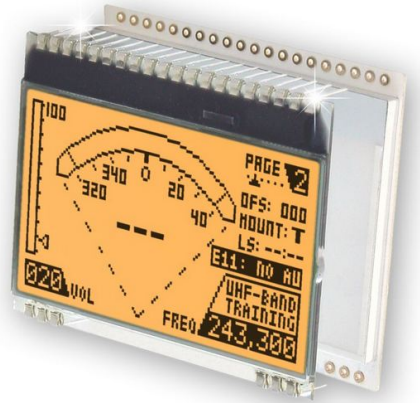


Figura 5.8: LCD DOGM128, adaptado de [3].

Este display é caracterizado por usar um interface de comunicação do tipo SPI(usando apenas 4 portas para controlar a parte gráfica do LCD e uma porta para ligar ou desligar a iluminação, neste caso foi usada a cor âmbar) esta ligação permite poupar muitas portas do microcontrolador, figura 5.9 pode-se observar as ligações efetuadas.

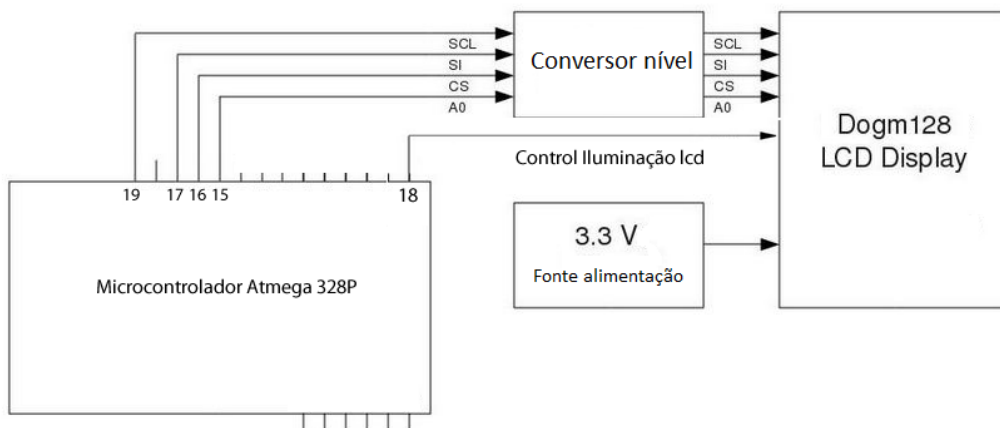


Figura 5.9: Esquema de ligação entre o microcontrolador e o LCD.

5.2.5 Relé

Na figura 5.10 é apresentado o interface com o exterior que é feito através da ativação de um relé quando a temperatura ou a humidade se encontra fora do intervalo

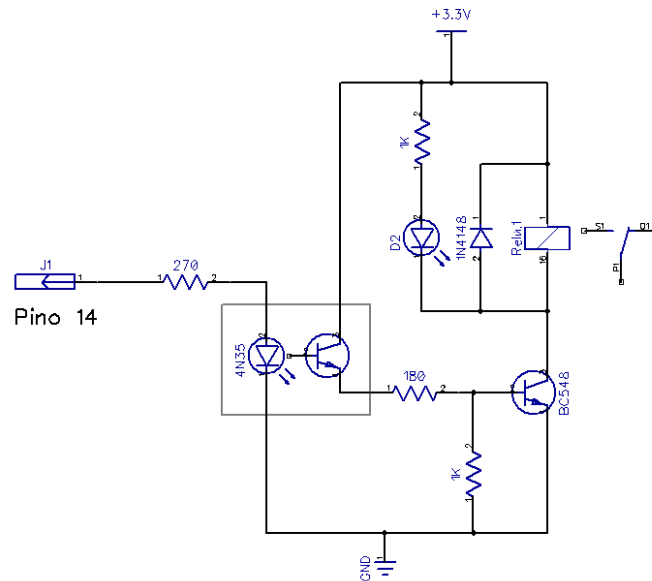


Figura 5.10: Diagrama esquemático do circuito para ativação do relé.

pré configurado, assim o microcontrolador coloca o pino 14 a nível alto que vai atuar o opto acoplador que tem a função de fazer o isolamento entre o circuito de potência e o circuito de comando, dado o opto acoplador não ter corrente suficiente para ativar a bobine do relé foi usado um transistor para o ativar. Este circuito permite assim ativar equipamentos com tensões de alimentação mais elevadas, até 230V com uma corrente máxima de 8A, como por exemplo uma ventoinha ou um sistema de aquecimento. Esta funcionalidade é para ser configurável através do software.

5.2.6 Dispositivo Wi-Fi



Figura 5.11: Módulo Wi-Fi, adaptado de [7].

Para transmitir os dados via Wi-Fi, foi usado o módulo RN-XV, também designado por Wifly [7] da empresa Roving Networks, figura 5.11. Este módulo Wi-Fi é

baseado no módulo eletrônico RN-171 que incorpora a norma IEEE 802.11b/g, processador de 32 bits, pilha TCP/IP, real time, unidade de gerenciamento de alimentação e entradas para sensores analógicos. Vem já com um *firmware* desenvolvido pelo fabricante que facilita a integração e minimiza o tempo de desenvolvimento. O módulo necessita de apenas quatro conexões (PWR, TX, RX e GND) para criar uma conexão de dados sem fio, a potência de transmissão do sinal pode ser configurável desde 0dBm a +12dBm tendo um consumo muito reduzido cerca de 4uA no modo sleep, 40mA a ler dados e 180mA a transmitir dados a 10dBm. Consegue transmissões de dados a uma taxa de 464Kbps e suporta os protocolos de segurança WEP, WPA e WPA2-PSK. A tensão de funcionamento é de 3.3V e vai comunicar com o microcontrolador através da seguinte ligação, do pino 4 ao pino UART_TX do Wifly, pino 5 ao UART_RX do Wifly e o pino 15 que tem como função passar o dispositivo para o modo ADHOC que ocorre quando o pino 8 do Wifly toma nível lógico 1). O fabricante deste dispositivo disponibiliza um programa (Tera Term) para poder fazer a configuração remotamente através de um PC usando a rede wireless local ou também pode ser configurado através da porta serie do PC. Para configurar o dispositivo basta apenas estabelecer a ligação e entrar no modo de comando através da introdução do seguinte comando \$\$\$, aguardar pela resposta AOK , em seguida indicar a que rede vai estabelecer a ligação, o modo de autenticação e no fim fazer o save e o reboot. Depois de configurado este dispositivo envia os dados que aparecem na porta UART como ilustrado na figura 5.12. De modo a testar o alcance de transmissão foram testados dois tipos de antena cujo o resultado dos testes é apresentado na secção 6.1.1.

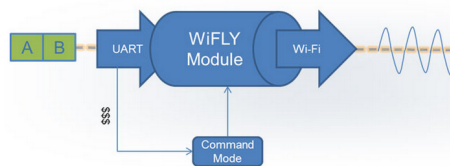


Figura 5.12: Modo funcionamento do Wifly, adaptado de [7].

5.2.7 Fonte de Alimentação

Na figura 5.13 pode-se ver a fonte de alimentação principal, módulo comercializado pela Cincon Electronics [4] e a bateria de backup. Todo o dispositivo é alimentado pela tensão da rede elétrica 230V que é convertida em 5V através da fonte de alimentação. Como já foi referido anteriormente, todo o circuito trabalha a 3.3V, para isso foi usado um regulador de tensão para estabilizar a tensão a 3.3V. A fonte de alimentação alimenta também o carregador da bateria de *lithium* que tem como função fornecer corrente durante a falha de tensão da rede principal. Fornece uma tensão de 3.7V e uma corrente de 1700mA o que é suficiente para alimentar o circuito por pequenos intervalos de tempo, cerca de 60 minutos. É carregada através do controlador MCP73831T.



Figura 5.13: Bateria e fonte de alimentação do dispositivo.

5.2.8 Relógio de Tempo Real

Na figura 5.14 é apresentado o RTC (Real Time Clock) [6], módulo capaz de gerar uma contagem de tempo precisa a partir da oscilação de um cristal. A contagem de tempo é mantida mesmo que a energia principal seja desligada, através de uma bateria de backup. É muito utilizado nos computadores, geralmente integrado ao sistema conhecido como BIOS (Basic Input / Output System), onde armazena a data completa com ano, mês, dia, hora, minuto e segundo. A comunicação com o microcontrolador é feita via I2C.

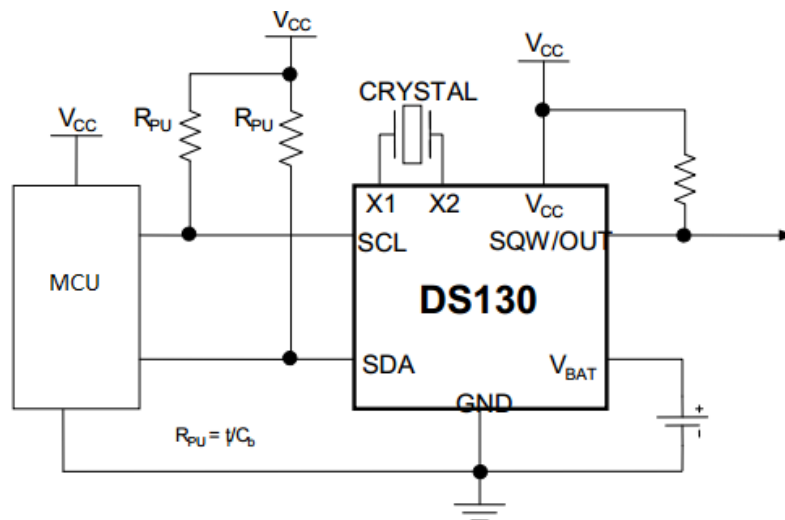


Figura 5.14: Diagrama esquemático do circuito RTC e conexão com microcontrolador, adaptado de [6].

5.2.9 Montagem

A montagem dos circuitos eletrônicos foi dividida em 3 placas de circuito impresso, como se pode observar na figura 5.15. Uma para alojar o LCD, outra para o controlador de tensão e carregador da bateria de lithium e a placa principal onde se encontra o microcontrolador, o módulo Wifly, o circuito da ativação do buzzer e do

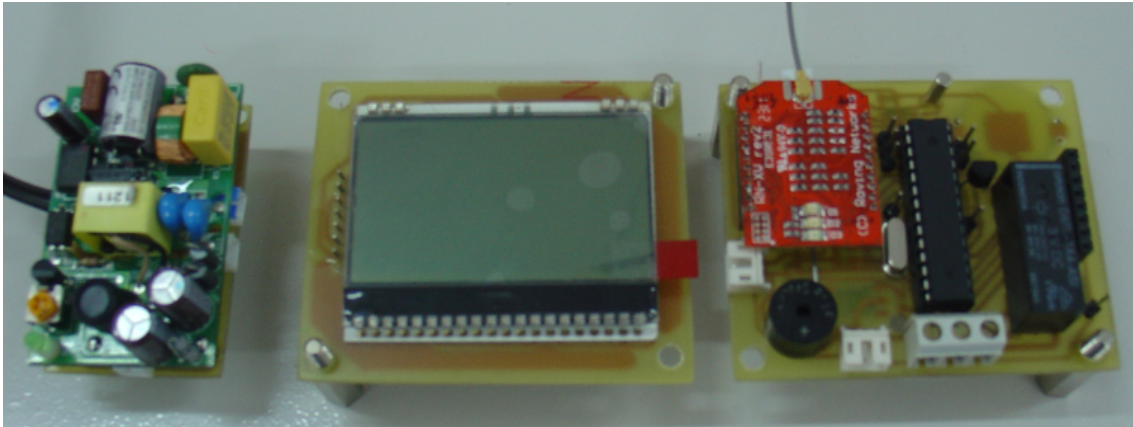


Figura 5.15: Placas de circuito impresso com componentes.

relé, o RTC que tem uma pilha de 3V tipo CR2032 para não perder a data e hora e ainda um regulador de tensão para manter a tensão de alimentação em 3.3V.

Na figura 5.16 podemos observar a distribuição dos componentes nas placas, desenhadas usando o IDE DipTrace e fabricadas usando o processo fotográfico, usando uma placa de fibra de vidro foto sensibilizada.

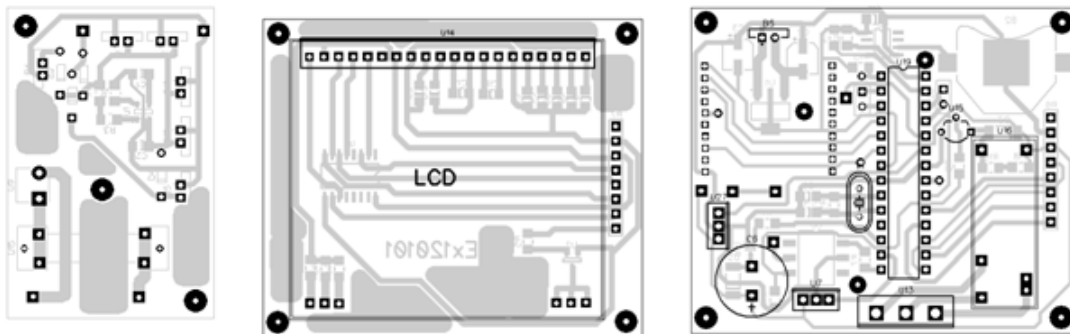


Figura 5.16: Desenhos das placas de circuito impresso.

5.2.10 Caixa de Alojamento

Na figura 5.17 podemos observar a caixa 123x75x48, fabricada em ABS recorrendo a uma máquina de prototipagem rápida que consegue imprimir em 3D. Esta tecnologia permite produzir uma réplica exata da caixa necessária num curto intervalo de tempo de modo a obter um produto personalizado, o que facilita a instalação das diferentes placas no seu interior, separar fisicamente a fonte de alimentação e a bateria das outras placas. Um dos aspetos negativos é o custo elevado da impressão da caixa, cerca de 50 euros só em material ABS.

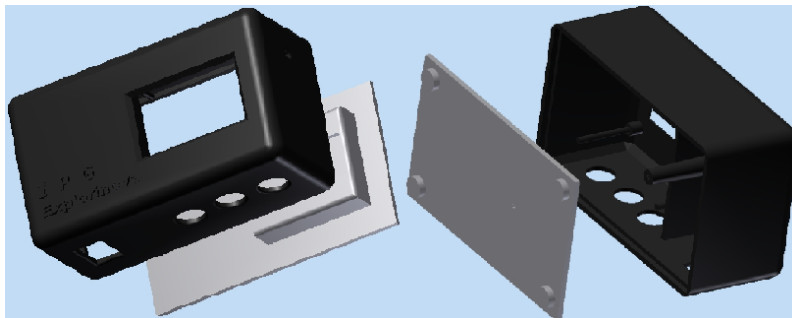


Figura 5.17: Caixa de alojamento.

5.3 Desenvolvimento do Software do Dispositivo

O desenvolvimento do software do projeto está dividido em quatro partes: a programação do microcontrolador, Web Page, que é responsável pela visualização e envio de informação para o utilizador; o módulo da base de dados que mostra como as leituras são armazenadas, e por fim o módulo funcional responsável pelo processamento dos dados recebidos no lado do servidor.

5.3.1 Programa do Microcontrolador

O ambiente de desenvolvimento do Arduino, figura 5.18 contém um editor de texto para escrita do código, uma área de mensagem, uma área de controle de informações, uma barra de ferramentas com botões para funções comuns e um conjunto de menus. Esse ambiente conecta-se ao hardware Arduino Uno via USB para carregar os programas e se comunicar com eles. Os programas escritos com o ambiente Arduino são chamados de sketches.

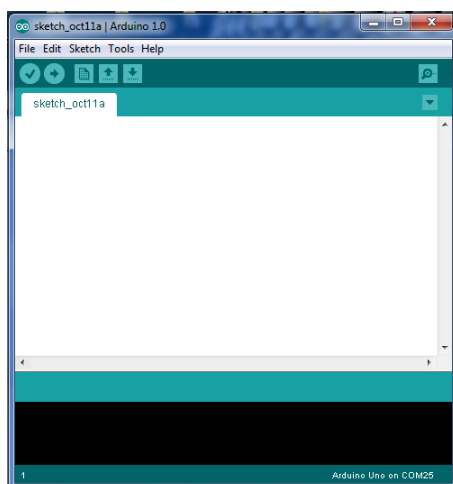


Figura 5.18: Ambiente de programação, adaptado de [2].

O ambiente de desenvolvimento do Arduino possibilita que os programas sejam organizados em torno de duas funções, mas primeiro é necessário declarar as variáveis e fazer o include das APIs a usar.

`include <Wire.h>` - para usar a porta I2C para poder comunicar com o RTC.
`include "RTClib.h"` - para obter a data e hora do RTC.
`include "DHT.h"` - para obter as leituras do sensor de temperatura e humidade.
`include <SoftwareSerial.h>` - comunicar via serie com o módulo Wifly.
`include <WiFlyHQ.h>` - configurar, enviar e receber dados do Wifly.
`include <U8glib.h>` - configurar, inicializar e atualizar o LCD.

Em seguida na função `setup()` que é executada uma única vez no início do programa, é usada para iniciar e configurar as variáveis, inicializar a porta UART, RTC, sensor de temperatura e o LCD.

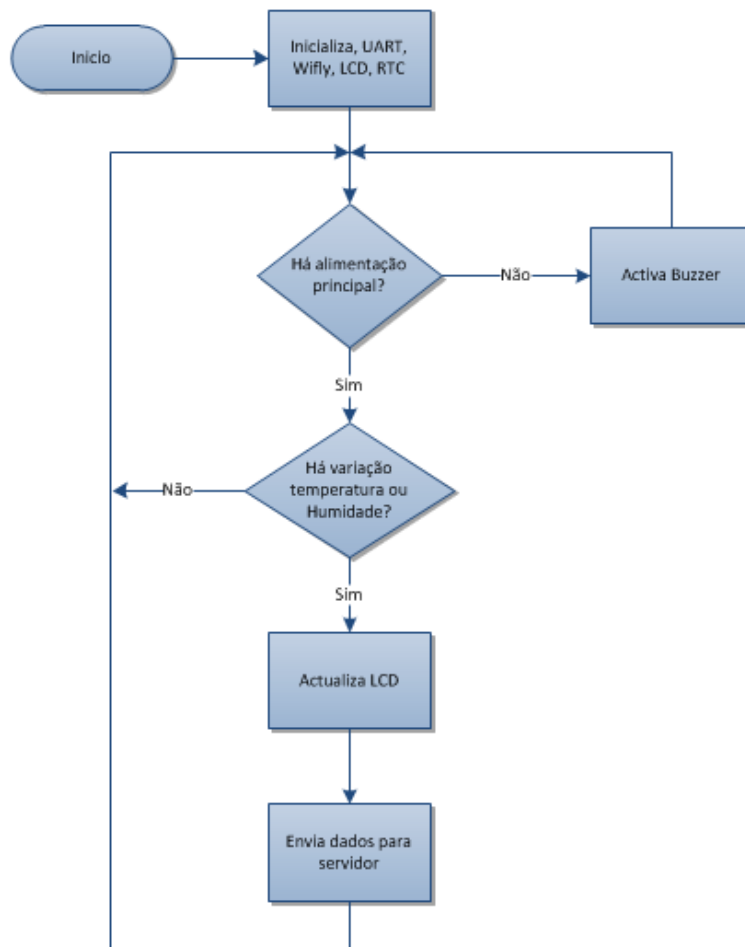


Figura 5.19: Fluxograma do programa do microcontrolador.

Na função `loop()`, que é executada repetidamente até que o módulo seja desligado, o programa vai confirmar a presença de tensão através do pino 23 que foi previamente configurado como entrada analógica. Se encontrar um nível lógico baixo ativa um sinal sonoro através do buzzer. Ao fazer a leitura do sensor, vai comparar os dados com a ultima leitura recebida e se houver alteração vai atualizar o LCD e enviar os dados para o servidor. Na figura 5.19 pode-se observar o fluxograma do programa do microcontrolador.

MAC:	Mac do dispositivo	Data:	Data	SHumidade:	Leitura humidade	STemperatura:	Leitura temperatura
MAC:	00:06:66:71:87:cf	Data:	2012-08-12 12:05:02	SHumidade:	24.30	STemperatura:	33.0

Figura 5.20: Formato pacote dados que o microcontrolador envia para o servidor.

Na figura 5.20 é mostrado o pacote de dados que o microcontrolador constroi para entregar ao módulo Wifly, para este enviar para o servidor via Wi-Fi. É através do endereço MAC do módulo Wi-Fi que o servidor vai identificar o dispositivo. O programa do microcontrolador é apresentado no apêndice A.1.

5.3.2 Configuração do RTC

O DS1307 utilizado como RTC, necessita de ser configurado com a data e a hora atual quando ligado pela primeira vez, ou quando é retirada a bateria de 3V. Esta operação pode ser feita recorrendo ao ambiente de desenvolvimento do Arduino para fazer o programa e passá-lo para o microcontrolador. Depois basta realizar a ligação ao DS1307 que já foi referida anteriormente na secção de hardware e reiniciar o microcontrolador. Este dispositivo comunica via I2C sendo necessário fazer o include da API que vem com o compilador *include <Wire.h>* e o *include RTCLib.h* para poder comunicar e carregar a data e hora no DS1307, na figura 5.21 e no apêndice A.3 é apresentado o respetivo código.

```
#include <Wire.h>
#include "RTCLib.h"

RTC_DS1307 RTC;

void setup () {
  Serial.begin(57600);
  Wire.begin();
  RTC.begin();

  if (! RTC.isrunning()) {
    RTC.adjust(DateTime("AUG 27 2012", "12:54:56"));
  }
}

void loop () {
  DateTime now = RTC.now();
}
```

Figura 5.21: Programa de configuração do RTC.

5.3.3 Configuração do Wi-Fi

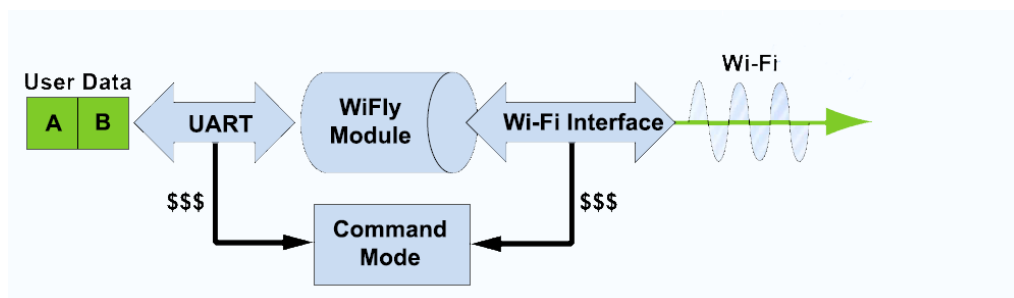


Figura 5.22: Modo de funcionamento do módulo Wifly, adaptado de [7].

O módulo Wi-Fi, para poder aceder e enviar os dados a uma rede, necessita de ser configurado primeiro. É necessário dizer com quem vai e como vai comunicar sendo necessário introduzir alguns comandos.

Estes comandos como podem ser introduzidos de duas maneiras distintas, usando uma conexão serial ou usando o programa Tera Term Pro disponibilizado pelo fabricante o que possibilita estabelecer uma conexão remota via *Wireless*.

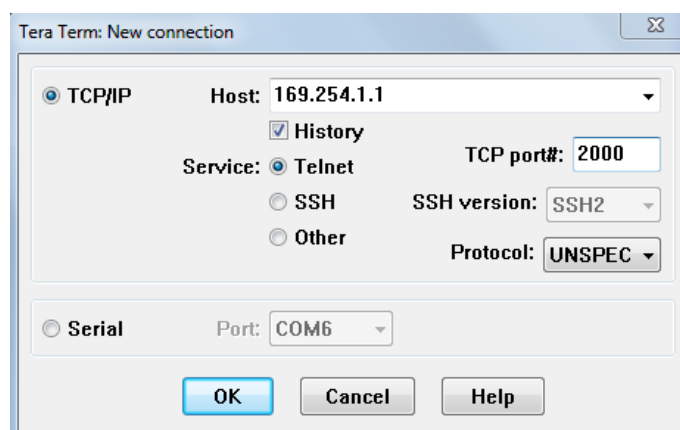


Figura 5.23: Configuração através modo comando via *Wireless*.

Para usar o modo remoto é necessário alimentar o módulo a 3.3V e ligar o pino 8 a +3.3V para entrar no modo Adhoc.

Estabelecer a conexão entre o computador e o módulo.

Abrir o programa Tera Term, introduzir o IP 169.254.1.1, porta 2000 para estabelecer a ligação via telnet como se pode ver na figura 5.23.

Depois de conectado digitar \$\$\$, figura 5.24, o dispositivo responde com HELLO o que significa que se encontra no modo comando. Agora basta introduzir os comandos apresentados a seguir para configurar a rede.

Fazer o **save** e o **reboot** para o módulo reiniciar e ligar-se automaticamente ao AP.

Quando o módulo se conseguir ligar ao AP o led verde acende-se e todos os dados que surjam na porta UART são enviados para o servidor.

```

169.254.1.1 - Tera Term VT
File Edit Setup Control Window Help
*HELLO*
*OPEN*
set wlan ssid teste
AOK
<2.31> set wlan auth 3
AOK
<2.31> set wlan phrase passwifly
AOK
<2.31> set ip host 127.000.000.001
AOK
<2.31> set ip remote 63306
AOK
<2.31> save
Storing in config

```

Figura 5.24: Comandos necessários para configurar o módulo Wifly para aceder à rede.

set wlan ssid nome de sua rede - para especificar o nome da rede.

set wlan auth 3 - se usar o modo WPA1 ou WPA2-PSK para melhorar a segurança na transmissão dos dados deve ser usado um método de codificação dos dados que o AP possa suportar.

set wlan phrase palavra pass - palavra pass para poder aceder à rede.

set ip host ip do servidor - indicar o IP do servidor para onde os dados vão ser enviados.

set ip remote 8081 - porta de comunicação.

```

...
String IP = jtfIP.getText();
int port = Integer.valueOf(jtfPort.getText()).intValue();
connection = new Socket(IP, port);
j1Mensagem.setText("Ligação efectuada com sucesso.");

out = new PrintStream(connection.getOutputStream(), true);
in = new BufferedReader(new InputStreamReader(connection.getInputStream()));

cn = new ConfigureNetwork();
cn.start();
jbConnect.setText("Desligar");
Ligar = 1;
} else {
    connection.close();
    jbConnect.setText("Ligar");
    j1Mensagem.setText("Ligação terminada.");
    Ligar=0;
}
...

```

Figura 5.25: Extrato de código para conexão através de sockets com o módulo Wifly.

Para o utilizador final não ter dificuldades em configurar o módulo Wifly foi criada uma aplicação em Java através do uso de *Sockets*, que possibilitam a comu-

nicação entre aplicações distintas, o código 5.25 mostra como foi criado em Java. A figura 5.26 apresenta o ambiente gráfico. O programa em Java é apresentado no apêndice A.2.

Através deste programa o utilizador facilmente programa o módulo Wifly para se poder conectar à rede existente sem fazer o uso do programa Tera Term o que poderia criar alguma confusão.

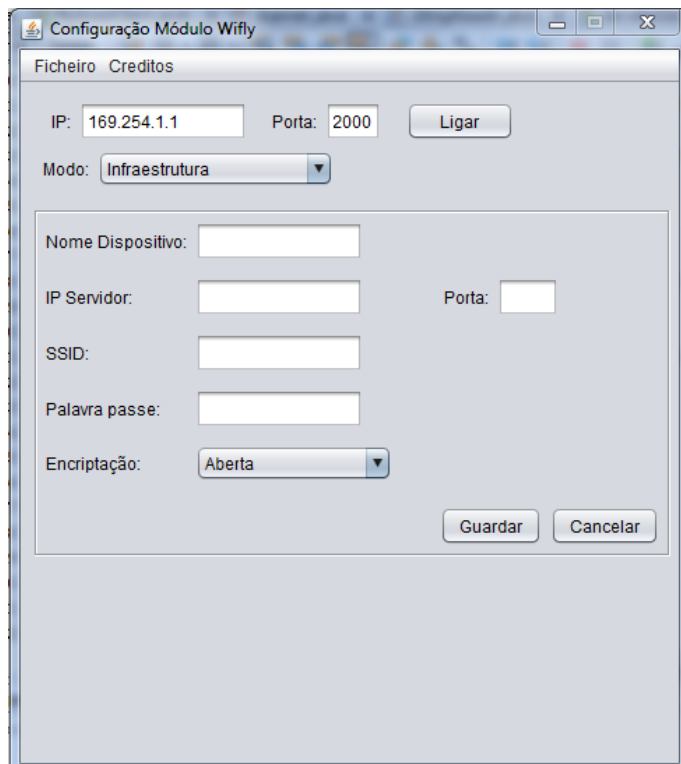


Figura 5.26: Interface da aplicação com o utilizador.

5.4 Implementação da Base de Dados

Na figura 5.27 podemos observar o diagrama de classes da base de dados do servidor que foi elaborada em MySQL que é um sistema de gestão de base de dados. Utiliza a linguagem SQL (Linguagem de Consulta Estruturada, do inglês Structured Query Language), como interface. É atualmente uma das ferramentas mais utilizadas para elaborar base de dados pelo facto de ser gratuita e de fácil instalação e utilização.

A base de dados é constituída por duas tabelas que vão armazenar a informação dos utilizadores e o seu perfil de utilização. A tabela dispositivo para guardar a informação relativa ao dispositivo, a tabela sensor para indicar os sensores que cada dispositivo tem e por fim a tabela dados para guardar as leituras efetuadas por cada um deles.

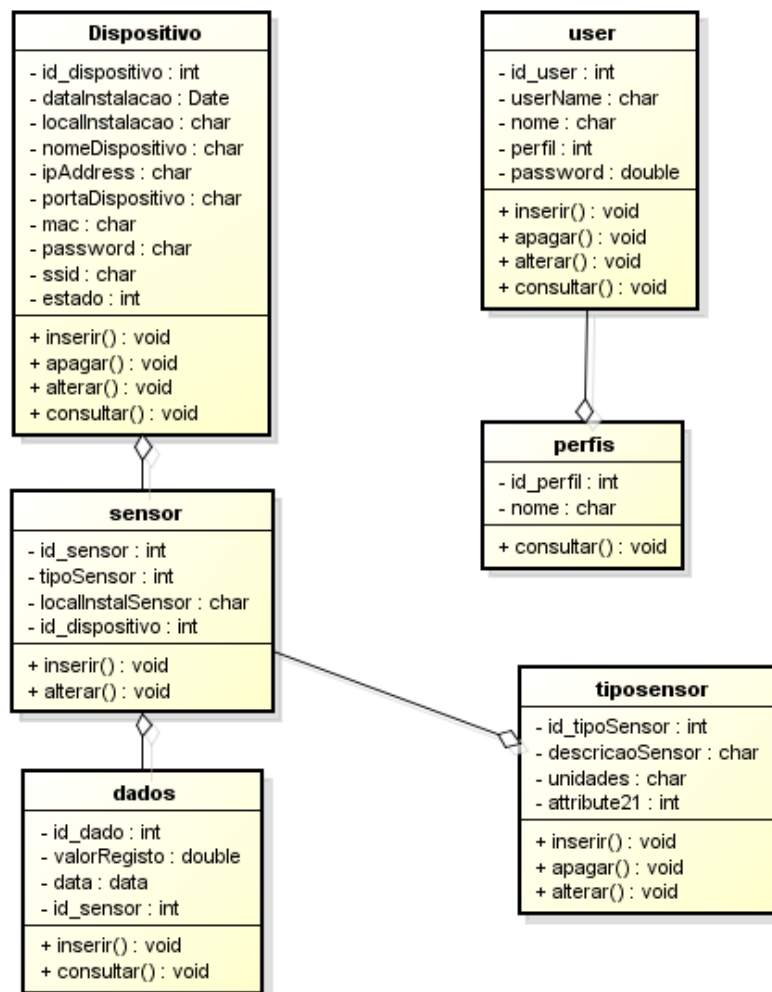


Figura 5.27: Diagrama classes da base de dados do servidor.

5.5 Configuração do Servidor

Na figura 5.28 podemos observar o IDE do servidor MiniServX PHP, é um servidor Web que pode ser executado automaticamente quando se liga o PC. Vem com PHP e MySQL 5, também aceita SQLite. É uma ferramenta grátis para experimentação mas para uso comercial tem um custo de 50euros.

Para receber os dados foi desenvolvida uma aplicação em *Java* que tem por função estabelecer uma nova conexão com um novo cliente *Socket* quando se tenta ligar. Verifica se é um cliente válido através do MAC address enviado pelo dispositivo em comparação com o que está na base de dados, se for válido e o resto do pacote corresponder ao pré definido realiza a extração das leituras e insere-as na base de dados. O programa do servidor é apresentado no apêndice A.4.

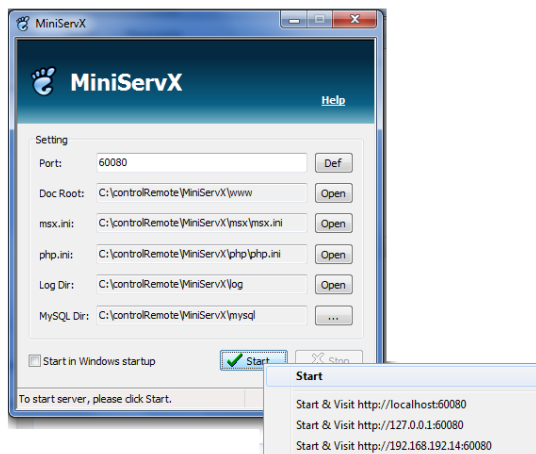


Figura 5.28: Servidor MiniServX.

5.6 Web Page

A Web Page permite que sejam visualizadas as leituras de um determinado dispositivo. Desenvolvida em JSP o utilizador tem ainda a possibilidade de adicionar, alterar ou apagar novos dispositivos e utilizadores. Cada caso de uso do diagrama mostrado na figura 5.29 que apresenta um requisito funcional do sistema.

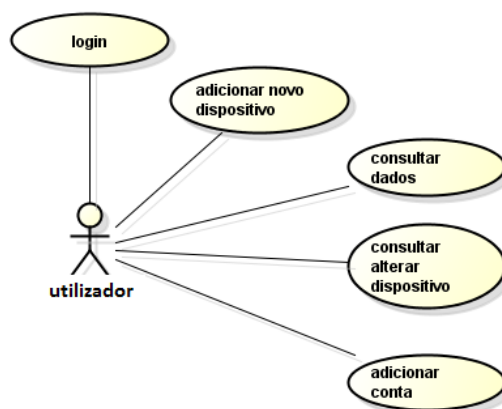


Figura 5.29: Diagrama de casos de uso utilizador.

Efetuar login: O sistema deve permitir apenas o acesso de um utilizador autorizado.

Adicionar novo dispositivo: Permite adicionar um novo dispositivo de comunicação de dados.

Consultar dados: Permite consultar os dados recebidos através da seleção do dispositivo a partir do endereço MAC.

Consultar alterar dispositivo: Permite apagar, alterar dados relativos ao dispositivo.

Adicionar conta: Permite adicionar um novo utilizador.

Logout: Permite sair da aplicação.

Na figura 5.30 é apresentado o diagrama de atividades do utilizador que apresenta o fluxo das atividades.

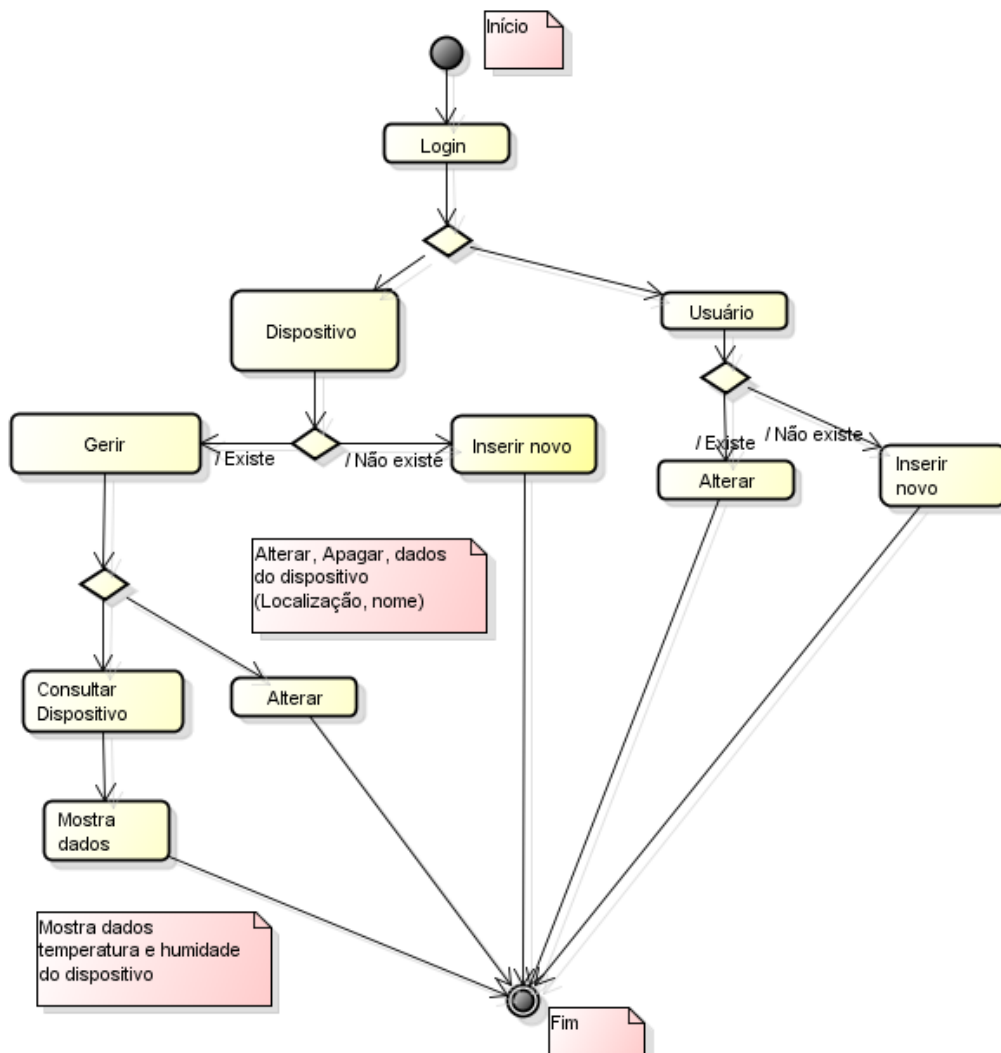


Figura 5.30: Diagrama de atividades do utilizador.

Descrição do diagrama de atividades:

Atores

Utilizador.

Pré-condição:

O utilizador tem que estar registado na base de dados.

Fluxo de Eventos (caminho básico):

1. O caso de uso começa quando o utilizador seleciona o menu dispositivo.
 - (a) Se selecionar consultar dispositivo.
 - i. O sistema mostra as leituras do dispositivo selecionado e o caso de uso termina.
 - (b) Se selecionar inserir novo.

- i. O sistema mostra a Web Page adicionar novo dispositivo.
 - ii. O utilizador insere os dados.
 - iii. O sistema insere novo dispositivo na base de dados e o caso de uso termina.
- (c) Se seleccionar alterar.
- i. O utilizador selecciona o dispositivo.
 - ii. O sistema mostra o dados atuais.
 - iii. O utilizador insere os novos dados.
 - iv. O sistema altera os dados na base de dados e o caso de uso termina.

Fluxo de Secundário (caminho alternativo):

1. O caso de uso começa quando o utilizador selecciona o menu Conta.
 - (a) Se seleccionar inserir novo utilizador.
 - i. O sistema mostra a Web Page adicionar novo utilizador.
 - ii. O utilizador insere os dados.
 - iii. O sistema insere novo utilizador na base de dados e o caso de uso termina.
 - (b) Se seleccionar alterar.
 - i. O utilizador selecciona o utilizador.
 - ii. O sistema mostra o dados atuais.
 - iii. O utilizador insere os novos dados.
 - iv. O sistema altera os dados na base de dados do utilizador e o caso de uso termina.

A figura 5.31 mostra o diagrama de sequência para o utilizador fazer o login para efetuar consultas ou inserir novos dispositivos, para isso tem que estar registado.

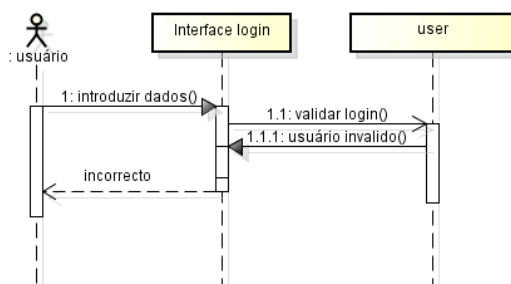


Figura 5.31: Diagrama de sequência para o utilizador fazer o login.

A figura 5.32 apresenta a Web Page para o utilizador realizar o login. O programa da Web Page é apresentado no apêndice A.5 e A.6.

Para que o utilizador possa efetuar consultas, inserir novos dispositivos, terá que estar registado, onde indicará o seu nome, username e password. No registo

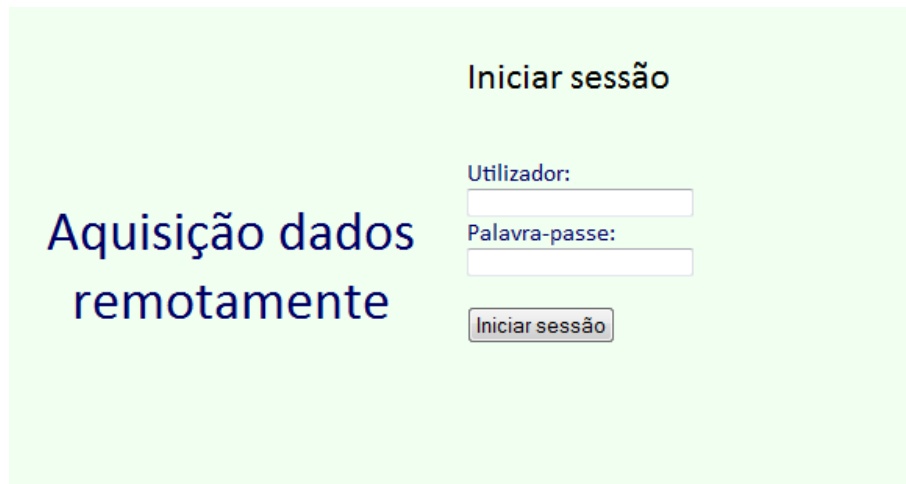


Figura 5.32: Web page, realizar login.

é atribuído o perfil do utilizador, o diagrama de sequência é apresentado na figura 5.33. Com o uso do perfil pretende-se limitar o acesso aos dados, por exemplo o perfil operador só tem acesso à Web Page das consultas das leituras do dispositivo. O perfil administrador permite adicionar e configurar novos dispositivos e utilizadores. O programa da Web Page é apresentado no apêndice A.11 e A.12.

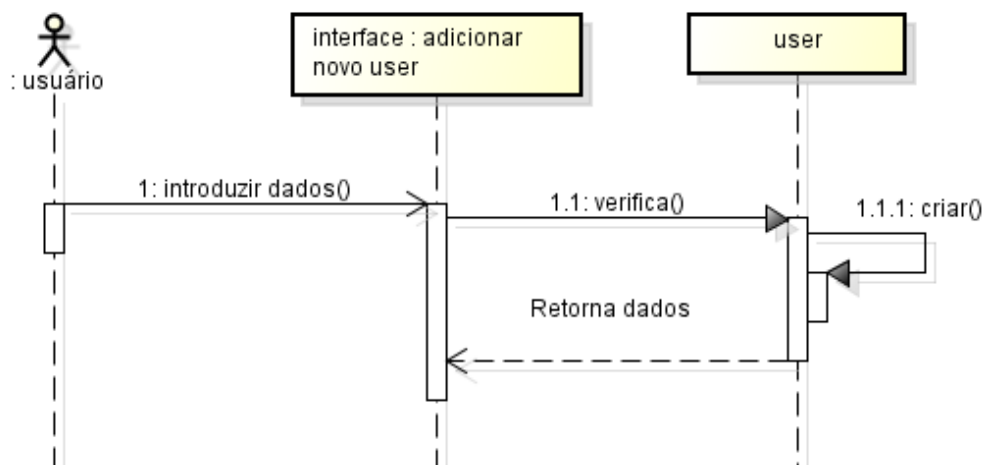


Figura 5.33: Diagrama de sequência para inserir nova conta.

Na figura 5.34 é apresentado o diagrama de sequência para adicionar um novo dispositivo e na figura 5.35 a Web Page para realizar essa tarefa. O endereço MAC é um atributo obrigatório porque é através dele que servidor vai identificar o dispositivo. O nome do dispositivo e o local de instalação e os dados relativos à rede também são atributos muito importantes. O programa da Web Page é apresentado no apêndice A.7, A.9 e A.10.

Depois do utilizador registado pode proceder à consulta das leituras efetuadas por cada dispositivo, a figura 5.36 mostra o diagrama de sequência para esta tarefa. O programa da Web Page é apresentado no apêndice A.8.

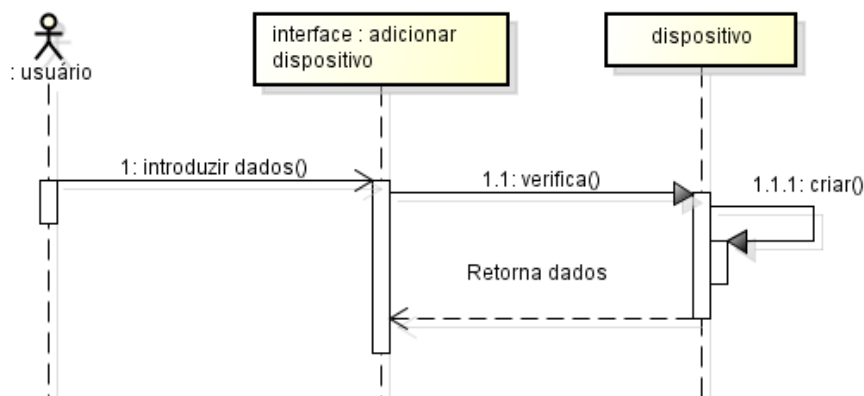


Figura 5.34: Diagrama de sequência para inserir novo dispositivo.

Aquisição dados Remotamente

Inicio Dispositivo Configurar Conta Ajuda Sair

Adicionar novo dispositivo

Data de instalação:	<input type="text" value="2012-08-12"/>
Mac address:	<input type="text" value="00:06:66:71:87:cf"/>
Nome Dispositivo:	<input type="text" value="teste1"/>
Local de instalação:	<input type="text" value="IPG"/>
Ip:	<input type="text" value="192.168.1.85"/>
Porta:	<input type="text" value="8081"/>
SSID:	<input type="text" value="explorinova"/>
Password:	<input type="text" value="123456789"/>
Estado:	<input type="text"/>

Figura 5.35: Web page para adicionar um novo dispositivo.

Para isso basta escolher o MAC do dispositivo e é apresentada uma tabela com a data, o local onde o dispositivo está instalado com todas as leituras da temperatura e humidade obtidas até à data atual como se pode observar na figura 5.37.

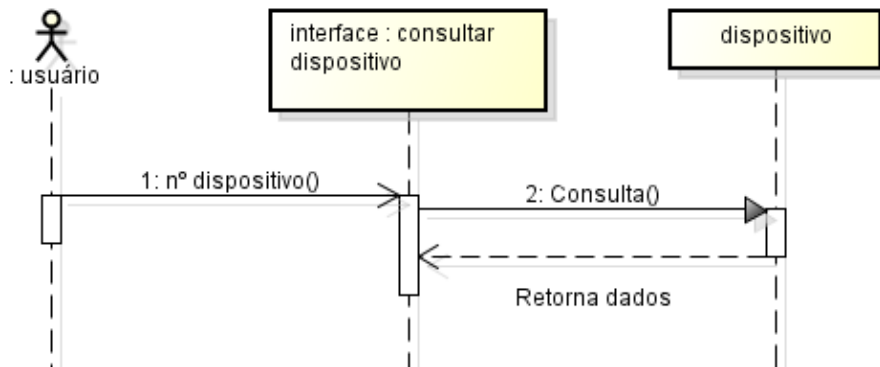


Figura 5.36: Diagrama de sequência para consultar dados.

Aquisição dados Remotamente

[Inicio](#) [Dispositivo](#) [Configurar](#) [Conta](#) [Ajuda](#) [Sair](#)

Dispositivo:

Data	Local	Sensor	Valor
2012-09-07 12:27:34.0	IPG	Temperatura	34.30
2012-09-07 12:27:34.0	IPG	Humidade	25.10
2012-09-07 11:10:42.0	IPG	Temperatura	31.30
2012-09-07 11:10:42.0	IPG	Humidade	28.40
2012-09-07 11:10:38.0	IPG	Humidade	28.60
2012-09-07 11:10:38.0	IPG	Temperatura	31.10
2012-09-07 11:10:34.0	IPG	Temperatura	31.10
2012-09-07 11:10:34.0	IPG	Humidade	28.70
2012-09-07 11:10:30.0	IPG	Temperatura	31.00
2012-09-07 11:10:30.0	IPG	Humidade	28.80

Figura 5.37: Registo da temperatura e humidade.

Capítulo 6

Conclusões e Trabalho Futuro

6.1 Conclusões

6.1.1 Verificação e Desempenho do Dispositivo

No decorrer da implementação da solução foram realizados vários testes intermédios com diversos micro controladores e sensores de temperatura de modo a obter mais rapidamente a solução pretendida, recaindo a escolha pelos componentes já apresentados.

Depois de implementada a solução na placa de circuito impresso foram realizados testes ao equipamento em ambiente de laboratório e em ambiente industrial. Os itens de teste foram a duração da bateria de backup, número máximo de transmissões em modo backup e taxa de sucesso de entrega de dados, em função do local e distância ao AP das quais conseguimos obter os resultados apresentados na tabela 6.1.

Parâmetro	Resultado
Autonomia bateria	60 minutos
Consumo médio	40 mA
Leituras modo backup	800
Performance Indoor	50m
Performance Outdoor	60m

Tabela 6.1: Desempenho do dispositivo.

Em ambiente de laboratório indoor(dispositivo distanciado e separado por paredes do AP), a distância ao AP é o principal influenciador no desempenho do dispositivo, com distâncias superiores a 30m começam a ser registadas falhas na entrega das leituras e demora na conexão entre o dispositivo e o servidor chegando a serem registados tempos superiores a 1 minuto. Em Outdoor (dispositivo em linha de vista com AP) o alcance aumenta e são detetadas muito menos falhas na entrega das leituras.

Para determinar a taxa de entrega das leituras foram registadas as alterações que ocorrem no dispositivo através da visualização no LCD durante 15 minutos para a

Equipamento	DigiXbee	@lertWifi	SureCross DX80	novo
Alimentação	3 x AA alkaline 1.5V	110/240V	10 to 30V dc	110/240V
Consumo	35 mA / 45 mA	desconhecido	60 mA	40mA
Frequência	2.4 GHz	2.4 GHz	900 MHz ou 2.4GHz	2.4 GHz
Norma	802.15.4	802.11b/g		802.11b/g
Potência transmissão	1.25 mW (+1 dBm)	desconhecido	18 dBm	desconhecido
Performance Indoor	40 m	desconhecido	3.2 Km	50m
Performance Outdoor	120 m	desconhecido	3.2 Km	60m
Sensor tensão	não	não	não	sim
Sensor temperatura	-18° C to 55° C	-55°C to 125°C	-50 to 1,000 C	-40°C to 80°
Sensor Humidade	0 to 100%	desconhecido	desconhecido	0 to 100%
LCD	não	não	sim	sim
Configuração leitura	desconhecido	sim	sim	não
Software	desconhecido	sim	sim	sim
Alerta sonoro	não	não	não	sim
Preço	100€	300€	1438€	95€

Tabela 6.2: Comparação do dispositivo desenvolvido com os dispositivos apresentados no capítulo 4.

distância de 10m, 20m, 30m, 40m, 50m e comparadas com as leituras que foram registadas na base de dados. Foi observado que quando aplicado numa zona industrial, onde estavam várias máquinas industriais em pleno funcionamento, não foram registados alterações significativas dos testes realizados em ambiente de laboratório. Foi observado que o tipo de antena influencia diretamente na taxa de sucesso de entrega de dados como se pode observar na figura 6.1.

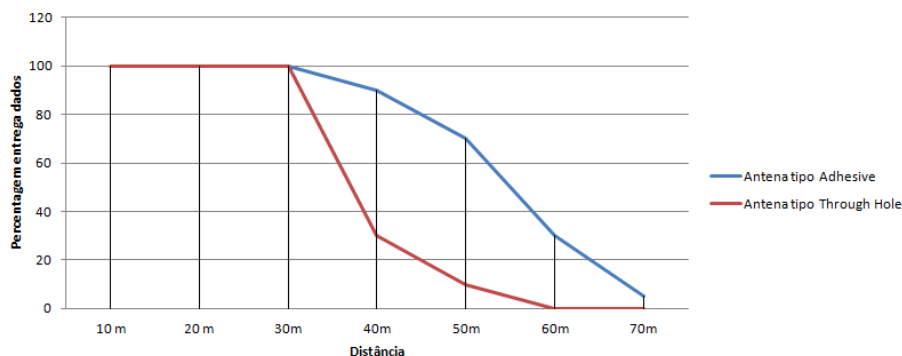


Figura 6.1: Taxa de sucesso de entrega dados.

A tabela 6.2 compara o novo dispositivo com os dispositivos apresentados no capítulo 4.

Os objetivos gerais traçados para este projeto foram alcançados, uma vez que se conseguiu construir um dispositivo com capacidade de enviar as leituras para um computador via Wi-Fi usando a norma 802.11 b/g por um custo de 95€ .

A maior dificuldade encontrada foi na escolha do hardware, devido há dificuldade em encontrar documentação ou APIs para poder interligar os diferentes módulos. Foi escolhido o microcontrolador com bootloader Arduino para o qual quase todos fabricantes de hardware disponibilizam APIs para poder comunicar com este dimi-

nuindo assim o tempo de execução do projeto. Outra dificuldade foi a escolha do LCD porque existem diversos modelos no mercado que depois de algumas aquisições e testes com foram registadas dificuldades na visualização dos dados a uma distância considerável ou simplesmente a não existência de APIs para comunicação com o microcontrolador.

Quando se tenta implementar projetos que envolvem hardware todos os parâmetros dos componentes ou módulos a usar devem ser tomados em consideração sobre pena de serem comprados e não satisfazer a real necessidade, para isso deve ser realizado anteriormente um plano e um estudo exaustivo sobre as necessidades do dispositivo e da aplicação.

6.2 Trabalho Futuro

O dispositivo respondeu positivamente às necessidades pretendidas quando aplicado a testes num cenário de aplicação real. Há possibilidade agora de explorar e implementar novas funcionalidades a nível de software e hardware.

Seria interessante que o dispositivo fosse capaz de armazenar as leituras em memória quando não conseguisse estabelecer a comunicação com o servidor adicionando um módulo de memória via I2C para executar esta tarefa.

Permitir a possibilidade da introdução da configuração do horário das leituras, através da aplicação do utilizador, dado o dispositivo já estar equipado com módulo RTC.

Comunicar com dispositivos móveis, para poder enviar alertas em tempo real permitindo o utilizador poder intervir mais rapidamente.

Testes de alcance na transmissão das leituras mais exaustivos em locais com diferentes características, restaurantes, cafés, ambiente urbano.

Bibliografia

- [1] Rht03 datasheet. <http://dlnmh9ip6v2uc.cloudfront.net/datasheets/Sensors/Weather/RHT03.pdf>.
- [2] arduino. arduino uno. <http://arduino.cc/en/Main/ArduinoBoardUno>, April 2012.
- [3] ELECTRONIC ASSEMBLY. Dogm128.e datasheet. <http://www.lcd-module.com/eng/pdf/grafik/dogm128e.pdf>, October 2009.
- [4] cincon. http://www.cincon.com.tw/products_3.html, Junhe 2012.
- [5] Digi. Digi datasheet. <http://xbeestore.lojavirtualfc.com.br/listaprodutos.asp?idloja=16187&idproduto=2561443&q=Digi+XBee+Sensor+-+Temperatura%2C+Umidade+e+Lumin%E2ncia>, Junhe 2012.
- [6] MAXIM. Ds1307 datasheet. <http://datasheets.maximintegrated.com/en/ds/DS1307.pdf>, 2008.
- [7] rovingnetworks. Wifly datasheet. http://www.rovingnetworks.com/products/RN_XV, 2012.
- [8] Temperature@lert. lert datasheet. <http://www.temperaturealert.com/Wireless-Temperature-Store/Temperature-Alert-WiFi-Sensor.aspx>, Junhe 2012.

Apêndice A

Listagem de programas

A.1 Programa Microcontrolador

```
#include <Wire.h>
#include "RTClib.h"

#include "DHT.h"
#include <SoftwareSerial.h>
#include <EEPROM.h>
#include <WiFlyHQ.h>

#include <U8glib.h>
RTC_DS1307 RTC;

SoftwareSerial mySerial(2, 3);
U8GLIB_DOGM128 u8g(13, 11, 10, 9);

int ledPin = 12;
#define DHTPIN 7
DHT dht(7, DHT22);

String MAC="00:06:66:71:87:cf";

int PortaLigarModoAP = 4; // era a 9 pino 6 no chip
int estado = HIGH;
int tensao = LOW;
int horaservidor;
int PinTensao =A0;
int PinBuzzer =6;
int PinRelay = 8;
char buf[32];
int opcaotemp;
int opcaohum;
int falhaleitura=0;
float h ;
float t ;
float humAnt;
float tempAnt;
int alteracao;
int estadoBuzzer;
char Dados;
char Buffer[20];
byte index=0;
int SemServidor;
WiFly wifly;
char floatBufferT[16];
char floatBufferH[16];
char value2[16];
uint8_t estadoLCD =0;
```

```

//*****
void setup() {

    estadoBuzzer = 0;

    pinMode(PortaLigarModoAP, OUTPUT);
    pinMode(PinTensao, INPUT);
    pinMode(PinBuzzer, OUTPUT);
    pinMode(PinRelay, OUTPUT);
    pinMode(ledPin, OUTPUT);

    digitalWrite(PortaLigarModoAP, LOW);
    Serial.begin(9600);
    Serial.println("Iniciar leitura do sensor!");

    mySerial.begin(9600);
    mySerial.flush();

    wifly.begin(&mySerial, &Serial);
    Wire.begin();
    RTC.begin();

    dht.begin();
    inicializaLCD();

}
//*****
void loop(){

    DateTime now = RTC.now();

    DetectaTensao();
    LerSensor();
    atualizaLCD();
    delay(2000);

    if (alteracao != 0){
        AbreLigacaoServidor();
        enviaDadosParaServidor( now, MAC, h, t);
        mySerial.flush();
    }

}

//*****
void enviaDadosParaServidor(DateTime now, String MAC, float h, float t)
{
    mySerial.println("enviar dados ");
    mySerial.print("MAC:");
    mySerial.println(MAC);
    mySerial.print("Data:");
    mySerial.print(now.year(), DEC);
    mySerial.print('-');
    mySerial.print(now.month(), DEC);
    mySerial.print('-');
    mySerial.print(now.day(), DEC);
    mySerial.print(' ');
    mySerial.print(now.hour(), DEC);
    mySerial.print(':');
    mySerial.print(now.minute(), DEC);
    mySerial.print(':');
    mySerial.println(now.second(), DEC);

    mySerial.print("SHumidade:");
    mySerial.println(h);
    mySerial.print("STemperatura:");
    mySerial.println(t);
}

```

```

// delay(500);

    mySerial.flush();
}

//*****
void FechaServidor()
{
    mySerial.print("$$$");
    delay(300);
    mySerial.println("close");
    mySerial.println("EXIT");
}
//*****
int DetectaTensao()
{

    tensao = digitalRead(PinTensao);

    if(tensao == 0){
        estadoBuzzer = 1;
        Buzzer(estadoBuzzer);
    }
    else{
        estadoBuzzer = 0;
        Buzzer(estadoBuzzer);
    }

    return estadoBuzzer;
}
//*****

void LerSensor()
{
    h = dht.readHumidity();
    t = dht.readTemperature();

    VerificaValores(h, t, humAnt, tempAnt);
    humAnt=h;
    tempAnt=t;
}
//*****
int VerificaValores(float h, float t, float humAnt, float tempAnt)
{
    float hum = h;
    float temp = t;

    if( t!=tempAnt || h!= humAnt){
        return alteracao = 1;
    }
    else if ( t == tempAnt || h == humAnt){
        return alteracao = 0;
    }
}
//*****
void Buzzer(int estadoBuzzer)
{
    if (estadoBuzzer == 1){
        digitalWrite(PinBuzzer,HIGH);
        analogWrite(PinBuzzer, 50);
        delay(200);
        analogWrite(PinBuzzer,0);
        //delay(200);
    }
    else {
        analogWrite(PinBuzzer,0);
    }
}
//*****
int AbreLigacaoServidor()

```

```

    {
        mySerial.print("$$$");
        delay(300);
        mySerial.println("open");
        delay(900);
    }
    /*******

void inicializaLCD()
{
    digitalWrite(ledPin, HIGH);

    if( u8g.getMode() == U8G_MODE_R3G3B2 ){
        u8g.setColorIndex(255);
    }
    else if ( u8g.getMode() == U8G_MODE_GRAY2BIT ){
        u8g.setColorIndex(3);
    }
    else if ( u8g.getMode() == U8G_MODE_BW ){
        u8g.setColorIndex(1);
    }
}
    /*******
void drawModo(void)
{
    switch(estado >> 3/*draw_state >> 3*/) {
        case 0: drawTemp(0); break;
        case 1: drawHum(0); break;
    }
}
    /*******
void drawTemp(uint8_t d)//d
{
    u8g.setFont(u8g_font_gdr10r);
    u8g.drawStr(8+d,10+d,"Temperatura:");
    u8g.setFont(u8g_font_gdr25r);
    dtostrf(t, 1, 2, floatBufferT);
    u8g.drawStr(20+d,45+d,floatBufferT);
}
    /*******
void drawHum(uint8_t d)
{
    u8g.setFont(u8g_font_gdr10r);
    u8g.drawStr(8+d,10+d,"Humidade:");
    u8g.setFont(u8g_font_gdr25r);
    dtostrf(h, 1, 2, floatBufferH);
    u8g.drawStr(20+d,45+d,floatBufferH);
}
    /*******
void actualizaLCD()
{
    u8g.firstPage();

    do {
        drawModo();
    } while( u8g.nextPage() );

    estado++;
    if(estado>= 2*8){
        estado=0;
    }
    // delay(50);
}
    /*******

```

A.2 Configurar módulo Wifly utilizador

```

/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */
package wiflyconfiguration;
import java.awt.event.ItemEvent;
import java.io.*;
import java.net.*;

/**
 *
 * @author Antonio
 */
public class WiflyConfig extends javax.swing.JFrame {

    /**
     * Creates new form WiflyConfig
     */
    public WiflyConfig() {
        initComponents();
        jpInicial.setVisible(true);
        jpConnect.setVisible(false);
        jpModo.setVisible(false);
        jpADHOC.setVisible(false);
        jpInfra.setVisible(false);
    }

    Socket connection;
    BufferedReader in;
    PrintStream out;
    int Selection;
    int Select_Encriptacao;
    ConfigureNetwork cn;
    int Ligar;

    /**
     * This method is called from within the constructor to initialize the form.
     * WARNING: Do NOT modify this code. The content of this method is always
     * regenerated by the Form Editor.
     */
    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated Code">
    private void initComponents() {

        jfSobre = new javax.swing.JFrame();
        jpSobre = new javax.swing.JPanel();
        jlSobre = new javax.swing.JLabel();
        jpInicial = new javax.swing.JPanel();
        jlLabel1 = new javax.swing.JLabel();
        jpConnect = new javax.swing.JPanel();
        jbConnect = new javax.swing.JButton();
        jtfIP = new javax.swing.JTextField();
        jtfPort = new javax.swing.JTextField();
        jlHost = new javax.swing.JLabel();
        jlPort = new javax.swing.JLabel();
        jpModo = new javax.swing.JPanel();
        jlMensagem = new javax.swing.JLabel();
        jcbModo = new javax.swing.JComboBox();
        jlModo = new javax.swing.JLabel();
        jpADHOC = new javax.swing.JPanel();
        jlNomeDispositivoADHOC = new javax.swing.JLabel();
        jtbNomeDispositivoADHOC = new javax.swing.JTextField();
        jtbIPServidorADHOC = new javax.swing.JTextField();
        jtbPortaADHOC = new javax.swing.JTextField();
        jlIPServidorADHOC = new javax.swing.JLabel();

```

```

jlPortaADHOC = new javax.swing.JLabel();
jbGuardarADHOC = new javax.swing.JButton();
jbCancelarADHOC = new javax.swing.JButton();
jpInfra = new javax.swing.JPanel();
jlNomeDispositivo = new javax.swing.JLabel();
jtbNomeDispositivoInfra = new javax.swing.JTextField();
jtbIPServidorInfra = new javax.swing.JTextField();
jtbPortaInfra = new javax.swing.JTextField();
jlIPServidorInfra = new javax.swing.JLabel();
jlPortaInfra = new javax.swing.JLabel();
jtbSSIDInfra = new javax.swing.JTextField();
jtbPasswordInfra = new javax.swing.JTextField();
jlSSIDInfra = new javax.swing.JLabel();
jlPasswordInfra = new javax.swing.JLabel();
jcbEncriptacaoInfra = new javax.swing.JComboBox();
jlEncriptacaoInfra = new javax.swing.JLabel();
jbGuardarInfra = new javax.swing.JButton();
jbCancelarInfra = new javax.swing.JButton();
jmbMenu = new javax.swing.JMenuBar();
jmFicheiro = new javax.swing.JMenu();
jmiConfigurar = new javax.swing.JMenuItem();
jmiSair = new javax.swing.JMenuItem();
jmCreditos = new javax.swing.JMenu();
jmiSobre = new javax.swing.JMenuItem();

jfSobre.setResizable(false);
jfSobre.getContentPane().setLayout(new javax.swing.BoxLayout(jfSobre.getContentPane(),
javax.swing.BoxLayout.LINE_AXIS));

jpSobre.setPreferredSize(new java.awt.Dimension(350, 150));

jlSobre.setFont(new java.awt.Font("Tahoma", 1, 12));
jlSobre.setText("Copyright (c) 2012 Projecto Informática");

javax.swing.GroupLayout jpSobreLayout = new javax.swing.GroupLayout(jpSobre);
jpSobre.setLayout(jpSobreLayout);
jpSobreLayout.setHorizontalGroup(
    jpSobreLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(jpSobreLayout.createSequentialGroup()
            .createSequentialGroup()
                .addGap(51, 51, 51)
                .addComponent(jlSobre)
                .addGap(51, Short.MAX_VALUE)
            )
);
jpSobreLayout.setVerticalGroup(
    jpSobreLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(jpSobreLayout.createSequentialGroup()
            .createSequentialGroup()
                .addGap(59, 59, 59)
                .addComponent(jlSobre)
                .addGap(76, Short.MAX_VALUE)
            )
);

jfSobre.getContentPane().add(jpSobre);

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
setTitle("Configuração Módulo Wifly");
setResizable(false);

jLabel1.setFont(new java.awt.Font("Tahoma", 1, 24));
jLabel1.setText("Configuração Módulo Wifly");

javax.swing.GroupLayout jpInicialLayout = new javax.swing.GroupLayout(jpInicial);
jpInicial.setLayout(jpInicialLayout);
jpInicialLayout.setHorizontalGroup(
    jpInicialLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(jpInicialLayout.createSequentialGroup()
            .createSequentialGroup()
                .addGap(29, 29, 29)
                .addComponent(jLabel1, javax.swing.GroupLayout.PREFERRED_SIZE, 377,
                    javax.swing.GroupLayout.PREFERRED_SIZE)
                .addGap(29, Short.MAX_VALUE)
            )
);

```

```

jpInicialLayout.setVerticalGroup(
    jpInicialLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(jpInicialLayout.createSequentialGroup()
            .addContainerGap()
            .addComponent(jLabel1, javax.swing.GroupLayout.PREFERRED_SIZE, 46,
                javax.swing.GroupLayout.PREFERRED_SIZE)
            .addContainerGap(333, Short.MAX_VALUE))
);

jbConnect.setText("Ligar");
jbConnect.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jbConnectActionPerformed(evt);
    }
});

jtfIP.setText("169.254.1.1");
jtfIP.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jtfIPActionPerformed(evt);
    }
});

jtfPort.setText("2000");

jlHost.setText("IP:");

jlPort.setText("Porta:");

javax.swing.GroupLayout jpConnectLayout = new javax.swing.GroupLayout(jpConnect);
jpConnect.setLayout(jpConnectLayout);
jpConnectLayout.setHorizontalGroup(
    jpConnectLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(jpConnectLayout.createSequentialGroup()
            .addContainerGap(12, true)
            .addComponent(jlHost)
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
            .addComponent(jtfIP, javax.swing.GroupLayout.PREFERRED_SIZE, 120,
                javax.swing.GroupLayout.PREFERRED_SIZE)
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
            .addComponent(jlPort)
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
            .addComponent(jtfPort, javax.swing.GroupLayout.PREFERRED_SIZE,
                javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
            .addComponent(jbConnect, javax.swing.GroupLayout.PREFERRED_SIZE, 77,
                javax.swing.GroupLayout.PREFERRED_SIZE)
            .addContainerGap(99, true))
);
jpConnectLayout.setVerticalGroup(
    jpConnectLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(jpConnectLayout.createSequentialGroup()
            .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, true)
            .addGroup(jpConnectLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                .addComponent(jtfIP, javax.swing.GroupLayout.PREFERRED_SIZE,
                    javax.swing.GroupLayout.DEFAULT_SIZE,
                    javax.swing.GroupLayout.PREFERRED_SIZE)
                .addComponent(jlPort)
                .addComponent(jtfPort, javax.swing.GroupLayout.PREFERRED_SIZE,
                    javax.swing.GroupLayout.DEFAULT_SIZE,
                    javax.swing.GroupLayout.PREFERRED_SIZE)
                .addComponent(jbConnect)
                .addComponent(jlHost))
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
            .addContainerGap(124, true))
);

jcbModo.setModel(new javax.swing.DefaultComboBoxModel(new String[]
{ "(Selecione uma Opção)", "ADHOC", "Infraestrutura" }));
jcbModo.setToolTipText("(Selecione o modo de ligação)");

```



```

jcbModo.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jcbModoActionPerformed(evt);
    }
});

jlModo.setText("Modo:");

javax.swing.GroupLayout jpModoLayout = new javax.swing.GroupLayout(jpModo);
jpModo.setLayout(jpModoLayout);
jpModoLayout.setHorizontalGroup(
    jpModoLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(jpModoLayout.createSequentialGroup()
            .addContainerGap()
            .addGroup(jpModoLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addComponent(jlModo)
                .addGroup(jpModoLayout.createSequentialGroup()
                    .addGap(10, 10, 10)
                    .addComponent(jcbModo, javax.swing.GroupLayout.PREFERRED_SIZE,
                        javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addGap(10, 10, 10)
                    .addComponent(jlMensagem)
                    .addGap(10, 10, 10))
                .addGap(10, 10, 10))
            .addContainerGap(10, true));
jpModoLayout.setVerticalGroup(
    jpModoLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(jpModoLayout.createSequentialGroup()
            .addContainerGap()
            .addGroup(jpModoLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addComponent(jlModo)
                .addGroup(jpModoLayout.createSequentialGroup()
                    .addGap(10, 10, 10)
                    .addComponent(jcbModo, javax.swing.GroupLayout.PREFERRED_SIZE,
                        javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addGap(10, 10, 10)
                    .addComponent(jlMensagem)
                    .addGap(10, 10, 10))
                .addGap(10, 10, 10))
            .addContainerGap(10, true));

jpADHOC.setBorder(javax.swing.BorderFactory.createEtchedBorder());

jlNomeDispositivoADHOC.setText("Nome Dispositivo:");

jlIPServidorADHOC.setText("IP Servidor:");

jlPortaADHOC.setText("Porta:");

jbGuardarADHOC.setText("Guardar");
jbGuardarADHOC.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jbGuardarADHOCActionPerformed(evt);
    }
});

jbCancelarADHOC.setText("Cancelar");
jbCancelarADHOC.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jbCancelarADHOCActionPerformed(evt);
    }
});

javax.swing.GroupLayout jpADHOCLayout = new javax.swing.GroupLayout(jpADHOC);
jpADHOC.setLayout(jpADHOCLayout);
jpADHOCLayout.setHorizontalGroup(
    jpADHOCLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(jpADHOCLayout.createSequentialGroup()
            .addContainerGap()
            .addGroup(jpADHOCLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addComponent(jlNomeDispositivoADHOC)
                .addGroup(jpADHOCLayout.createSequentialGroup()
                    .addGap(10, 10, 10)
                    .addComponent(jlIPServidorADHOC)
                    .addGap(10, 10, 10)
                    .addComponent(jlPortaADHOC)
                    .addGap(10, 10, 10)
                    .addGroup(jpADHOCLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                        .addComponent(jbGuardarADHOC)
                        .addComponent(jbCancelarADHOC)))
                .addGap(10, 10, 10))
            .addContainerGap(10, true));

```

```

javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
    .addComponent(jtbIPServidorADHOC,
        javax.swing.GroupLayout.PREFERRED_SIZE, 120,
        javax.swing.GroupLayout.PREFERRED_SIZE))
    .addGroup(jpADHOCLayout.createSequentialGroup()
        .addComponent(jlNomeDispositivoADHOC)
        .addPreferredGap(javax.swing.LayoutStyle.
            ComponentPlacement.RELATED)
        .addComponent(jtbNomeDispositivoADHOC,
            javax.swing.GroupLayout.PREFERRED_SIZE, 120,
            javax.swing.GroupLayout.PREFERRED_SIZE)))
    .addGap(29, 29, 29)
    .addComponent(jlPortaADHOC)
    .addPreferredGap(javax.swing.LayoutStyle.
        ComponentPlacement.RELATED)
    .addComponent(jtbPortaADHOC, javax.swing.GroupLayout.
        PREFERRED_SIZE, 44, javax.swing.GroupLayout.PREFERRED_SIZE)
    .addGap(0, 76, Short.MAX_VALUE))
    .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
        jpADHOCLayout.createSequentialGroup()
        .addGap(0, 240, Short.MAX_VALUE)
        .addComponent(jbGuardarADHOC)
        .addPreferredGap(javax.swing.LayoutStyle.
            ComponentPlacement.RELATED)
        .addComponent(jbCancelarADHOC))
    .addContainerGap())
);
jpADHOCLayout.setVerticalGroup(
    jpADHOCLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(jpADHOCLayout.createSequentialGroup()
        .addContainerGap()
        .addGroup(jpADHOCLayout.createParallelGroup
            (javax.swing.GroupLayout.Alignment.BASELINE)
        .addComponent(jlNomeDispositivoADHOC)
        .addComponent(jtbNomeDispositivoADHOC,
            javax.swing.GroupLayout.PREFERRED_SIZE,
            javax.swing.GroupLayout.DEFAULT_SIZE,
            javax.swing.GroupLayout.PREFERRED_SIZE))
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
        .addGroup(jpADHOCLayout.createParallelGroup
            (javax.swing.GroupLayout.Alignment.BASELINE)
        .addComponent(jtbIPServidorADHOC,
            javax.swing.GroupLayout.PREFERRED_SIZE,
            javax.swing.GroupLayout.DEFAULT_SIZE,
            javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(jtbPortaADHOC,
            javax.swing.GroupLayout.PREFERRED_SIZE,
            javax.swing.GroupLayout.DEFAULT_SIZE,
            javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(jlIPServidorADHOC)
        .addComponent(jlPortaADHOC)
        .addPreferredGap(javax.swing.LayoutStyle.
            ComponentPlacement.RELATED)
        .addGroup(jpADHOCLayout.createParallelGroup
            (javax.swing.GroupLayout.Alignment.BASELINE)
        .addComponent(jbGuardarADHOC)
        .addComponent(jbCancelarADHOC))
        .addGap(0, 11, Short.MAX_VALUE))
);

jpInfra.setBorder(javax.swing.BorderFactory.createEtchedBorder());

jlNomeDispositivo.setText("Nome Dispositivo:");

jlIPServidorInfra.setText("IP Servidor:");

jlPortaInfra.setText("Porta:");

jlSSIDInfra.setText("SSID:");

```



```

        javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(jtbPasswordInfra,
        javax.swing.GroupLayout.PREFERRED_SIZE, 120,
        javax.swing.GroupLayout.PREFERRED_SIZE)))
    .addPreferredGap(javax.swing.LayoutStyle.
ComponentPlacement.RELATED, 37,
Short.MAX_VALUE)
    .addComponent(jlPortaInfra)
    .addPreferredGap(javax.swing.LayoutStyle.
ComponentPlacement.RELATED)
    .addComponent(jtbPortaInfra,
        javax.swing.GroupLayout.PREFERRED_SIZE, 44,
        javax.swing.GroupLayout.PREFERRED_SIZE)
    .addGap(78, 78, 78))
    .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
jpInfraLayout.createSequentialGroup()
    .addGap(0, 240, Short.MAX_VALUE)
    .addComponent(jbGuardarInfra)
    .addPreferredGap(javax.swing.LayoutStyle.
ComponentPlacement.RELATED)
    .addComponent(jbCancelarInfra)
    .addContainerGap()))
);
jpInfraLayout.setVerticalGroup(
    jpInfraLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(jpInfraLayout.createSequentialGroup()
        .addContainerGap()
        .addGroup(jpInfraLayout.createParallelGroup
        (javax.swing.GroupLayout.Alignment.BASELINE)
            .addComponent(jlNomeDispositivo)
            .addComponent(jtbNomeDispositivoInfra,
                javax.swing.GroupLayout.PREFERRED_SIZE,
                javax.swing.GroupLayout.DEFAULT_SIZE,
                javax.swing.GroupLayout.PREFERRED_SIZE))
        .addPreferredGap(javax.swing.LayoutStyle.
ComponentPlacement.UNRELATED)
        .addGroup(jpInfraLayout.createParallelGroup
        (javax.swing.GroupLayout.Alignment.BASELINE)
            .addComponent(jtbIPServidorInfra,
                javax.swing.GroupLayout.PREFERRED_SIZE,
                javax.swing.GroupLayout.DEFAULT_SIZE,
                javax.swing.GroupLayout.PREFERRED_SIZE)
            .addComponent(jtbPortaInfra, javax.swing.
GroupLayout.PREFERRED_SIZE,
                javax.swing.GroupLayout.DEFAULT_SIZE,
                javax.swing.GroupLayout.PREFERRED_SIZE)
            .addComponent(jlIPServidorInfra)
            .addComponent(jlPortaInfra))
        .addPreferredGap(javax.swing.LayoutStyle.
ComponentPlacement.UNRELATED)
        .addGroup(jpInfraLayout.createParallelGroup
        javax.swing.GroupLayout.Alignment.BASELINE)
            .addComponent(jtbSSIDInfra,
                javax.swing.GroupLayout.PREFERRED_SIZE,
                javax.swing.GroupLayout.DEFAULT_SIZE,
                javax.swing.GroupLayout.PREFERRED_SIZE)
            .addComponent(jlSSIDInfra))
        .addPreferredGap(javax.swing.LayoutStyle.
ComponentPlacement.UNRELATED)
        .addGroup(jpInfraLayout.createParallelGroup
        (javax.swing.GroupLayout.Alignment.BASELINE)
            .addComponent(jtbPasswordInfra,
                javax.swing.GroupLayout.PREFERRED_SIZE,
                javax.swing.GroupLayout.DEFAULT_SIZE,
                javax.swing.GroupLayout.PREFERRED_SIZE)
            .addComponent(jlPasswordInfra))
        .addPreferredGap(javax.swing.LayoutStyle.
ComponentPlacement.UNRELATED)
        .addGroup(jpInfraLayout.createParallelGroup
        (javax.swing.GroupLayout.Alignment.BASELINE)

```

```

        .addComponent(jcbEncriptacaoInfra,
            javax.swing.GroupLayout.PREFERRED_SIZE,
            javax.swing.GroupLayout.DEFAULT_SIZE,
            javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(jlEncriptacaoInfra)
        .addGap(18, 18, 18)
        .addGroup(jpInfraLayout.createParallelGroup
            (javax.swing.GroupLayout.Alignment.BASELINE)
            .addComponent(jbGuardarInfra)
            .addComponent(jbCancelarInfra))
        .addContainerGap(javax.swing.GroupLayout.
            DEFAULT_SIZE, Short.MAX_VALUE)
    );

    jmFicheiro.setText("Ficheiro");

    jmiConfigurar.setText("Configurar Rede");
    jmiConfigurar.addActionListener
    (new java.awt.event.ActionListener() {
        public void actionPerformed
            (java.awt.event.ActionEvent evt) {
            jmiConfigurarActionPerformed(evt);
        }
    });
    jmFicheiro.add(jmiConfigurar);

    jmiSair.setText("Sair");
    jmiSair.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            jmiSairActionPerformed(evt);
        }
    });
    jmFicheiro.add(jmiSair);

    jmbMenu.add(jmFicheiro);

    jmCreditos.setText("Creditos");

    jmiSobre.setText("Sobre");
    jmiSobre.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            jmiSobreActionPerformed(evt);
        }
    });
    jmCreditos.add(jmiSobre);

    jmbMenu.add(jmCreditos);

    setJMenuBar(jmbMenu);

    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(10, 10, 10)
            .addGroup(layout.createParallelGroup
                (javax.swing.GroupLayout.Alignment.LEADING)
                .addComponent(jpConnect,
                    javax.swing.GroupLayout.DEFAULT_SIZE,
                    javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                .addComponent(jpModo,
                    javax.swing.GroupLayout.DEFAULT_SIZE,
                    javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
                    layout.createSequentialGroup()
                        .createSequentialGroup()
                        .addGroup(layout.createParallelGroup
                            (javax.swing.GroupLayout.Alignment.TRAILING)
                            .addComponent(jpInfra,

```

```

        javax.swing.GroupLayout.DEFAULT_SIZE,
        javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
        .addComponent(jpADHOC,
        javax.swing.GroupLayout.DEFAULT_SIZE,
        javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
        .addContainerGap()))
    .addGroup(layout.createParallelGroup
(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(javax.swing.GroupLayout.Alignment.
TRAILING, layout.createSequentialGroup()
            .addContainerGap()
            .addComponent(jpInicial,
                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()
            .addGap(8, 8, 8)
            .addComponent(jpConnect,
                javax.swing.GroupLayout.PREFERRED_SIZE, 34,
                javax.swing.GroupLayout.PREFERRED_SIZE)
            .addGap(7, 7, 7)
            .addComponent(jpModo,
                javax.swing.GroupLayout.PREFERRED_SIZE,
                javax.swing.GroupLayout.DEFAULT_SIZE,
                javax.swing.GroupLayout.PREFERRED_SIZE)
            .addGap(11, 11, 11)
            .addComponent(jpADHOC, javax.swing.GroupLayout.PREFERRED_SIZE,
                javax.swing.GroupLayout.DEFAULT_SIZE,
                javax.swing.GroupLayout.PREFERRED_SIZE)
            .addGap(6, 6, 6)
            .addComponent(jpInfra, javax.swing.GroupLayout.PREFERRED_SIZE,
                javax.swing.GroupLayout.DEFAULT_SIZE,
                javax.swing.GroupLayout.PREFERRED_SIZE)
            .addContainerGap(25, Short.MAX_VALUE))
        .addGroup(layout.createParallelGroup
(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(layout.createSequentialGroup()
                .addContainerGap()
                .addComponent(jpInicial,
                    javax.swing.GroupLayout.PREFERRED_SIZE,
                    javax.swing.GroupLayout.DEFAULT_SIZE,
                    javax.swing.GroupLayout.PREFERRED_SIZE)
                .addContainerGap(27, Short.MAX_VALUE)))
    );

    pack();
} // </editor-fold>
//*****
private void jbCancelarInfraActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    Limpa_Campos_Modo_Infra();
    jpInfra.setVisible(false);
}
//*****
private void jbGuardarInfraActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    try {
        out.println("$$$");
        Thread.sleep(4000);
        //chama a função para inserir os parametros da rede para ligação ao AP
        Modo_Infra();
    } catch (InterruptedException ie) {
    }
}

}
//*****

```

```

private void jcbEncriptacaoInfraActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    Select_Encryptacao = jcbEncriptacaoInfra.getSelectedIndex();
    if (Select_Encryptacao == 0) {
        System.out.println("Aberta");

        jtbPasswordInfra.setEnabled(false);
    } else if (Select_Encryptacao == 1) {
        System.out.println("Wep-128");

    } else if (Select_Encryptacao == 2) {
        System.out.println("Wep-64");
    } else if (Select_Encryptacao == 3) {
        System.out.println("WPA1");
    } else if (Select_Encryptacao == 4) {
        System.out.println("WPA1 & WPA2-PSK");
    } else if (Select_Encryptacao == 5) {
        System.out.println("WPA2-PSK");
    }
}

}
//*****
private void jbCancelarADHOCActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    Limpa_Campos_Modo_ADHOC();
    jpADHOC.setVisible(false);
}

}
//*****
private void jbGuardarADHOCActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    try {
        out.println("$$$");
        Thread.sleep(4000);
        //chama a função para inserir os parametros da rede ADHOC
        Modo_ADHOC();

    } catch (InterruptedException ie) {
    }

}

}
//*****
private void jcbModoActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:

    Selection = jcbModo.getSelectedIndex();
    if (Selection == 1) {
        jpADHOC.setVisible(true);
        jpInfra.setVisible(false);

    } else if (Selection == 2) {
        jpInfra.setVisible(true);
        jpADHOC.setVisible(false);
    }

}

}
//*****
private void jtfIPActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
}

}
//*****
private void jbConnectActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
}

```

```

try {
    if(Ligar ==0){

        String IP = jtfIP.getText();
        int port = Integer.valueOf(jtfPort.getText()).intValue();
        // InetAddress address = InetAddress.getByName(IP);
        connection = new Socket(IP, port);
        jlMensagem.setText("Ligação efectuada com sucesso.");

        out = new PrintStream(connection.getOutputStream(), true);
        in = new BufferedReader(new InputStreamReader
            (connection.getInputStream()));

        cn = new ConfigureNetwork();
        cn.start();
        jbConnect.setText("Desligar");
        Ligar = 1;
    }else {
        connection.close();
        jbConnect.setText("Ligar");
        jlMensagem.setText("Ligação terminada.");
        Ligar=0;
    }

} catch (Exception e) {

    jlMensagem.setText("ERRO: Não foi possível efectuar a ligação.");
}

}

//*****
private void jmiConfigurarActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    jpInicial.setVisible(false);
    jpConnect.setVisible(true);
    jpModo.setVisible(true);
    jpADHOC.setVisible(false);
    jpInfra.setVisible(false);
}

//*****
private void jmiSobreActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    jfSobre.setSize(350, 150);
    jfSobre.setVisible(true);
}

//*****
private void jmiSairActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    System.exit(0);
}

//*****
/**
 * @param args the command line arguments
 */
public static void main(String args[]) {
    /*
     * Set the Nimbus look and feel
     */
    //<editor-fold defaultstate="collapsed" desc="
    " Look and feel setting code (optional) ">
    /*
     * If Nimbus (introduced in Java SE 6) is not available, stay with the
     * default look and feel. For details see
     * http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
     */
    try {
        for (javax.swing.UIManager.LookAndFeelInfo info :

```



```

        javax.swing.UIManager.getInstalledLookAndFeels() {
            if ("Nimbus".equals(info.getName())) {
                javax.swing.UIManager.setLookAndFeel(info.getClassName());
                break;
            }
        }
    } catch (ClassNotFoundException ex) {
        java.util.logging.Logger.getLogger(WiflyConfig.class.getName())
            .log(java.util.logging.Level.SEVERE, null, ex);
    } catch (InstantiationException ex) {
        java.util.logging.Logger.getLogger(WiflyConfig.class.getName())
            .log(java.util.logging.Level.SEVERE, null, ex);
    } catch (IllegalAccessException ex) {
        java.util.logging.Logger.getLogger(WiflyConfig.class.getName())
            .log(java.util.logging.Level.SEVERE, null, ex);
    } catch (javax.swing.UnsupportedLookAndFeelException ex) {
        java.util.logging.Logger.getLogger(WiflyConfig.class.getName())
            .log(java.util.logging.Level.SEVERE, null, ex);
    }
}
//</editor-fold>

/*
 * Create and display the form
 */
java.awt.EventQueue.invokeLater(new Runnable() {

    public void run() {
        new WiflyConfig().setVisible(true);
    }
});
}
// Variables declaration - do not modify
private javax.swing.JLabel jLabel1;
private javax.swing.JButton jbCancelarADHOC;
private javax.swing.JButton jbCancelarInfra;
private javax.swing.JButton jbConnect;
private javax.swing.JButton jbGuardarADHOC;
private javax.swing.JButton jbGuardarInfra;
private javax.swing.JComboBox jcbEncriptacaoInfra;
private javax.swing.JComboBox jcbModo;
private javax.swing.JFrame jfSobre;
private javax.swing.JLabel jlEncriptacaoInfra;
private javax.swing.JLabel jlHost;
private javax.swing.JLabel jlIPServidorADHOC;
private javax.swing.JLabel jlIPServidorInfra;
private javax.swing.JLabel jlMensagem;
private javax.swing.JLabel jlModo;
private javax.swing.JLabel jlNomeDispositivo;
private javax.swing.JLabel jlNomeDispositivoADHOC;
private javax.swing.JLabel jlPasswordInfra;
private javax.swing.JLabel jlPort;
private javax.swing.JLabel jlPortaADHOC;
private javax.swing.JLabel jlPortaInfra;
private javax.swing.JLabel jlSSIDInfra;
private javax.swing.JLabel jlSobre;
private javax.swing.JMenu jmCreditos;
private javax.swing.JMenu jmFicheiro;
private javax.swing.JMenuBar jmbMenu;
private javax.swing.JMenuItem jmiConfigurar;
private javax.swing.JMenuItem jmiSair;
private javax.swing.JMenuItem jmiSobre;
private javax.swing.JPanel jpADHOC;
private javax.swing.JPanel jpConnect;
private javax.swing.JPanel jpInfra;
private javax.swing.JPanel jpInicial;
private javax.swing.JPanel jpModo;
private javax.swing.JPanel jpSobre;
private javax.swing.JTextField jtbIPServidorADHOC;
private javax.swing.JTextField jtbIPServidorInfra;
private javax.swing.JTextField jtbNomeDispositivoADHOC;

```

```

private javax.swing.JTextField jtbNomeDispositivoInfra;
private javax.swing.JTextField jtbPasswordInfra;
private javax.swing.JTextField jtbPortaADHOC;
private javax.swing.JTextField jtbPortaInfra;
private javax.swing.JTextField jtbSSIDInfra;
private javax.swing.JTextField jtfIP;
private javax.swing.JTextField jtfPort;
// End of variables declaration
//*****
//função que vai configurar o Modo ADHOC
public void Modo_ADHOC(){

    if(Selection==1){
        try{
            out.println("set opt deviceid " + jtbNomeDispositivoADHOC.getText());
            Thread.sleep(2000);
            out.println("set ip host " + jtbIPServidorADHOC.getText());
            Thread.sleep(2000);
            out.println("set ip remote " + jtbPortaADHOC.getText());
            Thread.sleep(2000);
            out.println("save");
            Thread.sleep(2000);
            out.println("reboot");

        }catch(InterruptedException iee){}
    }
}
//*****
//função que vai configurar a ligação ao AP
public void Modo_Infra(){
    if(Selection==2){
        try{
            out.println("set opt deviceid " + jtbNomeDispositivoInfra.getText());
            Thread.sleep(2000);
            out.println("set ip host " + jtbIPServidorInfra.getText());
            Thread.sleep(2000);
            out.println("set ip remote " + jtbPortaInfra.getText());
            Thread.sleep(2000);
            out.println("set wlan ssid " + jtbSSIDInfra.getText());
            Thread.sleep(2000);
            Encriptacao();
            Thread.sleep(2000);
            out.println("set sys mask 0x21f0");
            Thread.sleep(1000);
            out.println("set sys output 0x0100 0x0100");
            Thread.sleep(5000);
            out.println("set sys output 0x0000 0x0100");
            Thread.sleep(1000);
            out.println("save ");
            Thread.sleep(2000);
            out.println("reboot");

        }catch(InterruptedException ie){}
    }
}
//*****
public void Encriptacao(){

    try{
        if(Select_Encriptacao==0){
            out.println("set wlan auth 0");

        }else if(Select_Encriptacao==1){
            out.println("set wlan auth 1");
            Thread.sleep(2000);
            out.println("set wlan key " + jtbPasswordInfra.getText());
        }else if(Select_Encriptacao==2){
            out.println("set wlan auth 8");
            Thread.sleep(2000);
            out.println("set wlan key " + jtbPasswordInfra.getText());
        }
    }
}

```

```

    }else if(Select_Encriptacao==3){
        out.println("set wlan auth 2");
        Thread.sleep(2000);
        out.println("set wlan phrase " + jtbPasswordInfra.getText());
    }else if(Select_Encriptacao==4){
        out.println("set wlan auth 3");
        Thread.sleep(2000);
        out.println("set wlan phrase " + jtbPasswordInfra.getText());
    }else if(Select_Encriptacao==5){
        out.println("set wlan auth 4");
        Thread.sleep(2000);
        out.println("set wlan phrase " + jtbPasswordInfra.getText());
    }
} catch (InterruptedException ie){}
}
}
//*****
public void Limpa_Campos_Modo_ADHOC(){

    jtbNomeDispositivoADHOC.setText("");
    jtbIPServidorADHOC.setText("");
    jtbPortaADHOC.setText("");

}

//*****
public void Limpa_Campos_Modo_Infra(){

    jtbNomeDispositivoInfra.setText("");
    jtbIPServidorInfra.setText("");
    jtbPortaInfra.setText("");
    jtbSSIDInfra.setText("");
    jtbPasswordInfra.setText("");

}

//*****
public class ConfigureNetwork extends Thread {

    public void run() {

        try {
            while (true) {
                System.out.println("/*jtaReceber.getText() */" + "\n" + in.readLine());
            }
        } catch (IOException e) {}

    }

}
}
//*****

```

A.3 Programa configurar RTC

```

#include <Wire.h>
#include "RTClib.h"

RTC_DS1307 RTC;

void setup () {
    Serial.begin(57600);
    Wire.begin();
    RTC.begin();

    if (! RTC.isrunning()) {
        Serial.println("RTC is NOT running!");
        // RTC.adjust(DateTime(__DATE__, __TIME__));
        RTC.adjust(DateTime("AUG 12 2012", "12:54:56"));
    }
}

```

```

}

void loop () {

    DateTime now = RTC.now();

    Serial.print(now.year(), DEC);
    Serial.print('/');
    Serial.print(now.month(), DEC);
    Serial.print('/');
    Serial.print(now.day(), DEC);
    Serial.print(' ');
    Serial.print(now.hour(), DEC);
    Serial.print(':');
    Serial.print(now.minute(), DEC);
    Serial.print(':');
    Serial.print(now.second(), DEC);
    Serial.println();
}

```

A.4 Aplicação Servidor

```

/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */

import java.io.*;
import java.net.*;
import java.sql.*;
import java.text.*;
import java.util.Scanner.*;
import java.util.*;

/**
 *
 * @author Antonio
 */
//*****
public class Servidor extends Thread {

    /**
     * @param args the command line arguments
     */

    public static void main(String[] args) {
        // TODO code application logic here
        int porta=8081;

        try{
            ServerSocket s = new ServerSocket(porta);//porta onde vai estar à escuta
            while(true){
                System.out.print("Servidor na porta:"+porta +" ->
                A espera que alguém se conecte...");
                //controi uma tread para se comunicar com o cliente
                Socket Cliente = s.accept();
                Thread tc = new Servidor(Cliente);
                tc.start();

                InetAddress inetAddress = Cliente.getInetAddress();
                System.out.println("Ligação aceite! ->" +inetAddress);
            }
        }catch(IOException e){
            System.out.println("IOException: " + e);
        }
    }
}

```

```

    }
}
String linha2;
String linha;
BufferedReader entrada = null;
String macAddr;
String valor;
String TipoSensor;
String Sensor;
String Data;

//*****
public void run() {

    try {
        //comunicação com o cliente
        entrada = new BufferedReader(new InputStreamReader
        (Cliente.getInputStream()));
        //saida = new PrintWriter(Cliente.getOutputStream(),true);
        //para enviar msg para o cliente
        //-----
        String EnvMAC = entrada.readLine();
        Scanner scanEnvMAC = new Scanner(EnvMAC);

        if(EnvMAC.contains("MAC:")){
            scanEnvMAC.useDelimiter("MAC:");
            macAddr=scanEnvMAC.next();
            if(macAddr!=null){
                try{

                    System.out.println("MAC:"+macAddr);
                    LigarBD();
                    ResultSet LerDispositivo = statement.executeQuery
                    ("Select *from Dispositivo "
                     + "WHERE MAC='"+macAddr+"' ");

                    if(LerDispositivo.isNull()){
                        System.out.println("Dispositivo nao existe");

                    }else{
                        System.out.println("Dispositivo ja existe");
                    }
                }catch(Exception e){System.out.println("ERRO");}
            }else{
                System.out.println("ERRO");
            }
        }

        //-----
        linha = EnvMAC;

        while (linha != null ) {
            linha = entrada.readLine();
            // espera que alguém escreva, senão anda sempre as voltas
            Scanner scanner = new Scanner(linha);
            linha2=linha;

            if(linha.contains("MAC:")){
                scanner.useDelimiter("MAC:");
                macAddr=scanner.next();
                System.out.println("o mac address recebido é:"+macAddr);
                if(macAddr!=null){
                    try{
                        System.out.println("MAC:"+macAddr);
                        LigarBD();
                        ResultSet LerDispositivo = statement.executeQuery
                        ("Select *from Dispositivo "
                         + "WHERE MAC='"+macAddr+"' ");

                        if(LerDispositivo.isNull()){
                            System.out.println("Dispositivo nao existe");

```

```

        }else{
            System.out.println("Dispositivo ja existe");
        }
        }catch(Exception e){System.out.println("ERRO");}
    }else{
        System.out.println("ERRO");
    }
}
else if(linha.contains("Data:")){
    scanner.useDelimiter("Data:");
    Data = scanner.next();
    // System.out.println("Data:" +Data);
}

else if(linha.contains("S")){
    scanner.useDelimiter("S");// dá o tipo de sensor

    Sensor = scanner.next();

    Scanner scanTipoSensor = new Scanner(Sensor);
    if(Sensor.contains(":")){
        scanTipoSensor.useDelimiter(":");
        TipoSensor =scanTipoSensor.next();
    }
    Scanner scanValor = new Scanner(Sensor);
    if(linha2.contains(":"))
    {
        scanValor.findInLine(":");
        valor= scanValor.next();
    }

    System.out.println("Data:" +Data);
    // System.out.println("Sensor:" +Sensor);
    System.out.println("TipoSensor recebido:" +TipoSensor);
    System.out.println("Valor recebido:" +valor);
    InserirDadosTBdados();
    Cliente.close();
}
// InserirDadosTBdados();

}

}catch(IOException e){
    //caso ocorra alguma excessao de E/S, mostre qual foi.
    System.out.println("IOException: " + e);
}
}

private Socket Cliente;
//*****
public Servidor(Socket s){
    Cliente=s;
}
Connection connection;
Statement statement;
//*****
public void LigarBD(){
    try{
        Class.forName("com.mysql.jdbc.Driver");
        connection = DriverManager.getConnection
            ("jdbc:mysql://localhost:63306/dbase","root","");
        statement = connection.createStatement();

    }catch(Exception e){
    }
}
}

```

```

//*****
public void InserirDadosTBdados(){
    try {
        LigarBD();
        //faz o insert na tabela dados da base de dados
        String sensor = null;
        String IdTipoSensor = null;
        ResultSet LerIdTipoSensor = statement.executeQuery
        ("SELECT ID_TIPOSENSOR FROM TIPOSENSOR WHERE "
         + "DESCRICAONSENSOR='"+TipoSensor+"'");

        while(LerIdTipoSensor.next()){
            IdTipoSensor = LerIdTipoSensor.getString("ID_TIPOSENSOR");
            System.out.println("O id do Tipo Sensor:" +IdTipoSensor);
        }
        ResultSet LerIdSensor = statement.executeQuery
        ("SELECT ID_SENSOR FROM SENSOR,DISPOSITIVO WHERE "
         + "SENSOR.ID_DISPOSITIVO= DISPOSITIVO.ID_DISPOSITIVO AND
         (SENSOR.TIPOSENSOR='"+IdTipoSensor+"' )"
         + "AND DISPOSITIVO.MAC='"+macAddr+"'");

        while(LerIdSensor.next()){
            sensor = LerIdSensor.getString("ID_SENSOR");
            System.out.println("o ID do Sensor é:"+sensor);
        }
        statement.executeUpdate("INSERT INTO DADOS (VALORREGISTO,ID_SENSOR,DATA) "
         + "VALUES('"+valor+"', '"+sensor+"', '"+Data+"')");
        System.out.println("Foi inserido um registro...");
    }catch(Exception e){
        System.out.println("erro ao inserir dado");
    }
}
}
//*****

```

A.5 WEB page index

```

<%--
Document    : index
Created on  : 5/Jul/2012, 9:51:07
Author      : Antonio
--%>

<%@page contentType="text/html" pageEncoding="UTF-8" import="java.sql.*"%>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">

<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
<meta name="keywords" content="">
<meta name="description" content="">
<meta charset="utf-8" />
<title>Remote Control</title>
<link rel="stylesheet" type="text/css" href="reset-fonts-grids.css">
<link rel="stylesheet" type="text/css" href="estilo1.css">

<style type="text/css">
<!--
.style1 {
font-size: 24px;
color: #000000;

```

```

}
.style2 {font-size: 36px}
-->
</style>
</head>

<body>

    <form name="form1" method="post" action="valida.jsp">
        <div id="geral">

            <div style="float: left; WIDTH: 49%">
                <SPAN style="position: absolute; top: 130px;
                left: 30px; width: 250px; height: 90px">
                    <p align="center" class="style2">
                        Aquisi&ccedil;&atilde;o dados</p>
                    <p align="center" class="style2">remotamente </p>
                </SPAN>
            </div>
            <div style="float: right; WIDTH: 49%">
                <p>&nbsp;</p>
                <p>&nbsp;</p>
                <p class="style1">Iniciar sess&atilde;o</p>
                <p>&nbsp;</p>
                <p>&nbsp;</p>
                <p>Utilizador: </p>
                <p>
                    <label>
                        <input name="txtUserName" type="text" id="txtUserName">
                    </label>
                </p>
                <p>Palavra-passe:</p>
                <p>
                    <label>
                        <input name="txtPassword" type="password" id="txtPassword">
                    </label>
                </p>
                <p>&nbsp;</p>
                <p>
                    <label>
                        <input type="submit" name="Submit" value="Iniciar sess&atilde;o">
                    </label>
                </p>
                <p>&nbsp;</p>
                <p>&nbsp;</p>
                <p>&nbsp;</p>
                <p>&nbsp;</p>
                <p>&nbsp;</p>
            </div>
            <a href="addcookie.jsp"></a>

        </form>
        <%
            String mensagem = (String) session.getAttribute("msg");
            if (mensagem == null){
                mensagem = "";
            }else{
                mensagem = mensagem;
            }
            out.println(mensagem);
            %>
            <h1>&nbsp;</h1>

            <h1>&nbsp;</h1>
        </body>
</html>

```


A.6 WEB page valida Utilizador

```

<!--
  Document   : valida
  Created on : 28/Mai/2012, 15:29:34
  Author    : Antonio
-->

<%@page contentType="text/html" pageEncoding="UTF-8" import="java.sql.*"%>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
    "http://www.w3.org/TR/html4/loose.dtd">

<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <title>JSP Page</title>
  </head>
  <body>
    <h1>Página validação</h1>

    <%
      Connection con;
      Statement stmt;

      try {
        Class.forName("com.mysql.jdbc.Driver");
      } catch (java.lang.ClassNotFoundException e) { %>
        <h1>ClassNotFoundException: <%=e.getMessage()%></h1> <%
        }%>

      <%
      try {
        con = DriverManager.getConnection
          ("jdbc:mysql://localhost:63306/dbase","root","");
        stmt = con.createStatement();

//ResultSet rs = stmt.executeQuery ("SELECT * FROM user");

        if(request.getParameter("txtUserName")!=null &&
          request.getParameter("txtUserName")!=" " &&
          request.getParameter("txtPassword")!=null &&
          request.getParameter("txtPassword")!=""){

          String username = request.getParameter("txtUserName").toString();
          String userpass = request.getParameter("txtPassword").toString();

// ResultSet rs = stmt.executeQuery("SELECT password
FROM user WHERE ((userName)=
'+request.getParameter("txtUserName")+' );");
ResultSet rs = stmt.executeQuery("SELECT perfil,
password FROM user WHERE ((userName)=
'+request.getParameter("txtUserName")+' ) AND
((password)='+request.getParameter("txtPassword")+')'););");

          if(rs.next()!= true){
            String msg = "<h1 style=\"color:red;text-align:center;\">
            Essa palavra-passe está incorreta. Tente novamente </h1>";
            session.setAttribute("msg", msg);
            response.sendRedirect("index.jsp");
            // response.sendRedirect("index.jsp");
          }
        }
      }
    %>
  </body>
</html>

```

```

    }
    else
    {
        int op = Integer.parseInt((rs.getString("perfil")));
        // String perfilUtilizador = rs.getString("perfil").toString();
        out.println("perfil " + op );

        switch (op){
            case 1:
                response.sendRedirect("aInicio.jsp");
                break;
            case 2:
                response.sendRedirect("sInicio.jsp");
                break;
            case 3:
                response.sendRedirect("oInicio.jsp");
            default:
                System.out.println("Erro");
        }
    }

}

else
{
    response.sendRedirect("index.jsp");
}

stmt.close();
con.close();
} catch(SQLException ex) {
%>
<h1>SQLException: <%=ex.getMessage()%></h1>
<% } %>
</body>
</html>

```

A.7 WEB page Adiciona Dispositivo

```

<%--
    Document    : AdicionaDispositivo
    Created on  : 5/Jul/2012, 16:23:37
    Author      : Antonio
--%>

<%@page contentType="text/html" pageEncoding="UTF-8" import="java.sql.*"%>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
    "http://www.w3.org/TR/html4/loose.dtd">

<html>
<head>
    <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
<meta name="keywords" content="">
<meta name="description" content="">

<title>CSS Layout</title>
<link rel="stylesheet" type="text/css" href="reset-fonts-grids.css">
<link rel="stylesheet" type="text/css" href="estilo.css">
    <script type="text/javascript" src="script2.js">
</script>
    <style type="text/css">
<!--

```

```

.style1 {
font-size: 24px;
color: #000000;
}
.style2 {font-size: 36px}
-->
</style>
</head>

<body>
<h1>Página validação</h1>

<%
    Connection con;
    Statement stmt;

    try {
        Class.forName("com.mysql.jdbc.Driver");
    } catch(java.lang.ClassNotFoundException e) { %>
        <h1>ClassNotFoundException: <%=e.getMessage()%></h1> <%
        }%>

    <%
        try {
            con = DriverManager.getConnection
                ("jdbc:mysql://localhost:63306/dbase","root","");
            stmt = con.createStatement();

            String dataInstal = request.getParameter("txtDataInstalacao").toString();
            String macAddress = request.getParameter("txtMac").toString();
            String nomeDispositivo = request.getParameter("txtNomeDisposit").toString();
            String localInstalacao = request.getParameter("txtLocalInstal").toString();
            String ipAddress = request.getParameter("txtIp").toString();
            String porta = request.getParameter("txtPorta").toString();
            String ssid = request.getParameter("txtSSID").toString();
            String password = request.getParameter("txtPassword").toString();

            String estado = request.getParameter("txtEstado").toString();

%>

    <div id= "geral">

        <div id= "topo">
topo
        </div>
        <div id= "menuNavegacao">
            <ul id="nav">
                <li><a href="/">Inicio</a></li>
            <li><a href="/about/">Dispositivo</a>
                <ul>
                    <li><a href="sAdicionarDispositivo.jsp">Adicionar</a></li>
                    <li><a href="sConsultarDispositivo.jsp">Consultar</a></li>
                </ul>
            </li>
            <li>
                <a href="/portfolio/">Configurar</a>
                <ul>
                    <li><a href="#">Sensores</a></li>
                </ul>
            </li>
            <li>

```

```

<a href="/portfolio/">Conta</a>
    <ul>
<li><a href="#">Adicionar</a></li>
<li>
    <div align="right"><a href="#">Consultar</a>

        </div>
</li>
</ul>
</li>

<li><a href="sAjuda.jsp">Ajuda</a></li>
<li><a href="logOut.jsp">Sair</a></li>
</ul>
</div>

        <div id= "conteudoGeral">
        <%

String SQL = "Insert into dispositivo(dataInstalacao,
localInstalacao,nomeDispositivo,
ipAddress,portaDispositivo,mac,password,ssid,estado)
VALUES('" + dataInstal + "',''" + localInstalacao + "','
'" + nomeDispositivo + "',''" + ipAddress + "',''" + porta + "','
'" + macAddress + "',''" + password + "',''" + ssid + "',''" + estado + "',''" );

        if( stmt.executeUpdate(SQL)>0);{%>
        <h1 style="color: green; text-align: center" >
        Dispositivo inserido com sucesso </h1>

        <%>

%>

        </div>

<div id= "rodape">
    <ul id="nav">
<li><a href="/">Inicio</a> </li>
<li><a href="/">Ajuda</a></li>
<li><a href="logOut.jsp">Sair</a></li>
</ul>
rodape
</div>
</div>

        <% stmt.close();
            con.close();
            } catch(SQLException ex) {
                <%
                <h1>SQLException: <%=ex.getMessage()%></h1>
                <% } %>
        </body>
</html>

```

A.8 WEB page Consultar Dispositivo

```

<!--
Document   : sConsultarDispositivo
Created on : 5/Jul/2012, 16:23:37
Author    : Antonio
-->

<%@page contentType="text/html" pageEncoding="UTF-8" import="java.sql.*"%>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">

```

```

<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
    <meta name="keywords" content="">
    <meta name="description" content="">

    <title>Remote Control</title>
    <link rel="stylesheet" type="text/css" href="reset-fonts-grids.css">
    <link rel="stylesheet" type="text/css" href="estilo.css">
    <script type="text/javascript" src="script2.js">
  </script>
    <style type="text/css">
  <!--
  .style1 {
  font-size: 24px;
  color: #000000;
  }
  .style2 {font-size: 36px}
  -->
  </style>
  </head>

  <body>

    <%
      Connection con;
      Statement stmt;

      try {
        Class.forName("com.mysql.jdbc.Driver");
      } catch(java.lang.ClassNotFoundException e) { %>
        <h1>ClassNotFoundException: <%=e.getMessage()%></h1> <%
        }%>

      <%
      try {
        con = DriverManager.getConnection("jdbc:mysql://localhost:63306/dbase","root","");
        stmt = con.createStatement(); %>

    <div id= "geral">

      <div id= "topo">
        <tr align="center"><td> <h1 style="color:#0000FF; text-align: Center" >
          Aquisição dados Remotamente </h1></td>

      </div>
      <div id= "menuNavegacao">
        <ul id="nav">
          <li><a href="sInicio.jsp">Inicio</a></li>
        <li><a href="/about/">Dispositivo</a>
          <ul>
            <li><a href="sAdicionarDispositivo.jsp">Adicionar</a></li>
            <li><a href="sConsultarDispositivo.jsp">Consultar</a></li>
          </ul>

```

```

</li>
<li>
<a href="/portfolio/">Configurar</a>
    <ul>
        <li><a href="testeSensor.jsp">Sensores</a></li>
    </ul>
</li>
<li>
<a href="/portfolio/">Conta</a>
    <ul>
<li><a href="sAdicionarConta.jsp">Adicionar</a></li>
<li>
    <div align="right"><a href="sConsultarUtilizador.jsp">Consultar</a>

        </div>
</li>
</ul>
</li>

<li><a href="sAjuda.jsp">Ajuda</a></li>
<li><a href="logOut.jsp">Sair</a></li>
</ul>
</div>

    <div id= "conteudoGeral">
        <% String dispositivoRecebido = request.getParameter("dispositivo");
        // out.println("Dispositivo " + dispositivoRecebido);
        if(request.getParameter("dispositivo")!= null){ %>

<table class="gridtable">
<tr>
    <th width="5%"> ID</th>
    <th width="15%">Mac</th>
    <th width="15%">Local Instalação</th>
    <th width="15%">Data Instalação</th>
    <th width="10%">Porta</th>
    <th width="10%">SSID</th>
    <th width="10%">Password</th>
    <th width="5%">Estado</th>
    <th width="5%">&nbsp;</th>
    <th width="5%">&nbsp;</th>
</tr>

<%
ResultSet rs = stmt.executeQuery("SELECT id_dispositivo,
mac,localInstalacao,dataInstalacao,portaDispositivo,
ssid,password,estado
FROM dispositivo Where dispositivo.mac =
'" +request.getParameter("dispositivo")+"'");
while (rs.next()){
%>
<tr>

    <th width="5%"> <%=rs.getString("id_dispositivo")%></th>
    <th width="15%"><%=rs.getString("mac")%></th>
    <th width="15%"><%=rs.getString("localInstalacao")%></th>
    <th width="15%"><%=rs.getString("dataInstalacao")%></th>
    <th width="10%"><%=rs.getString("portaDispositivo")%></th>
    <th width="10%"><%=rs.getString("ssid")%></th>
    <th width="10%"><%=rs.getString("password")%></th>
    <th width="5%"><%=rs.getString("estado")%></th>
    <% String iddispositivo = rs.getString("id_dispositivo");%>
    <td><div align="center"><a href="sAlterarDispositivo.jsp?
id=<%=iddispositivo%>&op=uc">...</a></div></td>
    <td><div align="center"><a href="sConfigurarDispositivo.jsp?
id=<%=iddispositivo%>&op=dc">x</a></div></td>
</tr>

```

```

        <% }
    %>
    <%}

        else {%>

<table class="gridtable">
    <tr>
        <th width="5%"> ID</th>
        <th width="15%">Mac</th>
        <th width="15%">Local Instalação</th>
        <th width="15%">Data Instalação</th>
        <th width="10%">Porta</th>
        <th width="10%">SSID</th>
        <th width="10%">Password</th>
        <th width="5%">Estado</th>
        <th width="5%">&nbsp;</th>
        <th width="5%">&nbsp;</th>
    </tr>
<%

    ResultSet rs = stmt.executeQuery("SELECT id_dispositivo,mac,
    localInstalacao,dataInstalacao,portaDispositivo,ssid,
    password,estado FROM dispositivo;");
    while (rs.next()){
    %>

        <tr>

            <th width="5%"> <%=rs.getString("id_dispositivo")%></th>
            <th width="15%"><%=rs.getString("mac")%></th>
            <th width="15%"><%=rs.getString("localInstalacao")%></th>
            <th width="15%"><%=rs.getString("dataInstalacao")%></th>
            <th width="10%"><%=rs.getString("portaDispositivo")%></th>
            <th width="10%"><%=rs.getString("ssid")%></th>
            <th width="10%"><%=rs.getString("password")%></th>
            <th width="5%"><%=rs.getString("estado")%></th>
            <% String iddispositivo = rs.getString("id_dispositivo");%>
            <td><div align="center"><a href="sAlterarDispositivo.jsp?
            id=<%=iddispositivo%>&op=uc">...</a></div></td>
            <td><div align="center"><a href="sConfigurarDispositivo.jsp?
            id=<%=iddispositivo%>&op=dc">x</a></div></td>

        </tr>
        <% }
    %>

<% }%>

    <form name="form1" method="post" action=
    "sConsultarDispositivo.jsp">
        Dispositivo:

        <select name="dispositivo">
            <%ResultSet rs = stmt.executeQuery(" SELECT *
            FROM dispositivo");
            while (rs.next()){
                String localizacao=rs.getString("mac");
                %> <option value= "<%=localizacao%>" >
                <% out.println(localizacao); %></option> <%
            } %>
        </select>
        <input type="submit" value="Procurar"><br><br>
    </form>
</table> </table>
    </div>

```

```

</div>

<% stmt.close();
        con.close();
        } catch(SQLException ex) {
            %>
            <h1>SQLException: <%=ex.getMessage()%></h1>
            <% } %>
    </body>
</html>

```

A.9 WEB page altera Dispositivo

```

<!--
    Document    : AdicionaDispositivo
    Created on  : 5/Jul/2012, 16:23:37
    Author      : Antonio
-->

<%@page contentType="text/html" pageEncoding="UTF-8" import="java.sql.*"%>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
    "http://www.w3.org/TR/html4/loose.dtd">

<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
        <meta name="keywords" content="">
        <meta name="description" content="">

        <title>CSS Layout</title>
        <link rel="stylesheet" type="text/css" href="reset-fonts-grids.css">
        <link rel="stylesheet" type="text/css" href="estilo.css">
        <script type="text/javascript" src="script2.js">
        </script>
        <style type="text/css">
        <!--
        .style1 {
        font-size: 24px;
        color: #000000;
        }
        .style2 {font-size: 36px}
        -->
        </style>
    </head>

    <body>

        <%
            Connection con;
            Statement stmt;

            try {
                Class.forName("com.mysql.jdbc.Driver");
            } catch(java.lang.ClassNotFoundException e) { %>
                <h1>ClassNotFoundException: <%=e.getMessage()%></h1> <%
                }%>

```



```

<%
try {
con = DriverManager.getConnection
("jdbc:mysql://localhost:63306/dbase","root","");
stmt = con.createStatement();

String id = request.getParameter("id").toString();
String dataInstalacao = request.getParameter
("txtDataInstalacao").toString();
String macAddress =
request.getParameter("txtMac").toString();
String nomeDispositivo =
request.getParameter("txtNomeDisposit").toString();
String localInstalacao =
request.getParameter("txtLocalInstal").toString();
String ipAddress =
request.getParameter("txtIp").toString();
String porta =
request.getParameter("txtPorta").toString();
String ssid =
request.getParameter("txtSSID").toString();
String password =
request.getParameter("txtPassword").toString();

String estado =
request.getParameter("txtEstado").toString();

%>

<div id= "geral">

<div id= "topo">
<tr align="center"><td> <h1 style="color:#0000FF; text-align: Center" >
Aquisição dados Remotamente </h1></td>

</div>
<div id= "menuNavegacao">
<ul id="nav">
<li><a href="sInicio.jsp">Inicio</a></li>
<li>
<a href="">Dispositivo</a>
<ul>
<li><a href="sAdicionarDispositivo.jsp">Adicionar</a></li>
<li><a href="sConsultarDispositivo.jsp">Consultar</a></li>
</ul>
</li>
<li>
<a href="">Configurar</a>
<ul>
<li><a href="testeSensor.jsp">Sensores</a></li>
</ul>
</li>
<li>
<a href="/portfolio/">Conta</a>
<ul>
<li><a href="sAdicionarConta.jsp">Adicionar</a></li>
<li>
<div align="right"><a
href="sConsultarUtilizador.jsp">Consultar</a>

</div>
</li>
</ul>
</li>
<li><a href="sAjuda.jsp">Ajuda</a></li>
<li><a href="logOut.jsp">Sair</a></li>
</ul>
</div>

```

```

        <div id= "conteudoGeral">

        <% String dispositivoRecebido =
        request.getParameter("id");
        // out.println("Dispositivo " + dispositivoRecebido);
        if(request.getParameter("id")!= null){

                String SQL = "UPDATE dispositivo
                SET id_dispositivo='"+ id +"',
                dataInstalacao='\" + dataInstalacao + '\",
                localInstalacao='
                \" + localInstalacao + '\",
                nomeDispositivo='\" + nomeDispositivo + '\",
                ipAddress='\" + ipAddress + '\",
                portaDispositivo='\" + porta + '\",
                mac='\" + macAddress + '\",
                password='\" + password + '\",ssid='\" + ssid + '\",
                estado='\" + estado + '\" WHERE
                id_dispositivo='\"+ id +\"\"';
                stmt.executeUpdate(SQL);

                if( stmt.executeUpdate(SQL)>0);{%>
                <h1 style="color: green; text-align: center" >
                Dispositivo Alterado com sucesso </h1>
        <%}

                }

        %>

        </div>

</div>

        <% stmt.close();
        con.close();
        } catch(SQLException ex) {
        %>
        <h1>SQLException: <%=ex.getMessage()%></h1>
        <% } %>

</body>
</html>

```

A.10 WEB page alterar Dispositivo

```

<%--
Document   : sAlterarDispositivo
Created on : 21/Jul/2012, 17:59:37
Author    : Antonio
--%>

<%@page contentType="text/html" pageEncoding="UTF-8" import="java.sql.*"%>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">

<html>
<head>
    <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
<meta name="keywords" content="">
<meta name="description" content="">

<title>CSS Layout</title>
<link rel="stylesheet" type="text/css" href="reset-fonts-grids.css">

```

```

<link rel="stylesheet" type="text/css" href="estilo.css">
  <script type="text/javascript" src="script2.js">
</script>
  <style type="text/css">
<!--
.style1 {
font-size: 24px;
color: #000000;
}
.style2 {font-size: 36px}
-->
  </style>
</head>

<body>
  <h1>Página validação</h1>

  <%
    Connection con;
    Statement stmt;

    try {
      Class.forName("com.mysql.jdbc.Driver");
    } catch(java.lang.ClassNotFoundException e) { %>
      <h1>ClassNotFoundException: <%=e.getMessage()%></h1> <%
    }%>

    <%
    try {
      con = DriverManager.getConnection("jdbc:mysql://localhost:63306/dbase","root","");
      stmt = con.createStatement();
      String SQL = "";
    %>

    <div id= "geral">

      <div id= "topo">
Alterar dispositivo
      </div>
      <div id= "menuNavegacao">
        <ul id="nav">
          <li><a href="sInicio.jsp">Inicio</a></li>
</li>
<a href="/about/">Dispositivo</a>
          <ul>
</li><a href="sAdicionarDispositivo.jsp">Adicionar</a></li>
<li><a href="sConsultarDispositivo.jsp">Consultar</a></li>
          </ul>
</li>
<li>
<a href="/portfolio/">Configurar</a>
          <ul>
            <li><a href="testeSensor.jsp">Sensores</a></li>
          </ul>
</li>
<li>
<a href="/portfolio/">Conta</a>
          <ul>
</li><a href="sAdicionarConta.jsp">Adicionar</a></li>
<li>
          <div align="right"><a href="#">Consultar</a>
        </div>
        </div>

```

```

</li>
</ul>
</li>

```

```

<li><a href="sAjuda.jsp">Ajuda</a></li>
<li><a href="logOut.jsp">Sair</a></li>
</ul>
</div>

```

```

<div id= "conteudoGeral">
<form name="form1" method="post" action="sAlterarDispositivo.jsp">
<table border="0" width=100% align="center">
<tr align="center"><td> <h1 style="color: blue;
text-align: left" >
Alterar dispositivo </h1></td>
</tr>
</table>
<br><br>
<%
if(request.getParameter("id")!= null){
ResultSet rs = stmt.executeQuery("SELECT id_dispositivo,mac,
nomeDispositivo,ipAddress,
localInstalacao,dataInstalacao,
portaDispositivo
,ssid,password,estado FROM dispositivo Where id_dispositivo =
'+request.getParameter("id")+'");
while (rs.next()){
%>

<table border="0" width="367" >

<tr><td width="159">ID:</td> <td width="192"><label
for="txtID"></label>
<input type="text" name="id" id="txtid" value="<%=
rs.getString("id_dispositivo")%>"></td></tr>

<tr><td>Data de instalação:</td> <td><label
for="txtDataInstalacao"></label>
<input name="txtDataInstalacao" type="text"
id="txtDataInstalacao"
value="<%=rs.getString("dataInstalacao")%>">

<tr><td>Mac address:</td><td><label for="txtMac"></label>
<input type="text" name="txtMac" id="txtMac"
value="<%=rs.getString("mac")%>"></td></tr>
<tr><td>Nome Dispositivo:</td><td><label
for="txtNomeDisposit"></label>
<input type="text" name="txtNomeDisposit"
id="txtNomeDisposit"
value="<%=rs.getString("nomeDispositivo")%>"></td></tr>
<tr><td>Local de instalação:</td> <td><label
for="txtLocalInstal"></label>
<input type="text"
name="txtLocalInstal" id="txtLocalInstal"
value="<%=rs.getString("localInstalacao")%>"></td></tr>
<tr ><td>Ip:</td> <td><label for="txtIp"></label>
<input type="text" name="txtIp" id="txtIp"
value="<%=rs.getString("ipAddress")%>"></td></tr>
<tr><td>Porta:</td> <td><label for="txtPorta"></label>
<input type="text" name="txtPorta" id="txtPorta"
value="<%=rs.getString("portaDispositivo")%>"></td></tr>
<tr><td>SSID:</td> <td><label for="txtSSID"></label>
<input type="text" name="txtSSID" id="txtSSID"
value="<%=rs.getString("ssid")%>"></td></tr>
<tr><td>Password:</td>
<td><label for="txtPassword"></label>

```

```

        <input type="text" name="txtPassword" id="txtPassword"
        value="<%=rs.getString("password")%>"></td></tr>
        <tr><td>Estado:</td> <td><label for="txtEstado"></label>
        <input type="text" name="txtEstado" id="txtEstado"
        value="<%=rs.getString("estado")%>"></td></tr>

        </table>

    <% } }%>

        <input type="submit" value="Alterar">
        </form>

    </div>

</div>

    <% stmt.close();
        con.close();
        } catch(SQLException ex) {
            %>
            <h1>SQLException: <%=ex.getMessage()%></h1>
            <% } %>
        </body>
</html>

```

A.11 WEB page Adiciona Conta

```

<%--
    Document    : sAdicionaConta
    Created on  : 22/Jul/2012, 18:21:42
    Author      : Antonio
--%>

<%@page contentType="text/html" pageEncoding="UTF-8" import="java.sql.*"%>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
    "http://www.w3.org/TR/html4/loose.dtd">

<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html;
            charset=iso-8859-1">
        <meta name="keywords" content="">
        <meta name="description" content="">

        <title>CSS Layout</title>
        <link rel="stylesheet" type="text/css" href="reset-fonts-grids.css">
        <link rel="stylesheet" type="text/css" href="estilo.css">
        <script type="text/javascript" src="script2.js">
        </script>
        <style type="text/css">
    <!--
        .style1 {
            font-size: 24px;
            color: #000000;
        }
        .style2 {font-size: 36px}
    -->
        </style>
    </head>

    <body>

```

```

<%
    Connection con;
    Statement stmt;

    try {
        Class.forName("com.mysql.jdbc.Driver");
    } catch (java.lang.ClassNotFoundException e) { %>
        <h1>ClassNotFoundException: <%=e.getMessage()%></h1> <%
        }%>

    <%
    try {
        con = DriverManager.getConnection
            ("jdbc:mysql://localhost:63306/dbase","root","");
        stmt = con.createStatement();

        String nome = request.getParameter("txtNome").toString();
        String userName = request.getParameter("txtUserName").toString();
        String password = request.getParameter("txtPassword").toString();
        String confirmPassword = request.getParameter
            ("txtConfPassword").toString();
        String perfil = request.getParameter("perfil").toString();

%>

    <div id= "geral">

        <div id= "topo">

            </div>
            <div id= "menuNavegacao">
                <ul id="nav">
                    <li><a href="sInicio.jsp">Inicio</a></li>
                </ul>
                <li><a href="">Dispositivo</a>
                    <ul>
                        <li><a href="sAdicionarDispositivo.jsp">Adicionar</a></li>
                        <li><a href="sConsultarDispositivo.jsp">Consultar</a></li>
                    </ul>
                </li>
                <li>
                    <a href="">Configurar</a>
                    <ul>
                        <li><a href="testeSensor.jsp">Sensores</a></li>
                    </ul>
                </li>
                <li>
                    <a href="">Conta</a>
                    <ul>
                        <li><a href="sAdicionarConta.jsp">Adicionar</a></li>
                    </ul>
                </li>
                <li>
                    <div align="right"><a href="sConsultarUtilizador.jsp">Consultar</a>
                </div>
            </li>
        </ul>
    </div>

    <li><a href="sAjuda.jsp">Ajuda</a></li>
    <li><a href="logOut.jsp">Sair</a></li>
</ul>
</div>

```

```

        <div id= "conteudoGeral">
        <%

                ResultSet rs = stmt.executeQuery("SELECT *
                FROM user Where user.userName =
                '"+request.getParameter("txtUserName")+"'");
                if(rs.next()!= true){

                        if (request.getParameter("txtPassword").
                        equals(request.getParameter("txtConfPassword"))) {

                                String SQL = "Insert into user(userName,nome,password,perfil)
                                VALUES('" + userName + "',
                                '" + nome + "', '" + password + "', '" + perfil + "') ";

                                if( stmt.executeUpdate(SQL)>0);{%>
                                        <h1 style="color: green; text-align: center" >
                                        Utilizador registado com sucesso </h1>
                                <%}}else{

                                        String msg = "<h1 style=\"color:red;text-align:center;\>
                                        Confirme a password </h1>";
                                        session.setAttribute("msg", msg);
                                        response.sendRedirect("sAdicionarConta.jsp");
                                }

                                }

                                }
                                else{

                                        String msg = "<h1 style=\"color:red;text-align:center;\>
                                        Já existe um usuário com esse nome </h1>";
                                        session.setAttribute("msg", msg);
                                        response.sendRedirect("sAdicionarConta.jsp");
                                }

                                }
        %>

        </div>

        <div id= "rodape">
                <ul id="nav">
                <li><a href="/">Inicio</a> </li>
                <li><a href="/">Ajuda</a></li>
                <li><a href="logOut.jsp">Sair</a></li>
                </ul>
                rodape
        </div>
        </div>

        <% stmt.close();
                con.close();
                } catch(SQLException ex) {
                %>
                <h1>SQLException: <%=ex.getMessage()%></h1>
                <% } %>

        </body>
</html>

```

A.12 WEB page Adicionar Conta

```

<!--
Document   : sAdicionarConta
Created on : 5/Jul/2012, 14:10:40
Author    : Antonio
--%>
<%@page contentType="text/html" pageEncoding="UTF-8" import="java.sql.*"%>

```

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">

<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
    <meta name="keywords" content="">
    <meta name="description" content="">

    <title>Remote Control</title>
    <link rel="stylesheet" type="text/css" href="reset-fonts-grids.css">
    <link rel="stylesheet" type="text/css" href="estilo.css">
    <script type="text/javascript" src="script2.js">
  </script>
    <style type="text/css">
  <!--
  .style1 {
font-size: 24px;
color: #000000;
}
.style2 {font-size: 36px}
-->
    </style>
  </head>

  <body>

    <%
      Connection con;
      Statement stmt;

      try {
        Class.forName("com.mysql.jdbc.Driver");
      } catch (java.lang.ClassNotFoundException e) { %>
        <h1>ClassNotFoundException: <%=e.getMessage()%></h1> <%
        }%>

      <%
      try {
        con = DriverManager.getConnection
          ("jdbc:mysql://localhost:63306/dbase","root","");
        stmt = con.createStatement();
      %>

    </div id= "geral">

      <div id= "topo">
        <tr align="center"><td> <h1 style="color:#0000FF; text-align: Center" >
          Aquisição dados Remotamente </h1></td>
        </div>
        <div id= "menuNavegacao">
          <ul id="nav">
            <li><a href="sInicio.jsp">Inicio</a></li>
          </ul>
          <li><a href="/about/">Dispositivo</a>
            <ul>
              <li><a href="sAdicionarDispositivo.jsp">Adicionar</a></li>
              <li><a href="sConsultarDispositivo.jsp">Consultar</a></li>
            </ul>
          </li>
          <li>

```



```

<a href="/portfolio/">Configurar</a>
  <ul>
    <li><a href="testeSensor.jsp">Sensores</a></li>
  </ul>
</li>
<li>
<a href="/portfolio/">Conta</a>
  <ul>
<li><a href="#">Adicionar</a></li>
<li>
  <div align="right"><a href="sConsultarUtilizador.jsp">Consultar</a>
    </div>
</li>
</ul>
</li>

<li><a href="sAjuda.jsp">Ajuda</a></li>
<li><a href="logOut.jsp">Sair</a></li>
</ul>
</div>

<div id= "conteudoGeral">
<form name="form1" method="get" action="sAdicionaConta.jsp">
  <table border="0" width=100% align="center">
    <tr align="center"><td> <h1 style="color:greenyellow;
      text-align: left" >
        Criação de novo utilizador </h1></td>
    </tr>
  </table>
  <br><br>
<table border="0" width="316" >
  <tr><td width="161">Nome:</td><td width="145">
    <label for="txtNome"></label>
    <input type="text" name="txtNome"
      id="txtNome"></td></tr>
  <tr><td>User name:</td><td><label
    for="txtUserName"></label>
    <input type="text" name="txtUserName"
      id="txtUserName"></td></tr>
  <tr><td>Password:</td><td><label
    for="txtPassword"></label>
    <input type="text" name="txtPassword"
      id="txtPassword"></td></tr>
  <tr><td>Confirmar Password:</td> <td><label
    for="txtConfPassword"></label>
    <input type="text" name="txtConfPassword"
      id="txtConfPassword"></td></tr>

  <tr><td>Perfil:</td> <td>
  <select name="perfil">
  <%ResultSet rs = stmt.executeQuery
  (" SELECT * FROM perfis");
  while (rs.next()){
    String nomePerfil=rs.getString("nome");
    String IdPerfil=rs.getString("id_perfil");
    %> <option value= "<%=IdPerfil%" >
    <% out.println(nomePerfil); %></option> <%
  } %>
  </select></td></tr> <br><br>

  <tr><td></td> <td align="left">
  <input type="submit" value="Adicionar"></td></tr>

</table>

```

```
                </form>
                <br><br>
<%
String mensagem = (String) session.getAttribute("msg");
if (mensagem == null){
    mensagem = "";
}else{
    mensagem = mensagem;
}
out.println(mensagem);
%>

                </div>

</div>

<% stmt.close();
        con.close();
        } catch(SQLException ex) {
        %>
        <h1>SQLException: <%=ex.getMessage()%></h1>
        <% } %>

</body>
</html>
```