Universidade Federal de Uberlândia
Faculdade de Engenharia Elétrica

VITOR HUGO DAMAS PAREJA

UTILIZAÇÃO DE RFID PARA CONTROLE DE EQUIPAMENTOS MÉDICO-HOSPITALARES

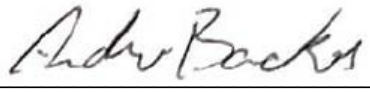
Uberlândia
2015

VITOR HUGO DAMAS PAREJA

UTILIZAÇÃO DE RFID PARA CONTROLE DE EQUIPAMENTOS MÉDICO-HOSPITALARES

Trabalho apresentado como requisito parcial de avaliação na disciplina Trabalho de Conclusão de Curso do Curso de Engenharia Biomédica da Universidade Federal de Uberlândia.

Orientador: André Ricardo Backes



Assinatura do Orientador

Uberlândia
2015

RESUMO

O objetivo deste trabalho foi propor um sistema que possibilita a utilização da tecnologia RFID no controle de equipamentos médico-hospitalares. Através das funcionalidades desta tecnologia, este projeto buscou identificar os principais dados que devem ser analisados para que tais ativos possam ser controlados. Isso contribui para a melhoria dos processos hospitalares que dependem destes recursos e diminuir tempos de busca de equipamentos, chances de roubo ou perdas e gastos de manutenção.

ABSTRACT

The purpose of this paper was the proposition of a system that makes possible the use of the RFID technology in the management of medical equipments. Through the functions of this technology, this project hoped to identify the essential data to be analyzed so that this actives may be managed. This improves the hospital`s processes that depend on those actives and shorten the time of search for those equipments, the theft rate and, also, downsizes the expenses in the maintenance section of the institution.

LISTA DE ILUSTRAÇÕES

LISTA DE ABREVIATURAS E SIGLAS

ABNT – Associação Brasileira de Normas Técnicas.

BD - Banco de Dados.

EAS - Estabelecimento de Atendimento de Saúde.

FEELT – Faculdade de Engenharia Elétrica.

ICSP - Programação Serial no Circuito (*In Circuit Serial Programming* em inglês).

IDE - Ambiente de Desenvolvimento Integrado (*Integrated Development Enviroment* em inglês).

OS - Ordem de Serviço

PWM - Modulação por Largura de Pulso (*Pulse Width Modulation* em inglês).

RFID - Identificação de Radio Frequência (*Radio Frequency Identification* em inglês).

SQL - Linguagem de Query Estruturada (*Structured Query Language* em inglês).

UFU – Universidade Federal de Uberlândia .

USB - Barramento Serial Universal (*Universal Serial Bus* em inglês).

UART - Receptor/Transmissor Assíncrono Universal (*Universal Asynchronous Reciever/Transmiter* em inglês).

SUMÁRIO

1 INTRODUÇÃO

Um dos grandes desafios dos gestores de Estabelecimentos Assistenciais de Saúde (EAS) é manter o controle sobre os ativos hospitalares: equipamentos, material hospitalar, medicamentos, banco de sangue, roupas de cama, vestuários e utensílios. Este controle tem influência direta no custo das organizações médicas, no desperdício e no mau uso do patrimônio e a sua utilização indevida têm contribuído para o aumento dos prejuízos dessas organizações [2].

O setor de manutenção em um EAS normalmente contempla um fluxo considerável de equipamentos, conforme o tamanho deste estabelecimento. De qualquer forma este fluxo de equipamentos nem sempre é registrado corretamente, causando, por vezes, desvio de materiais, prejudicando a instituição prestadora de serviços em saúde.

Uma solução que tem se mostrado eficaz consiste na etiquetação do acervo hospitalar por etiquetas de radiofrequência (TAGs), passivos e ativos, de diversos modelos em conjunto com a tecnologia de identificação por radiofrequência (sigla em inglês RFID) com a finalidade de controle de eficiência no plano de manutenção em vigor. Cada TAG é manufaturada de forma específica para cada tipo de material: ferro, papel, plástico, tecido e vidro.

O sistema proposto neste artigo contempla o uso de TAGs passivas (sem necessidade de baterias) de identificação por radiofreqüência (RFID sigla e inglês) associadas a equipamentos hospitalares, com o propósito de ser uma ferramenta para fornecer dados a partir dos quais se podem construir indicadores temporais de qualidade e de eficiência. Acoplado a um computador que serve de servidor, através de um software, contendo estas informações necessárias para a gestão de qualidade

Para desenvolvimento desta ferramenta foi utilizado como Banco de Dados (BD) o MySQL, como plataforma de programação para desenvolvimento do software o Visual Studio C#, além do microcontrolador ATMega na plataforma Arduino[14], para comunicação do software com os sensores de RFID. Na Seção 2 encontram-se informações básicas dos tipos de manutenção importantes na proposta do projeto. Em seguida, na Seção 3 as informações referentes ao BD escolhido, plataforma de desenvolvimento de software, microcontrolador e o sensor. Na Seção 4 encontram-se os detalhes para o desenvolvimento desta ferramenta. Serão descritos os processos de modelagem da base de dados, a conexão do microcontrolador com o software, desenvolvimento da interface do software, detalhes da implantação do sistema e, por último, detalhes sobre o hardware.

1.1 Justificativa

Um dos problemas da falta de controle é o desconhecimento da origem de manutenção destes equipamentos. É importante saber onde o equipamento não realizou seu objetivo para se realizar um estudo da causa desta manutenção para, a partir daí, implantar programas de treinamento de uso destes aparelhos. Ainda é importante documentar qual setor retirou o equipamento uma vez concertado, pois em posse destes dados pode-se controlar melhor a localização deste e evitar furtos ou ao menos determinar a responsabilidade por um desaparecimento.

Um sistema de manutenção deve diferenciar tipos de manutenção dados [19], com o objetivo de diminuir o número absoluto de manutenções devido às falhas operacionais em campo. No caso de um EAS isto é imprescindível pois uma falha em campo significa uma falha de diagnóstico ou, pior, uma falha ocorrida durante um procedimento médico. Este assunto é discutido na Seção 2 com mais detalhes.

A documentação correta das manutenções realizadas em cada equipamento, acrescido de outras informações relevantes como data, hora, setor de origem ou destino e tipo de manutenção, forma uma ferramenta de controle útil para a gestão do estabelecimento. Contudo se esta informação estiver mal organizada, dificultando seu acesso, surge o risco de uma gestão bem intencionada incorrer em uma demora muito grande para a tomada de decisão.

Diante disso, uma solução eficaz é requerida para que este setor possa ser melhor gerenciado, tanto no sentido de localização de aparelhos para uso imediato pelos profissionais da saúde como na gestão dos estoques e dos ciclos de manutenção preventiva e corretiva.

1.2 Objetivo

1.2.1 Objetivo geral

O objetivo geral deste trabalho é desenvolver um software para controle de equipamentos médico-hospitalares baseado em etiquetas de RFID

1.2.2 Objetivos específicos

Os objetivos específicos deste trabalho são:

- Armazenar a data de cada entrada ou saída no setor de manutenção;
- Integrar uma ferramenta para exportar dados para o Excel;

- Desenvolver uma interface intuitiva e funcional;

2 TIPOS DE MANUTENÇÃO

São vários os tipos de manutenção existentes em um ambiente hospitalar, classificadas usualmente em quatro grupos: manutenção corretiva, manutenção preventiva, manutenção preditiva e manutenção detectiva. As duas últimas são ferramentas para realização da manutenção corretiva programada, consistindo basicamente no conjunto de atividades de acompanhamento das variáveis ou parâmetros que indicam a performance ou desempenho dos equipamentos visando definir a necessidade ou não de intervenção [19].Portanto esta manutenção não implica na entrada ou saída do equipamento no setor de manutenção, logo não há necessidade de inclusão destas na ferramenta desenvolvida.

2.1 Manutenção preventiva

A manutenção preventiva é realizada quando se objetiva a diminuição da probabilidade de falhas ocasionais. A realização desta manutenção não implica diretamente numa maior confiabilidade no equipamento, mas implica numa proximidade maior com os padrões de fábrica ou com padrões estabelecidos na EAS para a boa função do equipamento.

Qualquer ativo físico solicitado para realizar uma determinada função estará sujeito a uma variedade de esforços. Estes esforços gerarão fadiga e isto causará a deterioração deste ativo físico, reduzindo sua resistência à fadiga. Esta resistência reduzir-se-á até um ponto no qual o ativo físico pode não ter mais o desempenho desejado. Em outras palavras, ele pode vir a falhar[18].

A vantagem deste tipo de manutenção é o seu baixo custo uma vez que é programada e realizada com data específica e não mediante a falha, podendo levar algum serviço a ser parado por tempo demasiado. Além do custo ainda, no caso das EAS o paciente pode correr risco de vida.

2.2 Manutenção corretiva

Este tipo de manutenção consite na atuação no equipamento para correção de uma falha ou de um comportamento abaixo do necessário. Usualmente se divide este tipo de manutenção em duas classes: não planejada ou previamente planejada.

2.2.1 Manutenção corretiva não planejada

Trata-se da correção de uma falha aleatória ou de desempenho menor que o desejado, ou seja, uma interrupção não prevista ocorreu em um processo, ocasionando provável ônus à instituição de atendimento em saúde. Não há tempo para a preparação de componentes e nem de planejar o serviço. Este tipo de manutenção usualmente não custa individualmente mais que uma manutenção preventiva, contudo o tempo de indisponibilidade do aparelho pode causar um prejuízo maior para a instituição [18].

2.2.1 Manutenção corretiva planejada

Neste tipo de manutenção uma falha ou condição anormal de operação de um equipamento e a correção depende de decisão gerencial, em função de acompanhamento preditivo ou pela decisão de operar até a quebra.

A decisão de adotar a política de manutenção corretiva planejada pode ser originada com base em vários fatores, tais como: negociação de parada do processo produtivo com a equipe de operação, aspectos ligados à segurança, melhor planejamento dos serviços, garantia de ferramental e peças sobressalentes, necessidade de recursos humanos tais como serviços contratados. Esse tipo de manutenção possibilita o planejamento dos recursos necessários para a intervenção de manutenção, uma vez que a falha é esperada. [12] [19].

3 FERRAMENTAS UTILIZADAS

O funcionamento idealizado do sistema consiste basicamente na interação do BD e do microcontrolador com o software e na interação do sensor RFID com o microcontrolador. Os dados de cada TAG partem do sensor para o microcontrolador através da comunicação *Serial Peripheral Interface* (SPI - Interface Serial Periférica), este microcontrolador salva os dados e os envia para o software uma vez que recebe os comandos deste, via USB. O software também realiza entrada de dados ao BD contendo informações recebidas pelo microcontrolador ou não, além de ter que realizar consultas ao BD através dos dados recebidos do microcontolador ou não. Portanto as ferramentas utilizadas para realizar o sistema proposto que, serão detalhadas a seguir são: MySQL (BD), Arduino (plataforma do microcontrolador), Visual Studio C# (Plataforma de desenvolvimento do software) e MFRC522 (Sensor RFID). Estas ferramentas se relacionam como mostra a Figura 1.

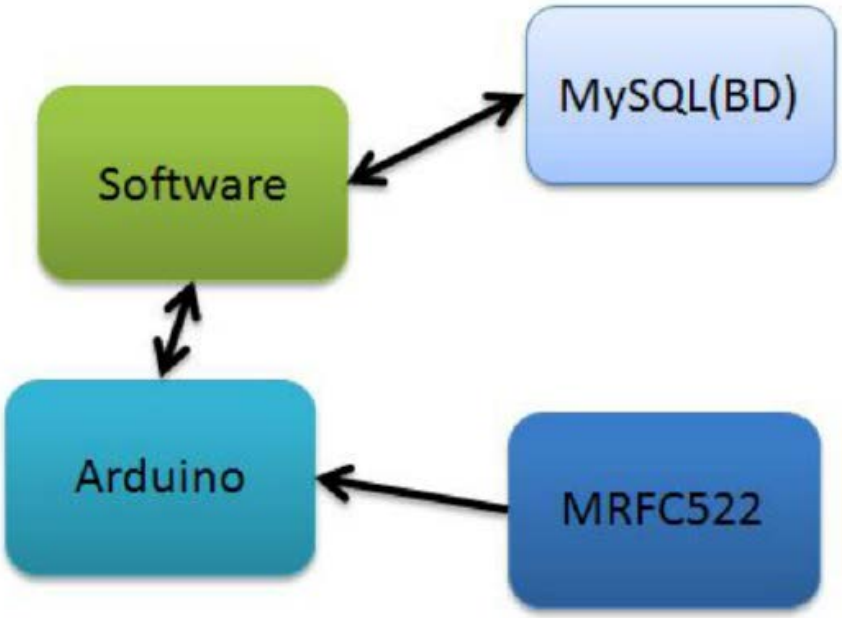


Figura 1. Arquitetura básica do sistema

3.1 Banco de dados

O BD deve se comunicar com o software para realização de operações de inserção, deleção, seleção e para realizar atualização de dados, portanto, não somente dados são enviados do software para o BD, mas também dados do BD são acessados pelo software.

Tendo em vista que para desenvolvimento de um banco de dados deve-se escolher uma plataforma para gerenciamento do mesmo verificou-se que, para os fins deste projeto, não seria

viável utilizar plataformas gerenciais de BD que não fossem gratuitas. Também foi considerada a compatibilidade com a plataforma de desenvolvimento escolhida para programação do software.

O mySQL é um sistema de gerenciamento de banco de dados gratuito muito usado pela facilidade de integração com ferramentas de construção de softwares como Visual Studio e a linguagem C#, além de aplicações web em PHP ou HTML. Apesar de ser open-source, ainda entrega um desempenho considerável e usa a linguagem Structure Query Language (SQL - Linguagem de Consulta Estruturada) que é uma linguagem desenvolvida pela IBM. Este software possui uma interface gráfica que facilita o aprendizado para quem nunca teve contato com bancos de dados estruturados com a linguagem SQL e possui artefatos que facilitam atualizações, deleções e inclusões diretamente na base de dados [16]. Estes artefatos facilitam o processo de criação do banco de dados em si por se tornar mais dinâmico e visual dispondo de ferramentas que geram a query (linha de comando do BD) baseado no que o usuário fez na interface. Assim este sistema permite que usuário não tenha, necessariamente, que escrever manualmente todas as querys para cumprir alguma tarefa, agilizando o manuseio desta base de dados.

A interface do *workbench* do mySQL é mostrada na Figura 2:

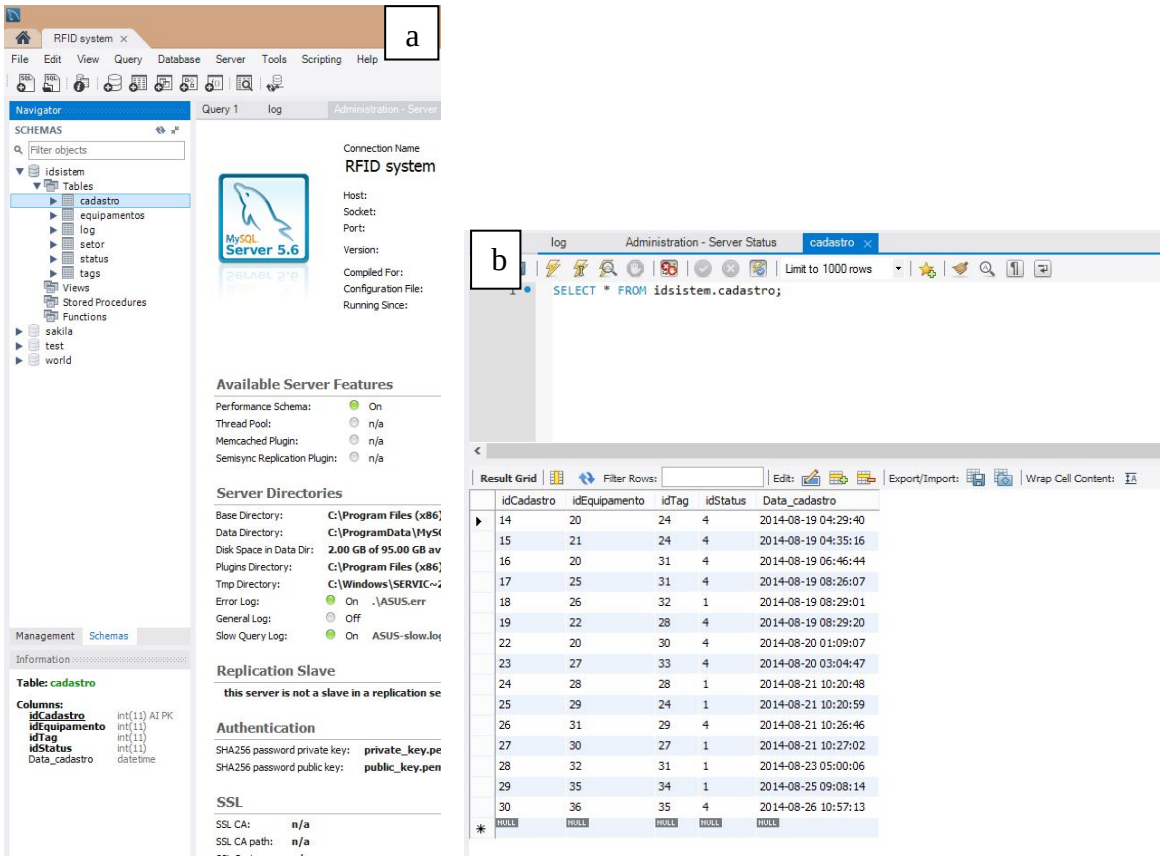


Figura 2.(a) Visão da lateral esquerda da plataforma gerencial. (b) Seleção do BD através da query .

3.2 Software

O software é o objeto de contato maior do usuário do sistema. É nele que se encontra a interface com a qual o usuário ira operar, e deve se comunicar tanto com o BD quanto com o microcontrolador. Este componente da arquitetura do projeto envia um dado para o microcontrolador que, ao recebê-lo, envia de volta os dados lidos pelo sensor RFID.

Para desenvolvimento deste software deve ser escolhido o ambiente de desenvolvimento deste, com base no sistema operacional típico do local de implantação do mesmo. No processo de determinação de qual plataforma escolher levou-se em consideração a implantação típica em wondows 7, windows 8 e windows 8.1, uma vez que são sistemas com maior popularidade em termos numéricos que sistemas como MacOS ou Linux [13].

A plataforma escolhida para desenvolvimento do software foi a plataforma .NET (lê-se dot Net) da Microsoft, que foi criada com objetivo de unir em uma plataforma o desenvolvimento e execução de sistemas e aplicações. De forma semelhante a plataforma Java da Oracle, qualquer código gerado nesta plataforma .NET poderá ser executado em um dispositivo que tenha este framework.

Neste contexto surgiram varias linguagens de programação para desenvolvimento de aplicações nesta plataforma e uma delas é o C# (lê-se C sharp). Ferramenta desenvolvida pela própria Microsoft.

O C# é uma linguagem de programação orientada a objeto baseada na linguagem C++ mas que inclui referências a linguagens como Java e object Pascal [15]. Esta linguagem traz um conjunto de ferramentas que a transformaram em uma linguagem simples, unindo características de simplicidade e facilidade do Visual Basic com a robustez de uma linguagem como C++.

A interface do Microsoft Visual C# 2010 Express segue na Figura 3:

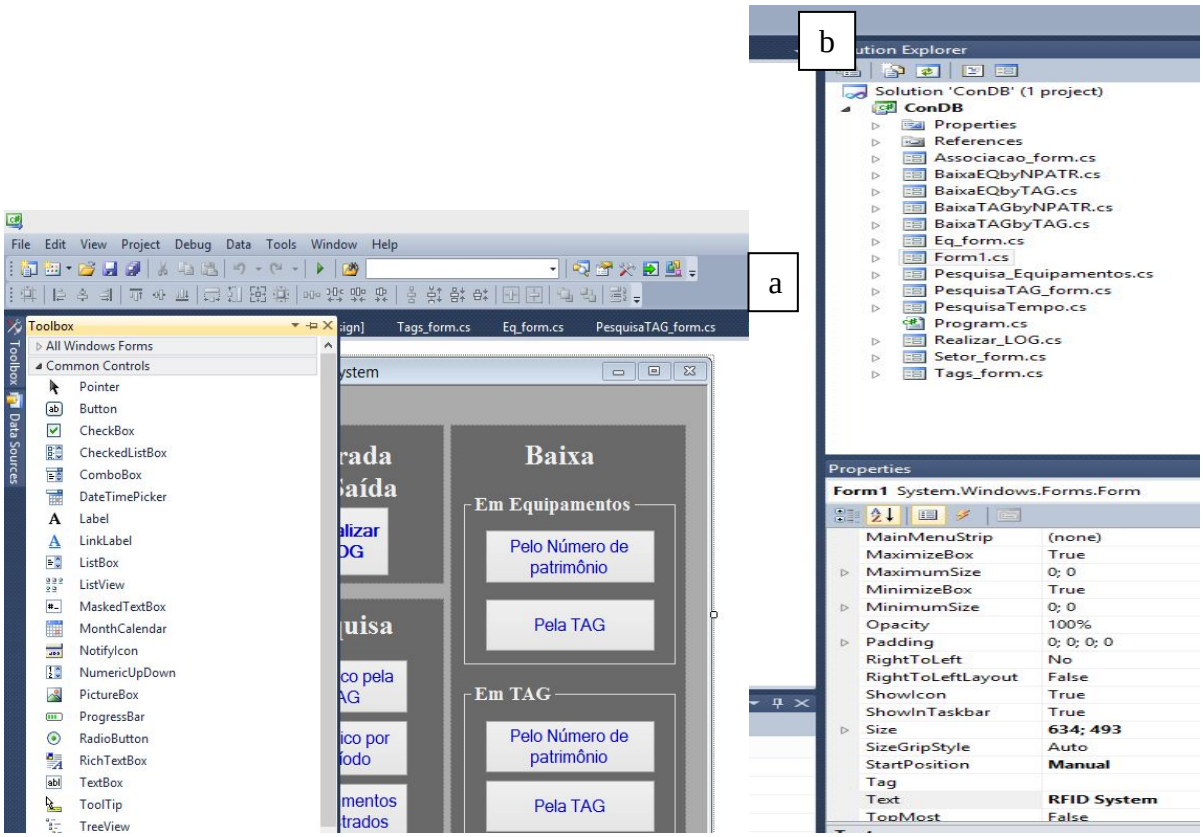


Figura 3. (a) Visão da lateral esquerda do Microsoft Visual C# 2010 Express. (b) Visão da lateral direita do Microsoft Visual C# 2010 Express.

3.3 Microcontrolador

O microcontrolador é basicamente um computador integrado em um chip, contendo um processador, memória e periféricos que podem ser programados para uma função específica, ao contrario dos microprocessadores de propósito geral usados em computadores.

A funçào do microcontrolador neste projeto é receber a identificação da TAG e enviá-la para o software quando o comando correto for recebido proveniente do mesmo, via USB. Para a determinação do microcontrolador utiliazado foi levado em conta as bibliotecas disponíveis para este microcontrolador, sua possibilidade de comunicação com o software e a linguagem da plataforma de desenvolvimento de seu microprocessador. O microcontrolador encontrado, mais compatível com esta aplicação foi o Arduino.

O Arduino é uma plataforma eletrônica open-source composta por uma placa que contém um microcontrolador ATmega1280 e por um ambiente de desenvilvimento que facilita a comunicação com o microcontrolador via USB. O software desta ferramenta, que é fornecido pelo fabricante da plataforma Arduino, foi escrito em Java e se baseia, principalmente, nas

linguagens de programação “Processing” e “avr-gcc”. Este ambiente de desenvolvimento integrado (IDE - *Integrated development enviroment* em inglês) permite a construção de sketches que podem ser verificadas em sua sintaxe automaticamente, permite gravação no microcontrolador via USB, além de permitir monitoramento da porta serial, o que facilita testes de comunicação com o software desenvolvido para a solução deste projeto.

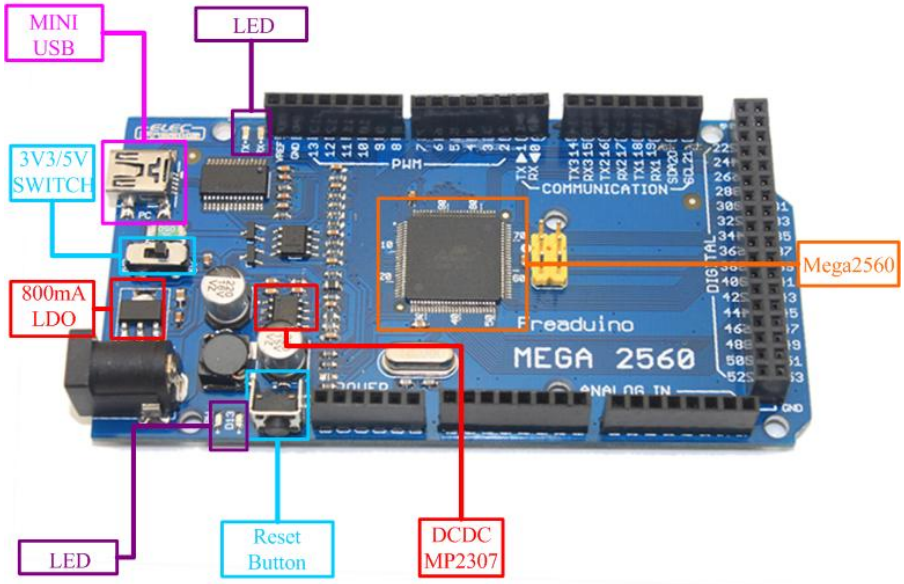


Figura 4. Arduino Mega 2560.

A placa de desenvolvimento Arduino MEGA (Figura 4) recebe o microcontrolador ATmega1280 da atmel, que contém cinquenta e quatro pinos digitais de entrada e saída dos quais catorze podem ser usados para saída PWM (*Pulse width modulation* em inglês), desseseis entradas analógicas, quatro UART’s (*Universal Asynchronous Reciever/Transmiter* em inglês), um cristal oscilador de 16 MHz como fonte de clock externa para o microcontrolador, uma porta USB, uma porta para fonte de alimentação, suporte para ICSP (*In Circuit Serial Programming* em inglês) e um botão de reset. Este microcontrolador opera com cinco ou três volts e recebe como entrada uma faixa de sete à doze volts [17].

Nesta solução este microcontrolador foi alimentado pela própria porta USB, ou seja, com cinco volts. Foram usadas as portas digitais para a cominicação SPI e a saida de 3.3 volts que alimenta o sensor RFID, além das portas analógicas para um display de LCD e um LED vermelho.

3.4 Sensores RFID

O sensor RFID para TAGs passivas funciona com base na leitura de ondas de radio-frequência retornadas pela TAG, ondas que são emitidas por ele mesmo atraves de uma antena. O sensor é dotado de um leitor de ondas deste tipo que recebe a onda e converte-a em informação digital.
A partir da comunicação SPI este sensor envia a informação digital para o microcontrolador que pode manipular os dados conforme sua aplicação. A complexidade deste leitor depende da TAG a ser usada e da aplicação desta tecnologia.
Para definição do sensor RFID nesta solução, constatou-se viável a utilização de módulos RFID de 13,56MHz com comunicação SPI compatível à bibliotecas disponíveis para Arduino. Logo foi definido o sensor de RFID MFRC 522[6], que utiliza 3.3V, diminuindo gasto energético, além de estar em conformidade com a ISO 14443 [8][9][10][11], que regulamenta cartões de identificação.
Foi determinado a partir do sensor escolhido, que a TAG ou cartão que será lido pelo sistema deve obedecer aos critérios da mesma ISO 14443 [8][9][10][11]. Após uma pesquisa de mercado, verificou-se a existência de TAG's do tamanho aproximado de uma moeda com a funcionalidade de chaveiro da empresa MIFARE que atende estas especificações. Esta mesma empresa disponibiliza este chipset em cartões e TAG's de RFID, modelo do chipset : MF1S503x [5]. A Figura 5 mostra uma foto do conjinto sensor + TAG.

Figura 5. No sentido horário: TAG cartão, MFRC522, Trilho e TAG chaveiro.

4 ATIVIDADES DESENVOLVIDAS

Primeiramente houve a determinação de qual seria o propósito do sistema: monitorar a entrada e saída de equipamentos no setor de manutenção via TAG's de RFID e adicionar informações importantes para a gerência tomar decisões.

Para esta finalidade deu-se início a discussão sobre quais dados seriam manipulados e acessados, além de quais seriam os tipos de dados que seriam utilizados. Na Seção 4.1 encontra-se um detalhamento do processo de criação destes dados.

Ainda houve uma preocupação com o reconhecimento automático, pelo software, da porta USB na qual o Arduino esta conectado, eliminando a necessidade do usuário procurar a nomenclatura desta porta (COM1, COM2...) no gerenciador de dispositivos. Na Seção 4.2 se encontra um detalhamento de como isto foi feito, nesta mesma contém informações sobre como o Arduino se comunica com o software.

Este software priorizou uma interface simples e funcional, com o objetivo de minimizar o erro do usuário. Nesta interface foram incluidas as ações possíveis ao usuário, estas serão descritas em maiores detalhes na Seção 4.3. Um aspecto importante deste sistema, o qual tem que ser detalhado, é a implementação do mesmo.

Na Seção 4.4 será detalhado o processo de implementação do Visual Studio C#, gerenciador de BD e drivers do Arduino. Nesta mesma serão, ainda, detalhados passos para exportação e importação de dados previamente no BD. Ainda serão expostos na Seção 4.4 detalhes sobre as bibliotecas utilizadas, sem as quais o sistema não funciona, apesar de tudo instalado corretamente, estas bibliotecas serão incluidas tanto no Visual C# 2010 como na IDE do Arduino.

Também foi desenvlvida uma estrutura para armazenar o hardware deste sistema, que contempla o Arduino MEGA, sensor RFID e display de LCD. Esta estrutura deveria levar em conta que o sensor não ficaria exposto, contudo não poderia sofrer interferência pela estrutura. A Seção 4.5 apresenta detalhes sobre os materiais escolhidos, bem como os métodos de prototipagem deste sistema. Ainda nesta, serão apresentadas as simulações realizadas para melhor visualização da disposição das conexões entre os componentes deste sistema.

A Seção 4.6 apresenta uma discussão sobre as dificuldades encontradas e maiores detalhes sobre o desenvlvimento desta ferramenta.

4.1 Modelagem da base de dados

Mediante a discussão para decisão dos tipos de dados, foi decidida a realização de um modelo de entidade-relacionamento com o propósito de visualizar os relacionamentos entre as entidades do BD, bem como a determinação de quais seriam os formatos destas entidades. Este modelo, descrito por Peter Chen [4], propõe uma forma de visualização organizada, que define três conceitos: Entidades, Atributos e Relacionamentos. Na prática estas entidades são as tabelas que serão construídas no BD, os atributos são as colunas destas tabelas, e os relacionamentos são como as tabelas se relacionam, 1:1, 1:N ou N:N .

Inicialmente foram propostas as tabelas setor, equipamentos, tags, cadastro e log. Sendo a tabela "log" idealizada para servir de arquivo para a principal função do sofware, registrando entradas e saídas do setor de manutenção do EAS e devendo conter, entre outras informações, qual aparelho, data e o tipo de manutenção ou indicação de saída do equipamento.

A tabela "setor" foi idealizada pela facilidade em eliminar o erro do usuário, uma vez que, ao invés do usuário digitar o setor de origem ou destino do aparelho, ele iria escolher a partir de uma lista de opções, eliminando entradas erradas na tabela "log".

As tabelas "equipamentos" e "tags" conteriam as informações referentes ao parque tecnológico do EAS e às TAG's adquiridas para implantação deste projeto, estas tabelas serveriam de base para a construção da tabela "cadastro".

A tabela "cadastro" foi planejada para conter as associações entre um equipamento e uma TAG, uma característica desta tabela é o fato que um equipamento pode conter uma ou mais entradas na mesma, para o caso de TAG's defeituosas ou desaparecidas. Outra característica é a possibilidade de mais de uma entrada de uma mesma TAG para o caso de equipamento descartado e TAG ainda utilizável.

Contudo, levando em conta que equipamentos, TAG's e cadastros poderiam alternar seus estados entre ativos e inativos, decidiu-se pela adição da tabela Status. Esta tabela também se relaciona com a tabela LOG para especificar o tipo de manutenção que o aparelho irá receber, além de poder significar a saída do aparelho na manutenção ou ainda a baixa deste pela impossibilidade de concerto.

Ainda havia sido determinado que a tabela "tags" teria como chave primária o código da TAG em hexadecimal, enviado pelo microcontrolador. Porém, verificou-se que o sistema ficaria mais simples se a chave primária fosse gerada pelo próprio sistema. De forma similar ocorreu uma mudança na tabela "equipamentos", que inicialmente tinha como chave primária o número de

patrimônio. Estas mudanças melhorariam os mecanismos de pesquisa utilizados para consulta e geração de relatórios no software.

Outra mudança do modelo inicial ocorreu na tabela "cadastro" que não contava com o atributo "Data_cadastro". Este atributo foi acrescido à tabela por facilitar na determinação da data na qual uma TAG foi associada a um equipamento e pela importância desta informação para o monitoramento de eficiência do sistema de controle de equipamentos.

O modelo entidade relacionamento final segue na Figura 6.

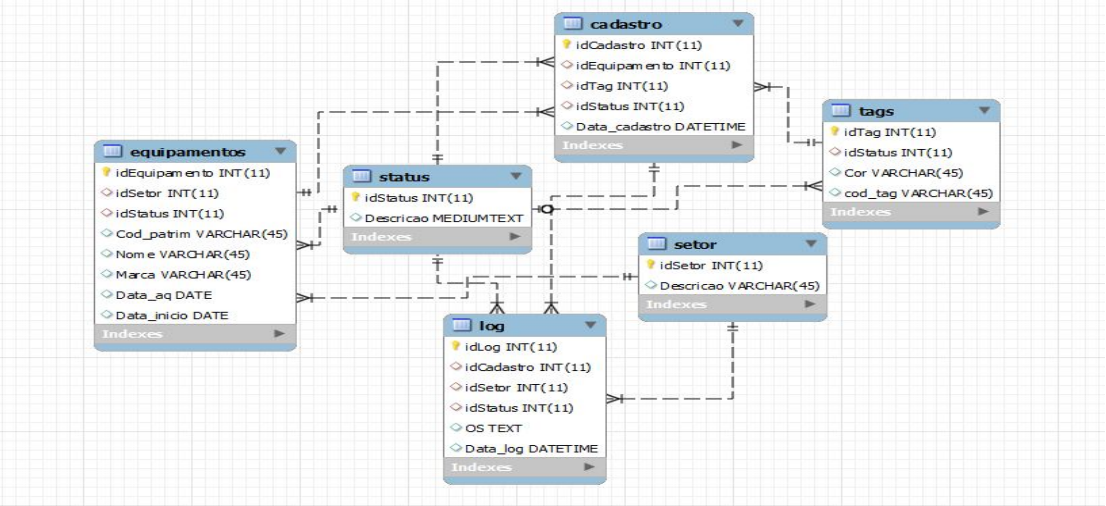


Figura 6. Diagrama de entidades do BD

Concluiu-se que todas as tabelas, exceto "setor", teriam a chave estrangeira "idStatus". A tabela "equipamentos" deveria conter uma referência a qual setor o equipamento foi originalmente destinado, portanto possuiria a chave estrangeira "idSetor". Na tabela "cadastro" optou-se por não conter o ultimo estado do equipamento na tabela "log", já as entidades "idTag" e "idEquipamento" seriam necessárias.

A tabela "log" conteria a entidade "idCadastro" a partir da qual se acessaria ambas "idTag" e "idEquipamento", conteria, também, "idSetor" para informar de qual setor o equipamento vem ou pra qual irá. Nesta mesma tabela contém a entidade "Data_log" primordial para o projeto. Segundo COHEN [3] este dado é primordial pra construção de indicadores de manutenção como tempo médio de reparo e outros.

Tambem optou-se por acrescentar a entidade "OS" na tabela "log" que armazenará um conjunto de caracteres que podem ser alfabéticos ou numéricos representando o código da ordem de serviço desta manutenção. Este dado se mostra útil para a gestão quando é necessário rastrear a ordem de serviço a partir da data de entrada do equipamento, no caso, este software daria acesso rápido à referência desta OS. Uma vez com o número de OS em mãos, o gestor pode

verifica-la melhor no sistema implantado para catalogação destas OS's, seja ele um software ou um sistema físico não automatizado.

4.2 Conexão com o Arduino

A conexão do software com o Arduino foi idealizada via USB e de forma inuitiva para o usuário sem muito conhecimento em informática. Para isso foi decidido que o software deveria reconhecer o Arduino automaticamente, uma vez que o processo para descobrir qual porta se comunica com o Arduino exige um conhecimento superior ao de uma parcela de usuários da ferramenta proposta.

Para este reconhecimento automático deve-se ocorrer a aquisição automática pelo software da porta na qual o Arduino está conectado. Isto foi feito por uma função, que através da biblioteca "System.management", acessa todas as portas USB pesquisando a descrição e o identificador de cada porta, e verificando se a descrição contem a palavra "Arduino", em caso positivo a função retorna uma *string* com o identificador da porta..

O método de acesso aos dados registrados pelo Arduino foi idealizado da seguinte maneira: O usuário deve informar o sistema quando uma TAG será lida, se o usuário não clicar no botão de iniciar a aquisição e aproximar a TAG do sensor uma mensagem aparecerá no display de LCD informando o erro, como mostra a Figura 7.

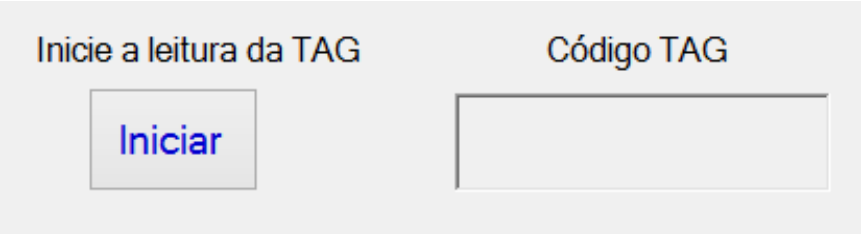


Figura 7. Método de acesso aos dados da TAG.

Este método foi usado como uma forma de sinalizar ao microcontrolador quando o dado deveria ser enviado ao computador. Ao pressionar o botão de iniciar a aquisição o software escreve a letra "y" na porta USB usando funções contidas na biblioteca "System.IO.Ports", este dado, enviado ao microcontrolador, entra em uma função condicional contida no código programado no microporocessador. Se a condição de verificação do caractere "y" for confirmada, o Arduino envia os cinco Bytes contendo o código da TAG ao software.

Para a manipulação dos dados do sensor de RFID foi utilizada uma biblioteca disponível em domínio público [1] chamada "RFID.h" e outra que faz a cumunicação do microcontrolador com o leitor de TAG's via Interfaceamento serial periférico (SPI - sigla em inglês), de nome SPI.h. Estas bibliotecas foram adicionadas ao programa executado pelo microcontrolador Arduino, através

delas foi possível acessar os estados do sensor e realizar a aquisição de dados byte a byte da identificação de cada TAG.

4.3 Desenvolvimento da Interface

Inicialmente verificou-se que o usuário deveria ter acesso à quatro tipos de ações: cadastrar, pesquisar, dar baixa e realizar entrada ou saída de equipamento. Estas ações foram dispostas de forma objetiva na interface inicial do software, como mostra a Figura 8.

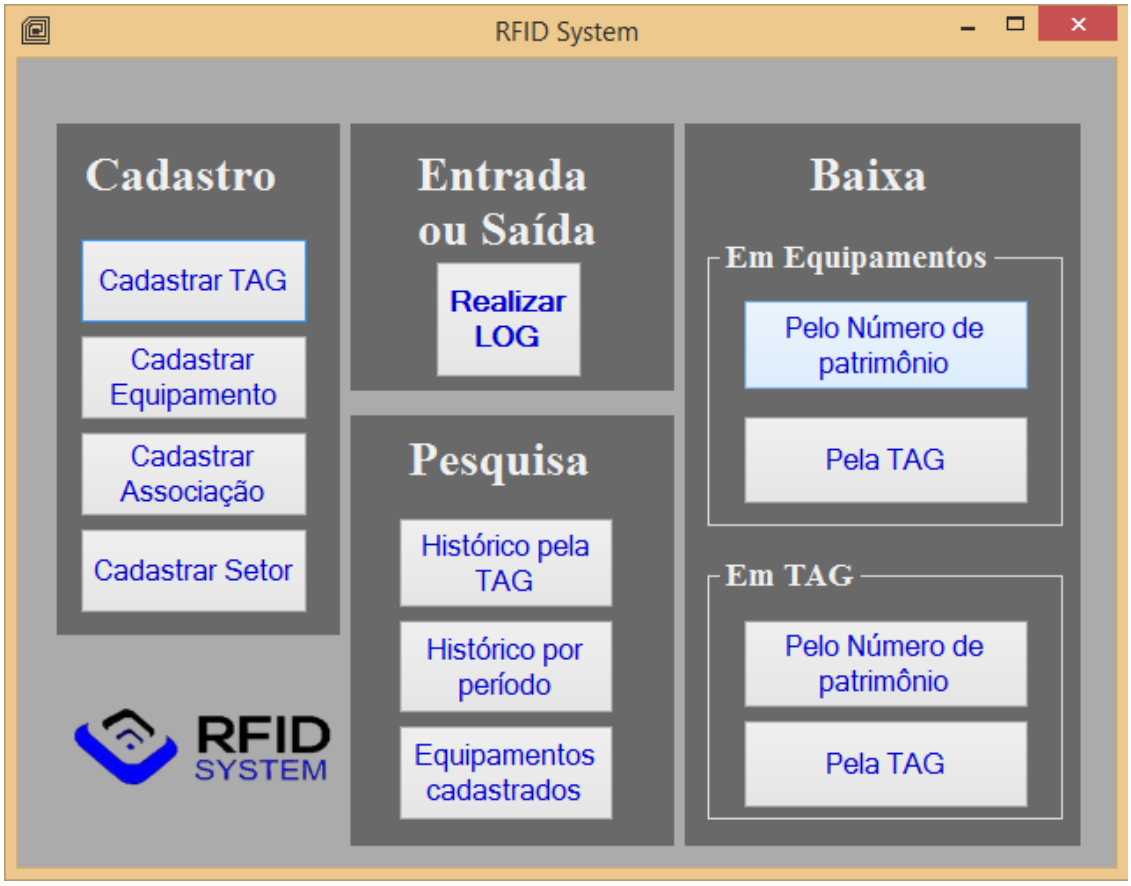


Figura 8. Interface principal do software.

A ação de cadastro realiza entradas nas tabelas idealizadas para o sistema. Contudo não é acessível ao usuário o cadastro de um novo status, isto deve ser feito pelo desenvolvedor do software, pois implica em mudanças na lógica de algumas funções.

Verificou-se a necessidade do usuário realizar o cadastro de todos os novos equipamentos que entram para o parque tecnológico, assim como de novas TAG's não cadastradas no sistema. O sistema só aceita uma nova associação de equipamento à TAG quando ambos estão cadastrados no BD, portanto este é outro tipo de cadastro implementado. Ainda realizou-se a

opção de cadastro de novo setor para o caso de expanção ou reorganização dos setores do EAS.

Foi identificada a necessidade do usuário poder dar baixa em equipamentos e em TAG's com defeito. Para suprir esta, foram realizados dois modos de baixa para cada um destes: um através do número de patrimônio e outro através da leitura da TAG. Estes métodos se justificam para os equipamentos no caso deste não ser retornado com a TAG acoplada (baixa através do número de patrimônio) e para o caso de número de patrimônio desconhecido (baixa através da leitura da TAG). No caso das TAG's estas baixas serão feitas por dois motivos: para TAG's com problemas de leitura (utlizando o número de patrimônio) e para TAG's com defeito no chaveiro (através da leitura desta TAG).

A ação de realizar entrada ou saída, principal função do sistema, foi desenvolvida para ser realizada em equipamentos com o cadastro ativo na tabela "cadastro" e é feita mediante leitura da TAG associada. Nesta ação o usuário deve informar o setor de origem ou destino do equipamento, a OS e o novo status do ativo. Portanto, se o equipamento está entrando no setor de manutenção, o sistema pode informar ao gestor qual setor gerou esta OS. Se estiver saindo do setor, o sistema fornece qual setor realizou a última requisição deste, fornecendo ao gestor um controle maior do fluxo destes equipamentos. Deste modo, pode-se realizar mudanças no esquema de distribuição de equipamentos no EAS, se necessário.

4.4 Implantação do Sistema

A implantação do sistema deve contemplar a instalação, primeiramente, do driver do Arduino, juntamente com sua IDE, disponível para download no site principal do produto. A instalação do driver para o Arduino é essencial para que a fase de reconhecimento automático via software funcione, uma vez que o driver estiver desatualizado a porta não conterá a palavra "Arduino" na descrição, como esperado no procedimento descrito na Seção 4.2. Ainda, deve-se salvar as bibliotecas utilizadas na pasta "libraries" para poder acessar o programa executando no microcontrolador.

Em seguida deve-se realizar a instalação do gerenciador de BD, o MySQL workbench 6.2. Recomenda-se o instalador MySQL installer 5.6.12 [7] pois a partir dele pode-se escolher com facilidade os componentes desejados ao invés de instala-los separadamente, correndo risco de erros ocorrerem durante o processo.

O próximo passo é a transferência do banco de dados, que através de uma ferramenta do *workbench* do MySQL pode ser facilmente exportado como mostra a Figura 9. Esta função gera

uma série de comandos, estruturados no esquema SQL, que quando importados no novo servidor, trazem as informações de todas as tabelas criadas, juntamente com suas entidades e características.

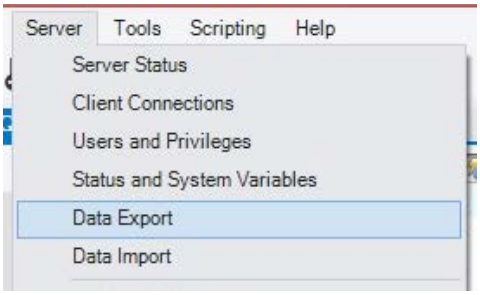


Figura 9. Exportando formato do BD e seus dados.

Exportando o formato do BD, juntamente com os dados previamente contidos nele, é rápido e intuitiva a importação destes para o novo servidor. Contudo é necessário precaução no momento da definição do endereço do novo servidor, se este for diferente de 127.0.0.1:3306 , se conter uma senha diferente de "vhdp" ou se o usuário não se chama "*root*", uma intervenção é necessária para alterar o modo de acesso ao BD pelo software.

Com estes dois componentes instalados, deve-se instalar o Microsoft Visual C# 2010 Express e realizar o cadastro gratuito obtendo uma licença para o produto. Após esta instalação deve-se abrir o projeto, copiado do original, e salvá-lo. Ainda é necessária a inclusão das bibliotecas utilizadas. O usuário recebe a indicação de bibliotecas não adicionadas, portanto deve incluir todas as bibliotecas requisitadas até o programa conseguir executar perfeitamente. As bibliotecas mais importantes são "MySql.Data.MySqlClient", "System.Management", "System.IO.Ports" e " Microsoft.Office.Interop.Excel " .

Com o software funcionando o operador deve cadastrar todos os equipamentos que serão integrados à este projeto, assim como as TAG's que serão usadas e setores do EAS. Também é necessária a associação dos equipamentos às TAG's, a partir daí o sistema esta em pleno funcionamento usando este computador como servidor.

4.5 Desenvolvimento do hardware

A parte do hardware compreende o Arduino MEGA, sensor RFID e display de LCD. Inicialmente, no projeto, seria utilizado apenas um LED para indicação de leitura da TAG, contudo decidiu-se realizar esta indicação através do display de LCD em conjunto com o LED.

Para a realização de testes destes componentes foi realizada uma montagem em uma placa de desenvolvimento utilizada para projetos, chamada protoboard. O esquema desta base de testes ficou como mostra a Figura 10

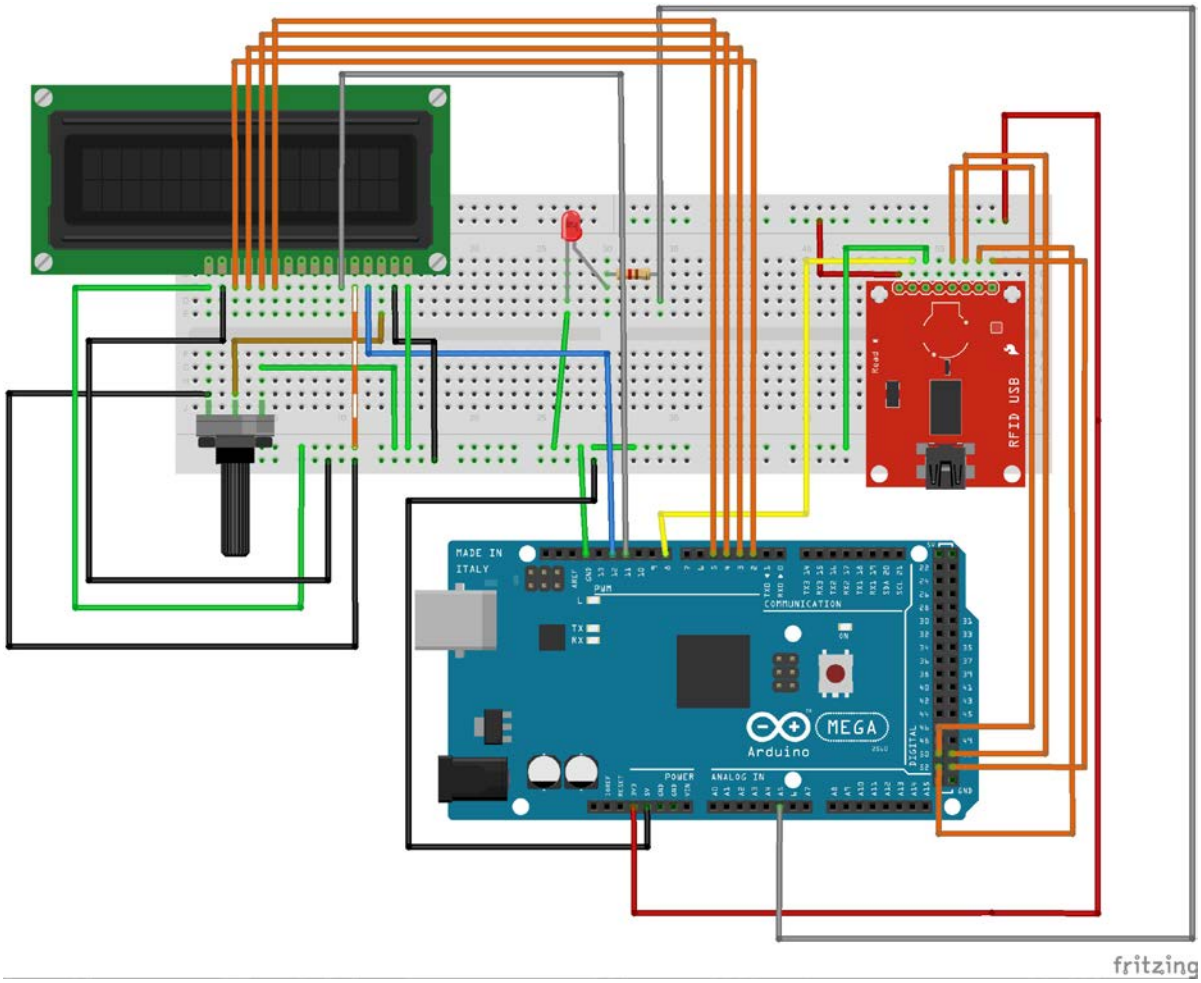


Figura 10. Esquema de ligação no protoboard.

Para o controle do contraste do dislplay de LCD foi incluido um potênciometro que permite este ajuste. Além disso, um resistor foi acoplado ao LED para limitar a corrente.

Ao fim do processo de testes foi decidido o desenvolvimento do esquema elétrico do projeto para melhor visualização deste, facilitando o processo de prototipagem. O resultado é mostrado na Figura 11.

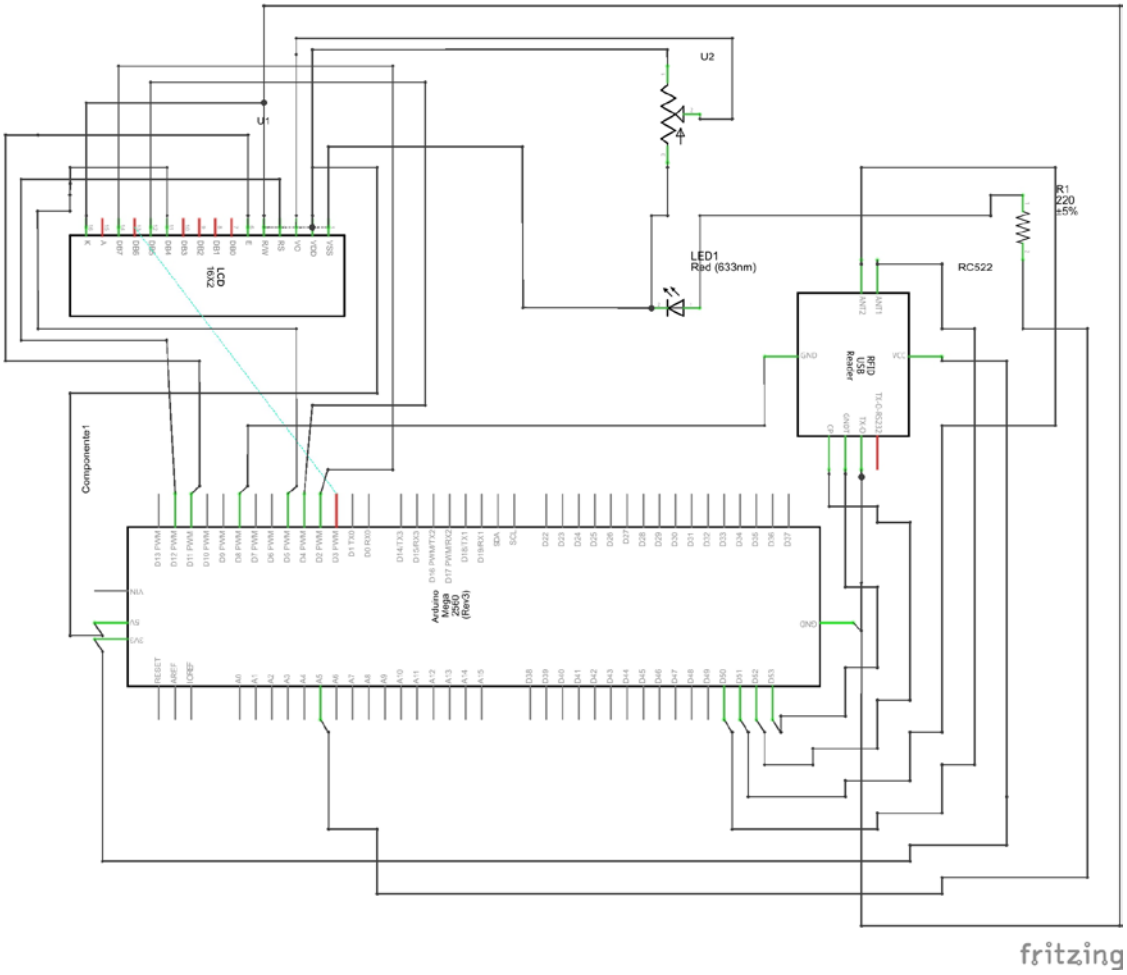


Figura 11. Esquema elétrico do hardware.

Os esquemas das Figuras 10 e 11 foram realizados com a ajuda de um software open-source chamado Fritzing, que possui bibliotecas contendo informações sobre os componentes utilizados, e ferramentas úteis para a modelagem deste projeto. Um dos desafios para esta modelagem foi achar um software que continha todos os componentes utilizados, e o Fritzing continha todos sem excessão.

A partir destes esquemas foi iniciado o desenvolvimento do protótipo para testes finais. Para esta aplicação, não constatou-se necessário o desenvolvimento de uma placa de circuito impresso, uma vez que uma solução barata e funcional seria usar placas de circuito impresso prontas (Figura 12). Estas placas, também chamadas de *Breadboars*, são placas de prototipagem que podem apresentar dois tipos de ligação em seus pontos de acesso, isolados e não isolados.

Por serem modeláveis, foram adaptados para um tamanho aceitável, haja vista que a caixa na qual o hardware ficaria não deveria ultrapassar em muito o tamanho do Arduino MEGA, o qual

deve se conectar ao computador via USB. Houve a modelagem de uma placa para o LCD e outra para o sensor RFID.

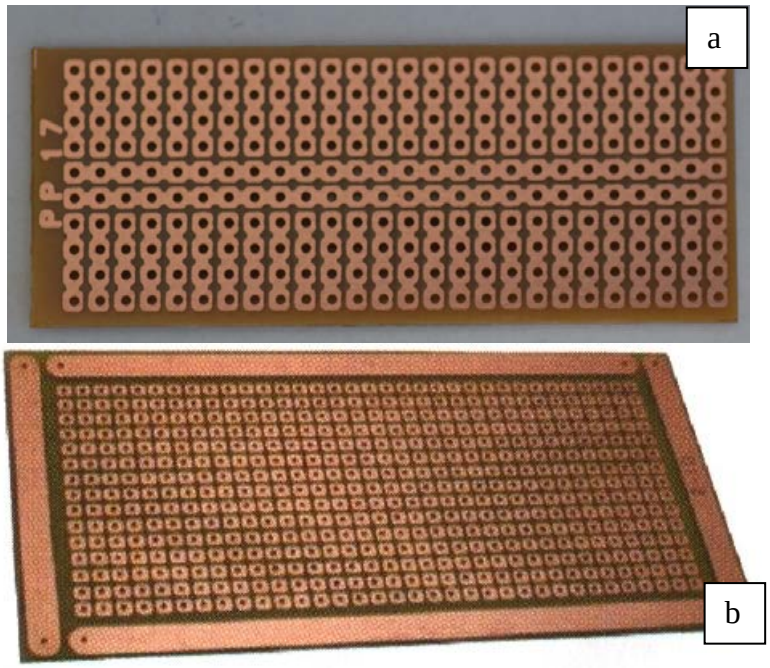


Figura 12. (a) Placa não isolada. (b) Placa isolada.

A conexão com o arduino será realizada com *jumpers* machos (Figura 13) soldados neste *Breadboard* modelado para o LCD e para o sensor.

Figura 13. Jumper macho.

Estas placas foram idealizadas para uma caixa de madeira, um material que não interferere no sinal RFID. Esta caixa tem 16x12x8cm e foi utilizada a fixação dos componentes por meio de parafusos autoparafusantes, sem o uso de roscas, e acondicionamento dos fios por fitas adesivas. O resultado final é mostrado na Figura 14.

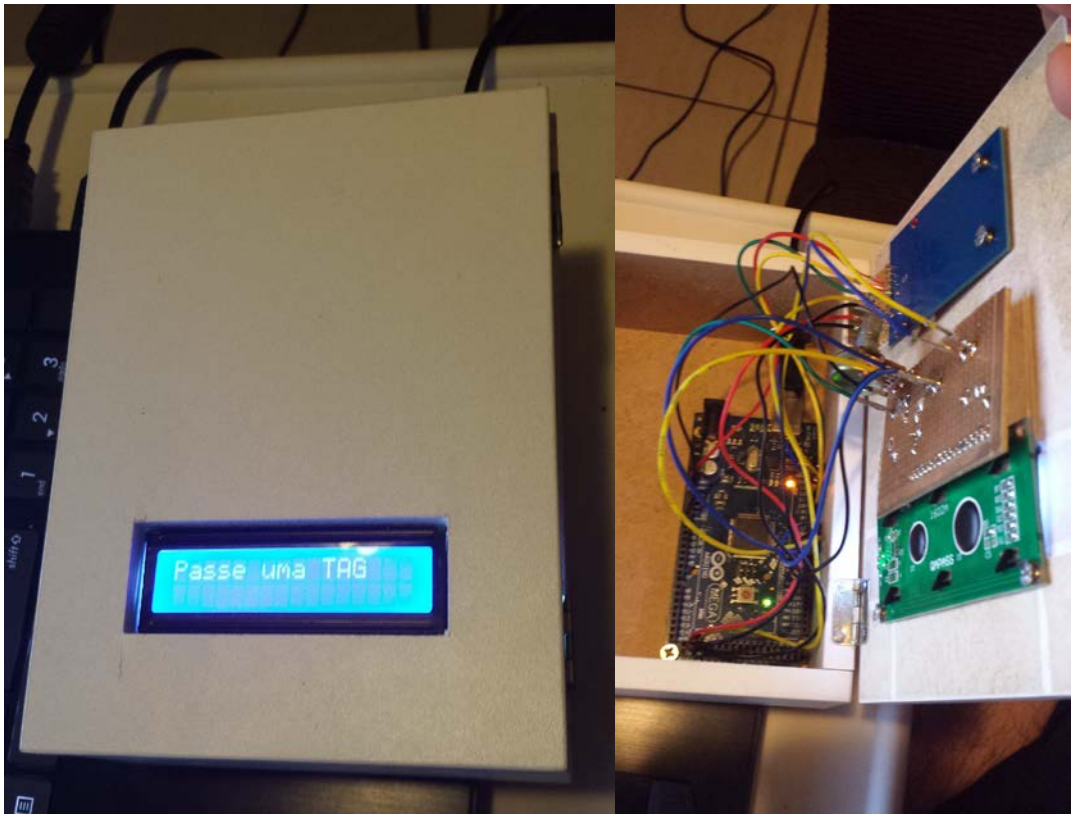


Figura 14. Hardware final.

4.6 Dificuldades encontradas

Foram constatadas algumas dificuldades comuns para este tipo de projeto. Quando se desenvolve um software para ser utilizado por qualquer faixa etária, deve-se priorizar a construção de uma interface que permite acesso rápido à suas funções.

Para isso, primeiramente foi idealizada uma interface com barra de ferramentas, que conteria o título genérico de cada função, como cadastro, e em seus campos, as opções para aquela função, como cadastro de setor e etc. Mantendo somente uma janela aberta, contudo o campo de trabalho não ficava estéticamente agradável, e a dinâmica de funcionamento das janelas de trabaho não era aceitável quando o usuário não seguia a ordem lógica de procedimentos. Portanto foi decidido por evitar que mais de uma função fosse executada ao mesmo tempo, somente uma função por vez, negando ao usuário realizar qualquer função enquanto uma ação não fosse finalizada.

Outro aspecto do projeto que se provou um desafio, foi realizar o tratamento de erros de operação do usuário. Erros como falhas de preenchimento de campos (para casos nos quais o usuário não preencher todos os campos necessários para realização da função desejada) foram tratados em todos as janelas do software.

Um erro que demandou uma habilidade maior para ser resolvido ocorria quando o usuário pressionava o botão "iniciar", para iniciar a leitura do sensor, e desejava passar para o próximo campo sem realizar a leitura da TAG. Foi implementado um código para desabilitar os outros campos enquanto a leitura não fosse realizada e exibir uma mensagem explicando ao usuário o que o sistema espera (Figura 15).

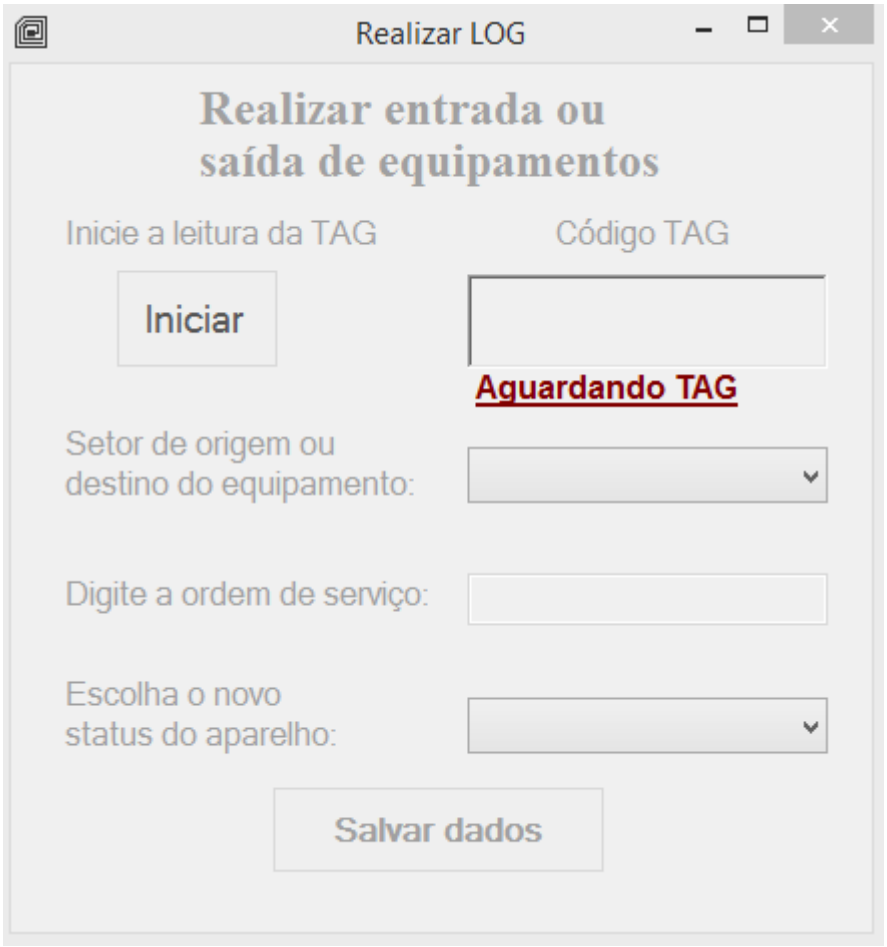


Figura 15. Tratamento para aguardar leitura.

Também foi tratado o erro de tentativa de leitura sem a conexão USB com o microcontrolador. Este erro foi tratado monitorando o retorno da leitura de aquisição automática descrita na Seção 4.2. Caso o retorno fosse nulo, significaria que o driver do arduino não estaria instalado, ou que

o arduino não estaria conectado ao computador. Este erro foi tratado por uma caixa de mensagem conforme a Figura 16.

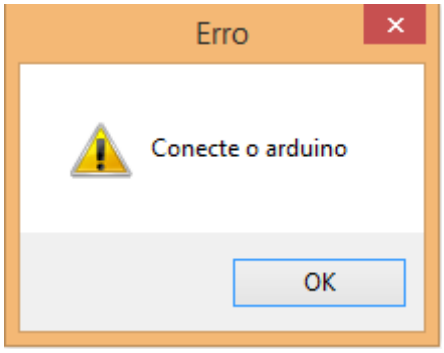


Figura 16. Erro de conexão USB.

Outro processo que teve de ser verificado com cautela foi o de cadastro de equipamentos, que teria que verificar se o equipamento já havia sido cadastrado no patrimônio, representado no BD pela tabela "equipamentos".

Também foi desenvolvido um tratamento na função de associação para evitar entradas de equiamentos já associados ou equipamentos não cadastrados. Para tratar este erro, fez-se com que o usuário não tenha que escrever o número do patrimônio, mas sim escolhe-lo a partir de uma lista que contém somente equipamentos sem associação ativa e cadastrados no patrimônio(Figura 17).

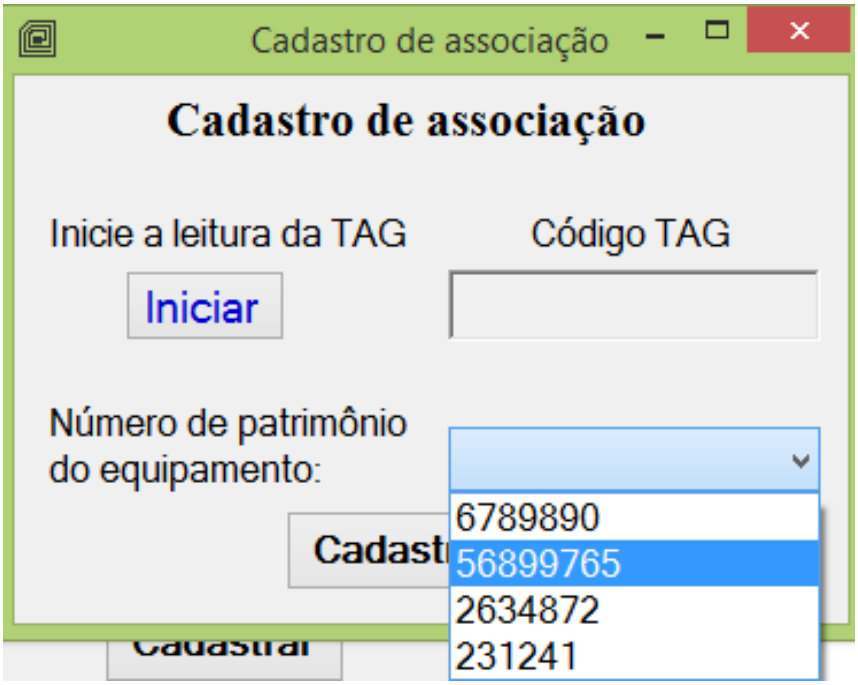


Figura 17. Tratamento de erro associação.

5 CONCLUSÕES

Com base neste trabalho podemos verificar a possibilidade de implementação de um sistema de monitoramento e gerenciamento de ativos hospitalares de pequeno porte com base no uso de TAG's RFID. Este tipo de controle se mostra eficaz, uma vez que o sistema pode gerar relatórios com facilidade, não exigindo muito do funcionário que lidará com entradas e saídas de equipamentos.

O sistema pode facilmente ser implantado em bombas de infusão enterais, bombas de infusão parenterais e esfigmomanômetros e estes são os equipamentos recomendados para se dar início ao uso da tecnologia em uma EAS. O maior obstáculo a se cumprir com a implementação deste seria a resistência do corpo de funcionários que operarão o sistema. Uma vez que qualquer mudança de rotina de trabalho causa desconforto de funcionários e resistência a tecnologia, este software mostra uma propósta de interface fácil para tentar amenizar a resistência destes funcionários.

Foi realizado um tratamento de erros para limitar o erro causado pelo usuário mal treinado, ou sem treinamento. Este tratamento de erros visou blindar o banco de dados de se corromper com dados invalidos e necessitar de assistência do desenvolvedor do sistema.

A implementação do software e hardware em um EAS depende do tipo de instituição se pretende implementar esta tecnologia. Em um estabelecimento público, deve ser realizado uma protocolação melhor. Portanto seria interessante acrescentar a função de impressão de pedidos de manutenção para posterior arquivamento por parte do estabelecimento. Processo que acontece em intituições como o Hospital das Clínicas de Uberlandia.

Para estudos posteriores seria interessante desenvolver-se um sistema, baseado neste, para controle de fluxo de equipamentos dentro do estabelecimento. Bombas de infusão comumente operam sob contrato comodato, portanto não teriam número de patrimônio, portanto o controle pode ser realizado por meio da tecnologia RFID. Com a existência de uma central de equipamentos dentro do estabelecimento fica fácil o controle da distribuição de equipamentos sob este tipo de contrato por meio do seu histórico neste sistema.

Conclui-se portanto que este trabalho atingiu seus objetivos e mostrou ser possível a realização de um sistema de controle de equipamentos hospitalares utilizando a tecnologia RFID para identificação destes.

6 REFERÊNCIAS

[1] Biblioteca comunicação MFRC522. Disponível em <http://playground.arduino.cc/Learning/MFRC522> Acesso em 21 de ago. 2014.

[2] CALIL, Saide Jorge. Gerenciamento de manutenção de equipamentos hospitalares. In: Saúde & Cidadania. Instituto para o Desenvolvimento da Saúde/Universidade de São Paulo. Faculdade de Saúde Pública. Núcleo de Assistência Médico-Hospitalar/Banco Itaú, 1998.

[3] COHEN, Ted et al. Benchmark indicators for medical equipment repair and maintenance. Biomedical instrumentation & technology/Association for the Advancement of Medical Instrumentation, v. 29, n. 4, p. 308-321, 1994.

[4] CHEN, Peter Pin-Shan. The entity-relationship model—toward a unified view of data. ACM Transactions on Database Systems (TODS), v. 1, n. 1, p. 9-36, 1976.

[5] Datasheet MF1S503x. Disponível em <http://www.nxp.com/documents/data_sheet/MF1S503x.pdf>. Acesso em 21 de ago. 2014.

[6] Datasheet MFRC522. Disponível em < http://www.nxp.com/documents/data_sheet/MFRC522.pdf>. Acesso em 21 de ago. 2014.

[7] Instalador do MySQL. Disponível em < http://dev.mysql.com/downloads/installer/>. Acesso em 21 de ago. 2014.

[8]. ISO/IEC 14443 – 1 Características físicas de sistemas RFID. Disponível em <http://www.iso.org/iso/iso_catalogue/catalogue_ics/catalogue_detail_ics.htm?csnumber=39693> . Acesso em 21 de ago. 2014.

[9]. ISO/IEC 14443 – 2 Características de sinal e de interface de sistemas RFID. Disponível em <http://www.iso.org/iso/home/store/catalogue_tc/catalogue_detail.htm?csnumber=50941>. Acesso em 21 de ago. 2014.

[10]. ISO/IEC 14443 – 3 Inicialização de anti colisão de sistemas RFID. Disponível em<http://www.iso.org/iso/home/store/catalogue_tc/catalogue_detail.htm?csnumber=50942>. Acesso em 21 de ago. 2014.

[11]. ISO/IEC 14443 – 4 Protocolos de comunicação de sistemas RFID. Disponível em<http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=50648>. Acesso em 21 de ago. 2014.

[12]. MUASSAB, José Roberto. Gerenciamento da manutenção na indústria automobilística. 2002. 98 f. Dissertação (especialização pelo curso MBA – Gerência de Produção) – Universidade de Taubaté, 2002.

[13]. Referência para parcela de mercado de sistemas operacionais. < http://marketshare.hitslink.com/>. Acesso em 29 de outubro.2014.

[14]. Referência para pesquisa sobre sintaxe em Arduino. Disponível em< http://arduino.cc/en/Reference/HomePage>. Acesso em 21 de ago. 2014.

[15]. Referência para pesquisa sobre sintaxe em C#. Disponível em <http://msdn.microsoft.com/en- us/library/67ef8sbd.aspx>. Acesso em 21 de ago. 2014.

[16]. Referência para pesquisa sobre sintaxe em Mysql. Disponível em< http://dev.mysql.com/doc/refman/5.6/en/>. Acesso em 21 de ago. 2014.

[17]. Referência sobre arduino MEGA. Disponível em< http://arduino.cc/en/Main/arduinoBoardMega2560>Acesso em 19 de outubro. 2014.

[18]. RAUSAND, Marvin. Reliability centered maintenance. Reliability Engineering & System Safety, v. 60, n. 2, p. 121-132, 1998.

[19]. XAVIER, Júlio Nascif. Manutenção–Tipos e Tendências. 2000.