

UNIVERSIDADE DO EXTREMO SUL CATARINENSE – UNESC

CURSO DE CIÊNCIA DA COMPUTAÇÃO

IGOR HENRIQUE PEREIRA FONTANA

**DESENVOLVIMENTO DE SISTEMA PARA CONTROLE DE HORÁRIOS DE
FUNCIONÁRIOS DOMÉSTICOS COM IDENTIFICAÇÃO POR
RÁDIOFREQUÊNCIA**

CRICIÚMA

2014

IGOR HENRIQUE PEREIRA FONTANA

**DESENVOLVIMENTO DE SISTEMA PARA CONTROLE DE HORÁRIOS DE
FUNCIONÁRIOS DOMÉSTICOS COM IDENTIFICAÇÃO POR
RÁDIOFREQUÊNCIA**

Trabalho de Conclusão de Curso, apresentado para obtenção do grau de Bacharel no curso de Ciência da Computação da Universidade do Extremo Sul Catarinense, UNESC.

Orientador: Prof. Esp. Sérgio Coral

CRICIÚMA

2014

IGOR HENRIQUE PEREIRA FONTANA

**DESENVOLVIMENTO DE SISTEMA PARA CONTROLE DE HORÁRIOS DE
FUNCIONÁRIOS DOMÉSTICOS COM IDENTIFICAÇÃO POR
RÁDIOFREQUÊNCIA**

Trabalho de Conclusão de Curso aprovado pela Banca Examinadora para obtenção do Grau de Bacharel no Curso de Ciência da Computação da Universidade do Extremo Sul Catarinense, UNESC, com Linha de Pesquisa em Sistemas Embarcados.

Criciúma, 24 de junho de 2014.

BANCA EXAMINADORA

Prof. Esp.  - Orientador - (UNESC)

Prof. MSc.  - (UNESC)

Prof. Esp.  - (UNESC)

RESUMO

Com a introdução de novas legislações para o empregado doméstico, novas obrigações surgiram com intuito de igualar os direitos entre os outros tipos de trabalhadores, transformando este emprego em um que também possa ser fiscalizado pelo empregador. Existem algumas formas de realizar o controle, porém a que oferece maior facilidade de controle é usando o meio eletrônico, e desta forma existem equipamentos para servirem de sistema eletrônico para o registro de ponto de empregados, contudo, seu preço pode ser alto e inviável para o uso de controle de empregados domésticos. Este trabalho possui o objetivo de construir um sistema eletrônico de forma a viabilizar a técnica para este tipo de uso específico. Há inúmeras maneiras de construir um sistema como este. Neste trabalho será abordado o uso da tecnologia de identificação por rádio frequência para conseguir distinguir cada empregado doméstico e oferecer facilidade de uso do sistema, e para o recolhimento de dados, processamento e base de dados será estudado a plataforma Arduino, que pode fornecer grande mobilidade devido ao seu tamanho como um microcontrolador e também seu baixo consumo de energia. Para que as informações sejam inseridas no Arduino com maior facilidade, é necessário um software gestor que comunicará através de uma rede local, este foi desenvolvido utilizando o Visual Studio da Microsoft.

Palavras-chave: Empregado doméstico. Identificação por radiofrequência. Sistema para controle de ponto. Arduino. Visual Studio.

ABSTRACT

With the introduction of new laws for domestic workers, new obligations emerged in order to level the rights for other types of workers, turning this kind of job into one that can also be supervised by the employer. There are some ways to perform this kind of control, but the easier one is by electronic means, even though it might be impracticable due to its high costs. This work aims to build an electronic system for this specific scenario. There are many ways of building a system like this. In this work we will address the use of radio frequency technology for identification in order to be able to distinguish each home employee and provide an easier use of the system. For data collection and processing, the Arduino platform was chosen due to its small size and low energy consumption. Additionally, a management software developed in Microsoft's Visual Studio will be used to input information into the Arduino platform, communicating through the local network.

Keywords: Domestic worker. Radio-frequency identification. Time clock system. Arduino. Visual Studio.

LISTA DE ILUSTRAÇÕES

Figura 1 – Exemplar de um Registrador Eletrônico de Ponto.....	17
Figura 2 – Modelo básico de um microprocessador.....	20
Figura 3 – Modelo básico de microcontrolador.....	21
Figura 4 – Modelo simplificado de um microcontrolador AVR.....	23
Figura 5 – Os possíveis modelos de diferentes Arduinos	27
Figura 6 – Um exemplar oficial do Arduino	27
Figura 7 – Adaptador para fonte externa de energia.....	29
Figura 8 – Ambiente de Desenvolvimento para Arduino	31
Figura 9 – Shield Ethernet para Arduino	32
Figura 10 – Modelo simplificado do sistema RFID	34
Figura 11 – Um leitor (preto), antena (branco) e 3 tipos de tags.....	35
Figura 12 – Os blocos básicos de um sistema RFID.....	36
Figura 13 – Modelo de uma tag, ou transponder.....	37
Figura 14 – Variedades de Tags	38
Figura 15 – Exemplo de funcionamento dos leitores.....	39
Figura 16 – Classificação de frequências.....	40
Figura 17 – Relação de frequências.....	42
Figura 18 – Visual Studio Professional.....	44
Figura 19 – Wireshark	45
Figura 20 – Kit de Arduino para iniciantes.....	49
Figura 21 – Módulo RFID-RC522.....	51
Figura 22 – Ethernet Shield acoplado em um Arduino UNO	52
Figura 23 – Sistema de ligação para barramento SPI.....	53
Figura 24 – Módulo RFID/NFC ElecFreaks	54
Figura 25 – Display LCD I2C.....	55
Figura 26 – Módulo RTC I2C.....	56
Figura 27 – Arduino MEGA	57
Figura 28 – Modelo do sistema	59
Figura 29 – Detecção da rede utilizando o Wireshark.....	61
Figura 30 – Classe exemplar de uma pessoa	62
Figura 31 – Instância da classe pessoa	62
Figura 32 – Objeto serializado em JSON	63

Figura 33 – Objeto serializado com método próprio.....	63
Figura 34 – Pacote para a comunicação de rede.....	64

LISTA DE TABELAS

Tabela 1 – Preços dos componentes utilizados no projeto final.....	66
Tabela 2 – Preços de componentes extras para aprendizado	67

LISTA DE ABREVIATURAS E SIGLAS

AC	Alternating Current
ADC	Analog-to-digital Converter
AFD	Arquivo-Fonte de Dados
CAN	Controller Area Network
CBMPA	Corpo de Bombeiros Militar do Pará
CEI	Cadastro Específico do INSS
CPU	Central Processing Unit
DC	Direct Current
EDM	Electrical Discharge Machine
EEPROM	Electrically Programmable Read-OnlyMemory
GHz	Gigahertz
GPS	Global Position System
HF	High Frequency
I2C	Inter-Integrated Circuit
IDE	Integrated Development Environment
ISM	Industrial Scientific and Medical
kHz	Kilohertz
LCD	Liquid Crystal Display
LED	Diodo Emissor de Luz
LF	Low Frequency
MF	Medium Frequency
MHz	Megahertz
MIT	Massachusetts Institute of Technology
MRP	Memória de Registro de Ponto
KB	Kilobyte
MB	Megabyte
MT	Memória de Trabalho
PIC	Peripheral Interface Controller
PWM	Pulse-width Modulation
RAM	Random Access Memory
REP	Registrador Eletrônico de Ponto
RFID	Radio-Frequency Identification

RISC	Reduced Instruction Set Computing
ROM	Read Only Memory
SPI	Serial Peripheral Interface
SRAM	Static Random Access Memory
SREP	Sistema de Registro Eletrônico de Ponto
UHF	Ultra High Frequency
USART	Universal Synchronous Asynchronous Receiver Transmitter
USB	Universal Serial Bus
VHF	Very High Frequency
WiFi	Wireless Fidelity

SUMÁRIO

1 INTRODUÇÃO	11
1.1 OBJETIVO GERAL	12
1.2 OBJETIVO ESPECÍFICOS.....	12
1.3 JUSTIFICATIVA	13
1.4 ESTRUTURA DO TRABALHO.....	14
2 FUNDAMENTAÇÃO TEÓRICA	15
2.1 LEGISLAÇÃO E LEI.....	15
2.1.1 Lei do Emprego Doméstico	15
2.1.2 Lei da Jornada de Trabalho	15
2.1.3 Sistema de Registro Eletrônico de Ponto	16
2.1.3.1 Retratação do Sistema de Registro Eletrônico de Ponto.....	16
2.1.3.2 Registrador Eletrônico de Ponto.....	16
2.1.3.2.1 <i>Requisitos do Registrador Eletrônico de Ponto</i>	17
2.1.3.2.2 <i>Funcionalidades do REP</i>	18
2.2 MICROCONTROLADOR.....	20
2.2.1 Conhecimentos Fundamentais	20
2.2.2 Usabilidade	21
2.2.3 Microcontrolador AVR	22
2.2.3.1 Portas de entradas e saídas.....	24
2.2.3.2 Memória EEPROM e SRAM.....	24
2.2.3.3 Dados de RAM	24
2.3 ARDUINO.....	25
2.3.1 Vantagens	25
2.3.2 Conhecimentos Básicos	27
2.3.4 Programação	29
2.3.4.1 Integrated Development Environment	30
2.3.5 Shields	31
2.4 IDENTIFICAÇÃO POR RADIOFREQUÊNCIA	32
2.4.1 Componentes	33
2.4.2 Tag	36
2.4.3 Leitores	38
2.4.4 Caracterização	39

2.4.4.1	Frequência, distância e acoplamento.....	40
2.4.5	Regulamentação.....	42
2.5	SOFTWARES DE APOIO PARA DESENVOLVIMENTO	43
2.5.1	Visual Studio e o .NET Framework	43
2.5.2	Wireshark.....	45
3	TRABALHOS CORRELATOS.....	47
3.1	RFID: ANÁLISE DA VIABILIDADE, VANTAGENS E DESVANTAGENS	47
	DA TECNOLOGIA E DESENVOLVIMENTO DE UM SISTEMA PARA.....	47
	DEMONSTRAÇÃO E TESTES	47
3.2	CONTROLE DE PRESENÇA UTILIZANDO RFID: UM ESTUDO DE CASO UTILIZANDO A LINGUAGEM RUBY	47
3.3	CONTROLE DO FLUXO DE PESSOAS USANDO RFID.....	48
4	DESENVOLVIMENTO DO SISTEMA	49
4.1	METODOLOGIA.....	49
4.1.1	Aprendizado e compreensão do Arduino	49
4.1.2	Seleção do Hardware necessário	50
4.1.3	Barramentos de comunicações utilizadas pelo Arduino	53
4.1.4	Seleção dos componentes para uso do barramento I2C.....	54
4.1.5	Mudança do Arduino para versão MEGA e testes necessários	56
4.1.6	Desenvolvimento do código do Arduino e seu gestor para Windows	59
4.1.6.1	Organização de código.....	59
4.1.6.1.1	<i>Estrutura para o código do Arduino</i>	<i>59</i>
4.1.6.1.2	<i>Estrutura para o código do gestor</i>	<i>60</i>
4.1.6.2	Definição da metodologia de desenvolvimento de código.....	60
4.1.6.3	Transmissão dos dados entre o gestor e o Arduino	61
4.1.6.4	Persistência dos dados para o Arduino	65
4.1.7	Materiais utilizados	66
4.2	RESULTADOS OBTIDOS.....	68
5	CONCLUSÃO	69
5.1	TRABALHOS FUTUROS	69

1 INTRODUÇÃO

A alteração da lei que promove os direitos trabalhistas permitiu que o empregado doméstico passe a ter novos direitos (BRASIL, 2013a). Desta forma, um trabalhador doméstico tem os direitos iguais aos outros trabalhadores. Os empregadores também têm direitos iguais, e um desses direitos é o de fiscalizar e assegurar que o trabalhador esteja cumprindo seus deveres e tal como pode cobrar os horários (BRASIL, 1989). Com a ideia de tornar a jornada de trabalho do trabalhador doméstico fiscalizada, a tecnologia permite que de forma automatizada, sua jornada seja acompanhada de forma eletrônica por meio de um sistema eletrônico para registros de pontos.

Existem várias formas de construir um sistema para o registro de pontos, principalmente para a forma de como é identificado o empregado, podendo ser por senha, leitura biográfica, identificação por radiofrequência, entre outros. A tecnologia de identificação por radiofrequência surgiu para um propósito similar ao do código de barras, porém pode ser aplicado em inúmeras ideias compatíveis.

Identificação por rádio frequência é um sistema que permite a comunicação de dados sem fios que permite os usuários identificar objetos ou seres vivos que terão etiquetas únicas (HUNT, 2007, tradução nossa). Podendo por exemplo, em um supermercado, saber quais produtos o cliente tem em seu carrinho de compras, e automaticamente mostrar o resultado da compra realizando uma leitura de todos os produtos ali encontrados via identificação por rádio frequência. Em outros casos, ter uma etiqueta em um relógio de pulso para a identificação em um local restrito. (FINKENZELLER, 2010, tradução nossa).

Segundo Finkenzeller (2010, tradução nossa), um sistema de identificação por rádio frequência necessita de um controlador para consumir as informações lidas pelo sistema, pois o mesmo apenas realiza as leituras das etiquetas e entrega para alguma aplicação responsável.

O responsável pode ser um microcontrolador, que segundo Souza (2005), fazem parte de um sistema embarcado e preenchem a parte de inteligência programável no sistema. São encontrados em vários equipamentos e normalmente usado para construir qualquer tipo de equipamento que necessita uma grande

quantidade de leitura de dados e não necessita de um grande desempenho em processamento e ao mesmo tempo precisa ocupar pouco espaço, pois, seu tamanho é normalmente pequeno, sua especialidade é ter entradas de dados de vários tipos de eletrônicos, e ao mesmo tempo podendo realizar saídas de dados e de energia, um exemplo básico seria ao apertar um botão, o microcontrolador acender um Diodo Emissor de Luz (LED).

Segundo Banzi (2011, tradução nossa), o funcionamento de componentes eletrônicos e suas teorias são de difícil entendimento, o que levaria muito tempo para compreendê-las e como utilizá-las. Com um simples kit de desenvolvimento de circuitos, é possível realizar trabalhos e protótipos com maior facilidade. Por esses motivos, e muitos outros, Arduino foi construído como um projeto aberto que nada mais é do que uma plataforma de computação física baseada em um sistema simples de entrada e saída em uma placa.

No Arduino, é possível conectar uma variedade de componentes que se comunicam com microcontroladores, e desta forma, existem leitores próprios para identificação de rádio frequência que tem total compatibilidade com o Arduino.

1.1 OBJETIVO GERAL

Desenvolver um sistema para registro eletrônico de ponto, utilizando Arduino e seus tipos de recursos. Sendo possível fornecer a gerência de horários de funcionários domésticos, com um custo considerável que permita a prática para empregadores que possuem serviços domésticos nos seus lares.

1.2 OBJETIVO ESPECÍFICOS

Os objetivos específicos deste trabalho são os seguintes:

- a) estudar o funcionamento e programação do Arduino;
- b) estudar o Visual Studio e .NET Framework;
- c) compreender o funcionamento do sistema de *Radio-Frequency Identification* (RFID);
- d) aplicar um sistema RFID para ser usado em Arduino;
- e) desenvolver um Sistema de Registro Eletrônico de Ponto (SREP);

- f) desenvolver um software desktop como gestor para o sistema REP e gerar relatórios, utilizando o Visual Studio;

1.3 JUSTIFICATIVA

O desenvolvimento de sistemas usando a plataforma Arduino oferece uma facilidade enorme, tanto para iniciantes, ou para pessoas experientes, pois é justamente uma plataforma que é construída para tornar o processo de prototipagem de sistemas mais rápido e para proporcionar uma maior facilidade para o aprendizado de eletrônica para iniciantes (BANZI, 2011, tradução nossa). Arduino, como uma plataforma *open-source*, é um dos mais utilizados ultimamente para projetos caseiros, e também projetos acadêmicos. Isso mostra que com o uso do Arduino, grandes conhecimentos podem ser alcançados com prática de desenvolvimento.

RFID, como um sistema de identificação única, é possível utiliza-lo em inúmeras aplicações, em estacionamentos para computação do tempo permanecido, em supermercados para identificar todos os itens em um carrinho de compras (IGOE, 2012). Esse tipo de sistema mostra facilidade de aplicá-lo, e uma enorme comodidade para as pessoas que o utilizam, pois ao se tratar de uma pequena etiqueta que identifica os objetivos, ela pode ser usada praticamente em qualquer lugar em qualquer formato (FINKENZELLER, 2010, tradução nossa).

O Arduino possui uma capacidade de receber *shields* acoplados para terem um funcionamento adicional, esses *shields* possuem várias finalidades, e alguns deles possuem a funcionalidade de atuarem como leitores de RFID. Esses componentes extras, normalmente, oferecem maior facilidade do que componentes comuns voltados à microcontroladores em geral, pois para o Arduino, existem bibliotecas prontas que facilitam a programação do sistema para o manuseio das informações de RFID (IGOE, 2012, tradução nossa).

Com um sistema RFID aplicado ao Arduino, é possível desenvolver um sistema que será capaz de realizar registros de entradas e saídas, bem como armazená-las em uma memória permanente e com um computador será possível realizar relatórios diários e também mensais, fornecendo uma maneira para efetuar

pagamento do funcionário. O sistema deve ser simples e proporcionar facilidade para os funcionários domésticos, deve ser desenvolvido desta forma para evitar problemas com pessoas que possuem dificuldades com equipamentos e tecnologias recentes.

1.4 ESTRUTURA DO TRABALHO

Este trabalho é dividido em cinco capítulos, há no primeiro uma introdução ao trabalho realizado, contendo informações sobre alguns tópicos usados para seu desenvolvimento, os objetivos que foram esperados alcançar e uma breve justificativa da escolha do trabalho e o seu tema.

No segundo capítulo há explicações e esclarecimentos sobre as principais tecnologias, componentes que foram utilizados para o desenvolvimento do sistema deste trabalho e explicações sobre as legislações que controlam a distribuição de um sistema deste tipo.

Alguns trabalhos correlatos a este foram postos no terceiro capítulo, desta forma possibilitando ter uma visibilidade do tema em outras pesquisas já realizadas.

O desenvolvimento do trabalho é encontrado no quarto capítulo, destaca-se neste todo o processo da construção do sistema, desde a parte do controlador até o seu gestor. As dificuldades, problemas, soluções e testes também podem ser encontradas neste capítulo. Para concluir o capítulo, há os resultados obtidos.

No quinto capítulo, é possível ter conhecimento sobre a conclusão do projeto, apresentando quais os objetivos que foram obtidos. Há ideias de possíveis melhorias para o sistema que por se tratarem de uma dificuldade com maior grau para o trabalho, não foram praticadas. Os trabalhos futuros também são encontrados neste capítulo.

2 FUNDAMENTAÇÃO TEÓRICA

2.1 LEGISLAÇÃO E LEI

Existem várias legislações para trabalhadores, empregadores e ultimamente temos as novas normas para os empregados domésticos que até então não tinham direitos iguais aos outros trabalhadores. O conhecimento sobre as regras são importantes para o desenvolvimento do projeto, pois também existem legislações para os meios de fiscalizações.

2.1.1 Lei do Emprego Doméstico

No início de Abril de 2013, houve uma alteração na redação do parágrafo único do art. 7 da Constituição Federal pela emenda constitucional n.º 72. Até então, pessoas que trabalhavam em residências prestando serviços não possuíam direitos trabalhistas iguais aos outros trabalhadores urbanos e rurais. A partir da alteração, foi estabelecido essa igualdade, de que trabalhadores domésticos teriam igualdade de direitos entre os outros tipos de trabalhadores. Com a alteração, todo empregador deve fornecer os direitos para o trabalhador doméstico que agora surgiram, bem como tem seus direitos de controle e fiscalização (BRASIL, 2013a).

Segundo a lei nº 5.859, os empregados domésticos são aqueles que prestam serviços de natureza não-econômica para uma pessoa ou uma família, e estes serviços são do tipo residencial (BRASIL,1972). Conforme os termos, são exemplos de ocupações dos empregados domésticos: empregado, cozinheiro, governanta, babá, lavadeira, faxineiro, vigia, motorista particular, jardineiro, acompanhante de idosos, dentre outros (BRASIL, 2013b).

2.1.2 Lei da Jornada de Trabalho

Segundo a lei nº 7.855, a jornada de trabalho pode ser fiscalizada por meios de registros manuais, mecânicos ou eletrônicos, porém, as instruções dos métodos utilizados devem ser expedidos pelo Ministério do Trabalho. Contudo, o

principal é ter anotação de cada entrada e saída. Para estabelecimentos com mais de dez trabalhadores, algum dos métodos citados devem ser obrigatoriamente implantados para uso (BRASIL, 1989).

Conforme a lei, o método eletrônico é possível ser usado para controlar a jornada de trabalho dos colaboradores, o método pode ser usado em qualquer ocasião. Em um exemplo, é possível ter um empregado doméstico sem ter sua jornada de trabalho fiscalizada, já que a casa apenas teria este único colaborador, contudo, fica a disponibilização do uso do método.

2.1.3 Sistema de Registro Eletrônico de Ponto

Nesta seção do trabalho utilizou-se em sua íntegra a fonte bibliográfica Brasil (2009) que refere-se a Portaria 1.510, documento publicado em 2009 pelo Ministério do Trabalho, que retrata o que é o Sistema de Registro Eletrônico de Ponto (SREP), qual sua aplicação, função e deveres, bem como suas normas de funcionamento.

2.1.3.1 Retratação do Sistema de Registro Eletrônico de Ponto

A Portaria publicada retrata que SREP, é um sistema embarcado, ou seja, é formado por um conjunto de equipamentos e programas, que tem funcionalidade de gravar todas as entradas e saídas das jornadas dos trabalhadores das empresas. Capacitado para emitir documentos fiscais e assim realizar fiscalizações daqueles que usam para registrar seus horários.

2.1.3.2 Registrador Eletrônico de Ponto

O Registrador Eletrônico de Ponto (REP) é um equipamento (figura 1) que pertence ao SREP, e é obrigatório como parte do sistema, que nada mais é do o responsável pela automação do sistema. É construído exclusivamente para o registro de jornada de trabalho, com capacidade de emitir os documentos fiscais e

realizar qualquer fiscalização necessária. É proibido usar quaisquer outros meios de registros para a utilização do SREP.

Figura 1 – Exemplar de um Registrador Eletrônico de Ponto



Fonte: Ideia Consultoria & Assessoria (2013).

2.1.3.2.1 Requisitos do Registrador Eletrônico de Ponto

Para a Portaria, o REP deve apresentar algumas características pré-estabelecidas como requisitos:

- a) um relógio interno de tempo real com precisão mínima de um minuto por ano, com capacidade de funcionar por no mínimo mil quatrocentos e quarenta horas;
- b) um visor mostrando hora, minutos e segundos;
- c) disponibilização de um mecanismo impressor em bobina de papel, com durabilidade mínima de cinco anos;
- d) memória para armazenamento permanente, que pela portaria é denominada de Memória de Registro de Ponto (MRP), onde os dados não podem ser apagados ou alterados;
- e) meio de armazenamento, denominado pela portaria de Memória de Trabalho (MT), onde será usada para os dados necessários à operação do REP;

- f) uma porta Universal Serial Bus (USB) externa, denominada de Porta Fiscal, para realização da leitura dos dados da MRP;
- g) quando a função de marcação de ponto for realizada, o REP não deverá depender de qualquer conexão com outro equipamento externo;
- h) não poderá ser realizado marcações de ponto quando o REP precisar se comunicar com outro equipamento, seja para carga ou leitura de dados.

A memória MT possui dados do empregador e dos empregados. São dados referentes a identificação de cada um como por exemplo nome, CNPJ ou CPF, CEI, razão social, PIS, entre outros.

Já na memória MRP, quando houver a inclusão ou alteração das informações do empregador na MT, deve conter a data e hora da inclusão ou alteração e outros dados que possam identificar o tipo de operação, e identificar o empregador. Devem ser gravados também, a marcação de ponto com número de PIS, data e hora da marcação, quando um ajuste do relógio interno acontecer, deve possuir dados para a data e hora antes do ajuste, a data e hora ajustada. Quando houver inserção, alteração e exclusão de dados de empregado na MT, deve conter a data e hora da operação, tipo de operação, número do PIS e nome do empregado.

2.1.3.2.2 Funcionalidades do REP

Em sua memória, é armazenado todas as informações referentes ao empregador e também de seus funcionários. Com as informações, o REP deve ter habilidades de marcar pontos dos funcionários com alguns passos pré-estabelecidos, respectivamente, são eles: receber a identificação do trabalhador sem necessidade de outro aparelho, obter a hora do relógio do sistema em tempo real, registrar a marcação de ponto na memória, imprimir o comprovante do trabalhador.

Ele também deve ser capaz de gerar um arquivo a partir da memória de registro de ponto para fiscalizações posteriores e exporta-lo para um dispositivo externo com memória por meio de uma porta denominada Porta Fiscal. O Arquivo-Fonte de Dados (AFD) é um dos arquivos que são gerados a partir dos dados na

MRP e deve ser possível a gravação desse arquivo em um dispositivo externo de memória, utilizando a Porta Fiscal.

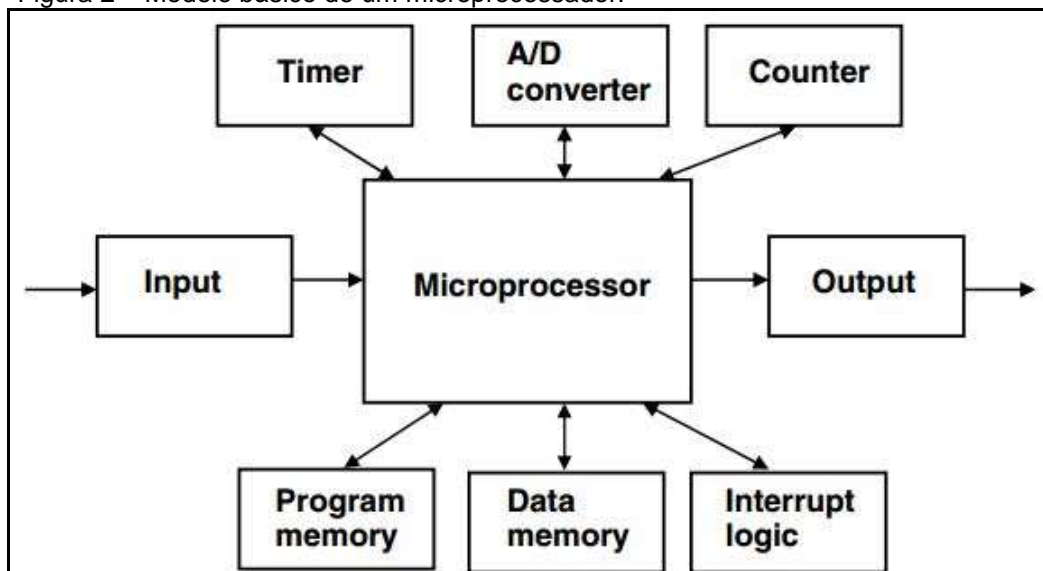
Considerando os requisitos e funcionalidades de um REP, ele é um dispositivo que requer processamento de informações. Uma maneira para que tenha um dispositivo móvel com um custo acessível para o processamento dessas informações é usar microcontroladores para realizar todos os processos.

2.2 MICROCONTROLADOR

2.2.1 Conhecimentos Fundamentais

Microcontroladores são, normalmente, pequenos, tendo sua função como componente eletrônico em um sistema. Dentro deste pequeno chip, podemos encontrar uma certa inteligência que é programável, e então, usá-la para controlar periféricos como: LED, botões, displays de segmentos, displays de cristal líquido, resistências, relês, sensores diversos e muitas outras coisas, esse controle de componentes eletrônicos por meio do microcontrolador são chamados de controles lógicos, justamente por as operações programáveis de dentro do microcontrolador, aquela parte inteligente, sofrer influência no estado dos periféricos que ali são colocados juntamente ao microcontrolador (SOUZA, 2005).

Figura 2 – Modelo básico de um microprocessador.



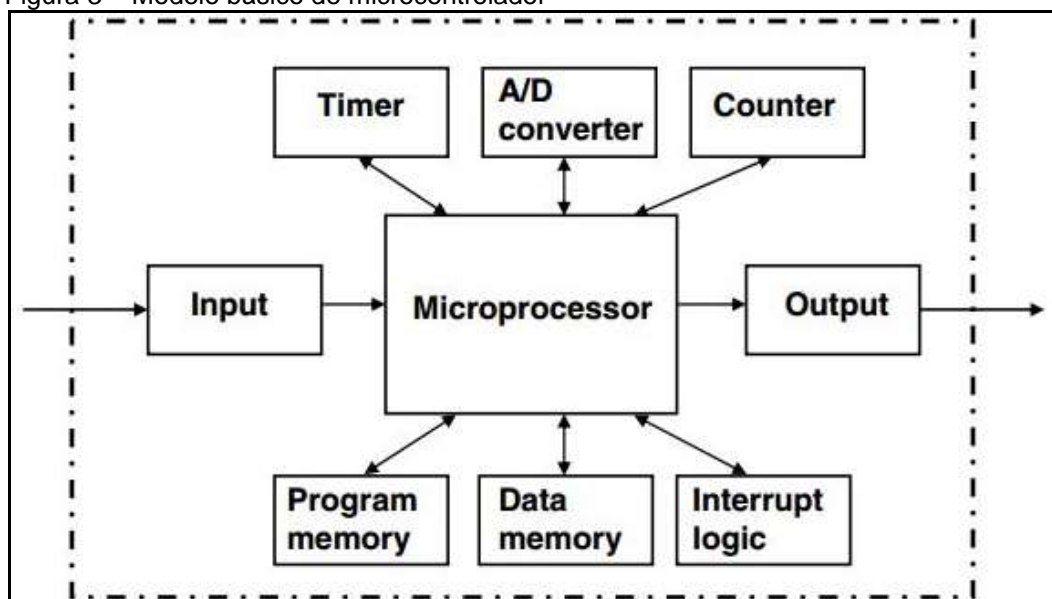
Fonte: Ibrahim (2012).

Na figura 2, encontra-se o microprocessador que é independente dos módulos de entrada de dados, da memória de programa, da memória de dados, entre outros módulos.

Os microcontroladores (figura 3) são bastante parecidos com os processadores que existem dentro dos computadores pessoais, porém tem

diferenças sutis que fazem eles terem aplicações diferentes. Um processador de computador requer vários outros componentes para tornar o computador usável, em outro lado, o microcontrolador é um chip único e independente com baixo consumo de energia. Uma vantagem que o microcontrolador não possui perante o processador é sua capacidade de aumentar seu poder para ter mais memória, ou sua quantidade de entradas e saídas de dados, o microcontrolador, por ser um chip fechado, para se obter mais memória e possuir mais entradas e saídas de dados, seria necessário adquirir outro chip microcontrolador (IBRAHIM, 2012, tradução nossa).

Figura 3 – Modelo básico de microcontrolador



Fonte: Ibrahim (2012).

2.2.2 Usabilidade

Os microprocessadores tem sua vantagem que é ser versátil, deixando a possibilidade de decidir a quantidade de *Random Access Memory* (RAM), *Read Only Memory* (ROM), e também quantas portas de entradas e saídas para se encaixar a tarefa que será designada (MAZIDI, 2010, tradução nossa).

Por suas vantagens, os microcontrolador são usados em várias aplicações, e aparecem bastante em produtos finais.

Estes chips são usados pela humanidade há algumas décadas, de várias formas estão presentes no nosso dia-a-dia. Seja em meios comerciais, industriais ou educacionais. São encontrados em impressoras, calculadoras, televisores, veículos, leitores de código de barras, vídeo games, celulares, entre outros (IBRAHIM, 2012, tradução nossa).

Mas as razões mais fortes que identificam o uso de microcontroladores como solução desses problemas, Mazidi (2012, tradução nossa) explica que o uso dos microcontroladores nesses equipamentos, é pela questão de que nem todas as aplicações de processamento levam mais em consideração o espaço usado, a consumação de energia, e o preço por unidade do que o poder de processamento.

Há vários tipos de microcontroladores que podem ser encontrados, porém, há cinco bastante usados. São eles, Freescale Semicondutor 68HC08/68HC11, Intel 8051, Atmel AVR, Zilog Z8 e *Peripheral Interface Controller* (PIC) da Microchip Technology. Cada um desses é diferente do outro, eles tem instruções únicas que diferem entre si, diante disso, é possível saber que eles não são compatíveis entre eles (MAZIDI, 2010, tradução nossa). Desses todos mencionados, o microprocessador usado no Arduino é o AVR desenvolvido pela Atmel (SCHMIDT, 2011, tradução nossa).

2.2.3 Microcontrolador AVR

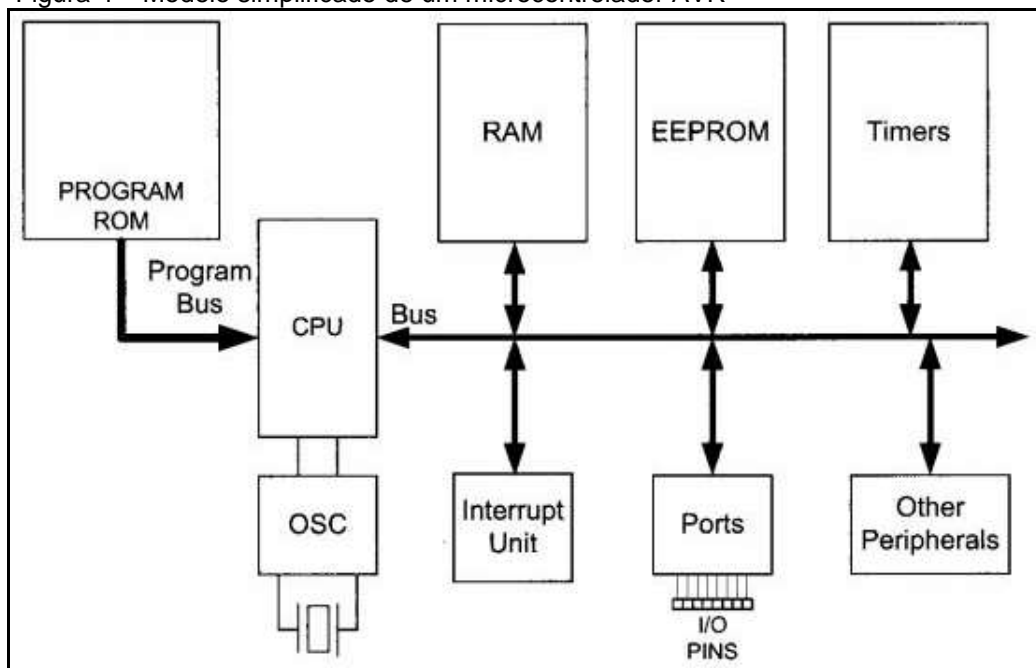
A arquitetura básica do AVR (figura 4) foi desenvolvida por dois estudantes do Instituto Norueguês de Tecnologia. Os AVR usam microprocessadores de 8 *bits*, isso significa que computam 8 *bits* de dados por vez. Dados maiores que 8 *bits* são forçados a serem quebrados em partes de 8 *bits* para serem processados. Um dos problemas que possuem, é devido sua compatibilidade. Os códigos escritos para um modelo de AVR necessita ser recompilado para um outro modelo de AVR (MAZIDI, 2010, tradução nossa).

Em geral, um AVR é um chip microcontrolador de 8 *bits* do tipo *Reduced Instruction Set Computer* (RISC) que possui uma arquitetura Harvard. A maioria dos AVR possui alguns aspectos padrões, como programas diretamente no chip ROM, dados em RAM, dados em *Electrically Programmable Read-Only Memory*

(EEPROM), temporizadores e portas de entrada e saída. Alguns AVR possuem algumas características adicionais como *Analog-to-digital Converter (ADC)*, *Pulse-width Modulation (PWM)* e diferentes interfaces seriais como *Universal Synchronous Asynchronous Receiver Transmitter (USART)*, *Serial Peripheral Interface (SPI)*, *Inter-Integrated Circuit (I2C)*, *Controller Area Network (CAN)*, USB, entre outros (MAZIDI, 2010, tradução nossa).

Na computação, memória RAM e ROM, estão relacionadas entre si, para tanto, todos os dados devem ser gravados em memória RAM ou ROM. A função básica da ROM é guardar informações que são permanentes ou fixos. Já a memória RAM, é uma memória que leva ao fato de guardar informações que podem mudar com o tempo (MAZIDI, 2010, tradução nossa).

Figura 4 – Modelo simplificado de um microcontrolador AVR



Fonte: Mazidi (2010).

Na figura 4, é possível localizar os componentes do AVR, como sua memória RAM, EEPROM, as portas de entrada e saída de dados que podem ser chamadas de portas *input/output (I/O)* e também a memória ROM reservada para o programa que será executado pela CPU.

2.2.3.1 Portas de entradas e saídas

Portas de entradas e saídas são bidirecionais. Uma porta de saída pode facilmente ser trocada para exercer o papel de uma porta de entrada, a mesma coisa pode ser feita para uma porta de entrada se tornar uma de saída. Os pinos dessas portas precisam ser configuradas antes de serem usadas, apontando a elas se precisam ser de saída ou entrada (IBRAHIM, 2012, tradução nossa).

Os AVR possuem de 3 até 86 portas de entrada e saída. Este número de portas depende do modelo de família do AVR (MAZIDI, 2010, tradução nossa).

2.2.3.2 Memória EEPROM e SRAM

Nos microcontroladores, a memória EEPROM é usada para guardar os programas escritos para eles, que são designados a serem executados por eles mesmos, então, por essa razão, algumas vezes é chamado também de programa ou código ROM. Nos AVR essa memória pode ter até 8MB, porém nem todos os modelos vem com toda essa memória instalada no chip. Seu tamanho varia de 1KB até 256KB, dependendo da família. Os AVR foram um dos primeiros microcontroladores a usar memória Flash em seu chip para o armazenamento do programa. Esse tipo de memória é mais rápida do que os outros tipos de memórias até então usadas, pois seu conteúdo pode ser substituído mais rapidamente. O principal fator que determina esse tipo de memória ser utilizada para o armazenamento do programa é por causa de sua capacidade de não perder os dados quando o microcontrolador não estiver sendo alimentado por uma fonte externa de energia elétrica (MAZIDI, 2010, tradução nossa).

Enquanto a EEPROM é usada para armazenar os dados que raramente são modificados e não se apagarão enquanto não estiver alimentado de energia, a memória *Static Random Access Memory* (SRAM) é usada para armazenar dados e parâmetros que são mudados frequentemente (MAZIDI, 2010, tradução nossa).

2.2.3.3 Dados de RAM

Esta memória é designada para o armazenamento geral para dados. Os AVR tem no máximo 64K bytes de espaço para esses dados. Ela é disposta para guardar dados de propósitos gerais de registros, memória das portas de entradas e saídas, entre outros (MAZIDI, 2010, tradução nossa).

Os microcontroladores em geral, podem ser encontrados em produtos e projetos comerciais e também projetos de desenvolvimento livre. Arduino é um destes projetos livres que se beneficia de um microcontrolador em sua plataforma, com o intuito de oferecer uma maneira diferente de desenvolvimento, que em comparação ao uso direto do microcontrolador, é mais intuitivo e fácil.

2.3 ARDUINO

Uma placa Arduino é um pequeno computador que é possível programá-lo para processar entradas e saídas de dados entre dispositivos e componentes externos que é conectado à placa. Arduino é conhecido como uma plataforma em sistemas embarcados. Sendo possível usá-lo para ligar uma lâmpada por um período de tempo, ou após pressionar um botão. Este mesmo conceito poderia ser estendido para algo ainda maior e fazê-lo com que detecte a presença de alguém em uma sala, e fazer com que a lâmpada dessa sala seja acessa pelo sistema (BANZI, 2011, tradução nossa).

Em alguns sistemas, é possível encontrar desenvolvimentos de interação entre computadores conectados, seja pela rede, ou até mesmo pela internet para trocar dados, e realizar processos com esses dados. Alguns exemplos de objetos que podem ser conectados ao Arduino são LEDs, botões, interruptores, motores, sensores de temperatura, sensores de pressão, sensores de distância, *Global Position System* (GPS), *Ethernet* ou módulos WiFi, quase qualquer objeto que tenha dados de saída pode ser controlado (MCROBERTS, 2013, tradução nossa).

2.3.1 Vantagens

Arduino é um projeto aberto para desenvolvedores, consiste em oferecer uma plataforma de fácil desenvolvimento sobre um sistema simples baseado em

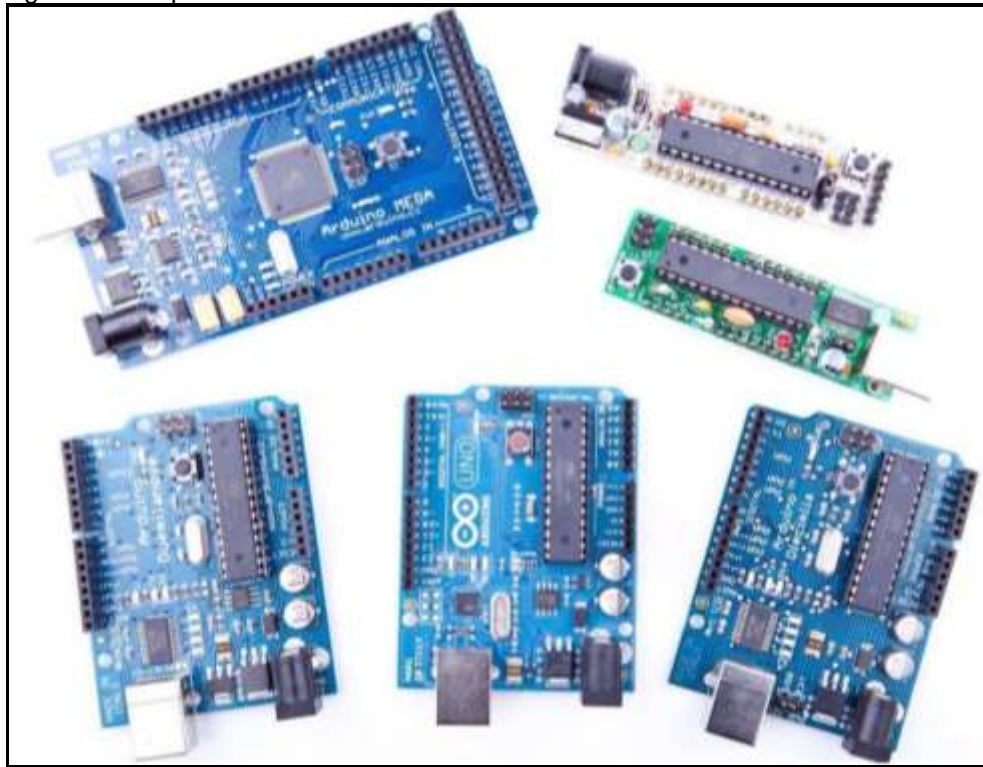
entrada e saída de dados. Pode ser usado para criação de sistemas embarcados simples, e até mesmo conexões interativas com o computador.

É um sistema diferente dos que existem no mercado por possuir um ambiente em multiplataforma, podendo ser executado no Windows, Macintosh e Linux. É fácil de realizar a conexão com o computador, já que usa uma porta USB e não uma porta serial, que é uma porta que não existe em muitos computadores modernos.

O Arduino é aberto em termos de hardware e software, pois caso seja de interesse de um desenvolvedor criar uma ideia e realiza-la com o Arduino, não é preciso pagar algo para os criadores do Arduino. O próprio hardware em si é barato, a placa inteira custa 35 dólares e caso algum componente queime, a troca é fácil e não custa muito mais do que 5 dólares. Ultimamente, muitas pessoas estão usando o Arduino para desenvolver suas ideias, e com isso, existe uma comunidade bastante grande e comunicativa que estão dispostas a ajudar umas às outras (BANZI, 2011, tradução nossa).

Na figura 5 tem-se uma variedade de modelos do Arduino, contendo o Arduino MEGA, Arduino UNO, entre outros. Os modelos se diferem principalmente pela quantidade de portas de entrada, processador e tamanho.

Figura 5 – Os possíveis modelos de diferentes Arduinos



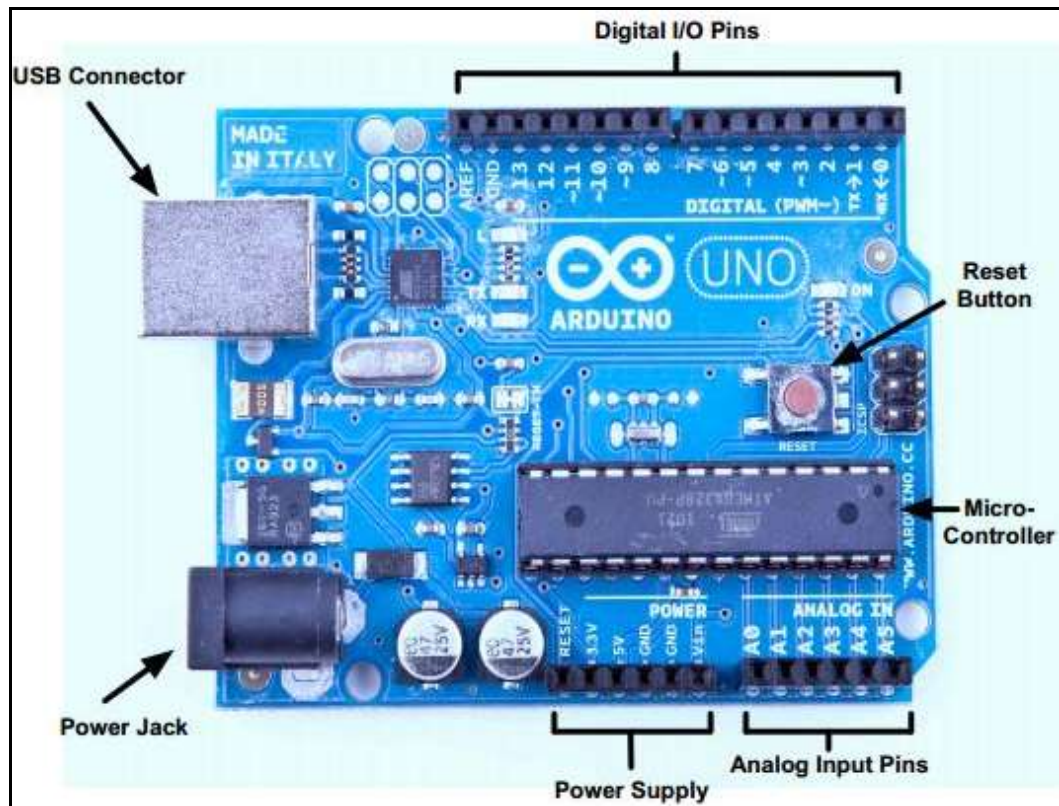
Fonte: Schmidt (2011).

Sabendo-se das vantagens que o Arduino oferece, segundo Schmidt (2011, tradução nossa), podemos encontrar incontáveis clones do Arduino pela internet. Qualquer pessoa é permitida a usar e modificar qualquer tipo de placa oficial, e muitas dessas pessoas que modificaram as placas oficiais, criaram suas próprias versões do Arduino que são compatíveis com o projeto Arduino. Juntamente com essas versões criadas, é possível encontrar o Freeduino, Seeduino, Boarduino, e até mesmo o Paperduino, que tem o modelo impresso numa folha de papel e suas partes são fixadas no papel que tem a impressão.

2.3.2 Conhecimentos Básicos

Para conhecer melhor a placa oficial, Schmidt (2011, tradução nossa) aponta uma exploração básica de uma placa exemplar do Arduino Uno.

Figura 6 – Um exemplar oficial do Arduino



Fonte: Schmidt (2011).

Schmidt (2011, tradução nossa), explica que o componente que é apontado pelo *USB Connector* na figura 6, é simplesmente uma porta para conexão de cabos USB, que podem ser usados para vários fins como realizar a gravação de um novo software para a placa em sua memória de programa. Pode ser usada para obter comunicação com a placa Arduino e o computador, e por fim, pode ser usada para alimentar o Arduino com energia.

É possível alimentar o Arduino com um computador por meio da conexão USB existente, mas não é uma condição apropriada em alguns casos, pois o Arduino ficará isolado de qualquer computador por falta da necessidade do mesmo, e por assim ser, seria necessário obter alimentação por outro meio. Se isso não é o bastante, ainda há casos que um projeto precisará mais do que 5 volts, que é a voltagem máxima que o computador pode fornecer pela conexão USB. Perante a uma dessas necessidades e situações, a melhor solução, normalmente, é usar um adaptador *Alternating Current (AC)* que fornecerá 9 volts (SCHMIDT, 2011, tradução nossa).

Figura 7 – Adaptador para fonte externa de energia



Fonte: Schmidt (2011).

Schmidt (2011, tradução nossa), explica que com o adaptador AC (figura 7), será necessário que a conexão seja do tipo ponta de barril com um centro positivo de 2,1mm. É possível então seguindo esses parâmetros, conectar aonde é apontado por *Power Jack*. Ao conectar o adaptador no local indicado, o Arduino ligará imediatamente, mesmo que o Arduino esteja ligado ao computador, ele usará a energia do adaptador se estiver disponível.

Na parte de baixo, na ponta direita da placa, é possível ver os pinos para entradas analógicas, que são nomeadas de A0-A5. É possível usa-las para conectar sensores analógicos ao Arduino, como sensor de temperatura. O trabalho consiste em captar o valor do sensor e transmiti-lo de forma numérica entre 0 e 1023 (SCHMIDT, 2011, tradução nossa).

No topo da placa, os pinos de entrada e saída para o Arduino são nomeados de D0-D13. Dependendo da aplicação é possível usá-los como entrada ou saída, e também é possível usa-los como saída analógica, convertendo valores de 0 à 255 em uma voltagem analógica (SCHMDIT, 2011, tradução nossa).

2.3.4 Programação

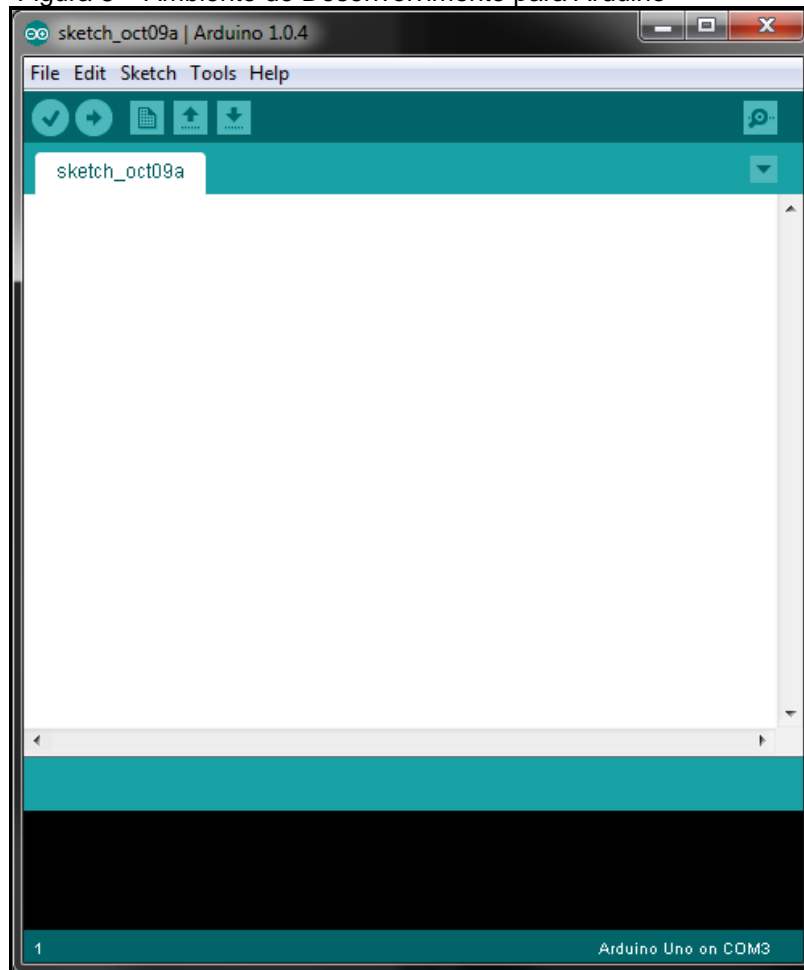
Programar e gravar os códigos para dentro do Arduino, Margolis (2011, tradução nossa) explica que os programas são chamados de *sketches*, esses são criados através de um ambiente de desenvolvimento próprio para o Arduino. A IDE

possibilita a escrita do programa e a edição dos mesmos e ainda converte este código em instruções para que o hardware do Arduino possa entender. A IDE, contudo, é a responsável por transferir essas instruções para o Arduino através da conexão com o computador pela USB.

2.3.4.1 Integrated Development Environment

Na figura 8, encontra-se a IDE usada para desenvolver *sketches* para o Arduino. No topo é possível visualizar botões com suas funcionalidades, que respectivamente representam ações para verificar, *upload*, novo, abrir e salvar. O botão de verificar realiza a função de transformar o código que está escrito dentro da IDE para um código transportável para o Arduino, contudo, qualquer tipo de evento imprevisto, como um código errado, será mostrado na parte preta colocando um aviso do que possivelmente está errado. O botão de upload é responsável por transferir um código compilado para dentro do Arduino, antes disso ele faz uma verificação e uma compilação para que só depois o programa pronto vá para a memória do Arduino. Os botões de novo, abrir e salvar são referentes ao *sketches* (SCHMIDT, 2011, tradução nossa).

Figura 8 – Ambiente de Desenvolvimento para Arduino



Fonte: Do autor (2013).

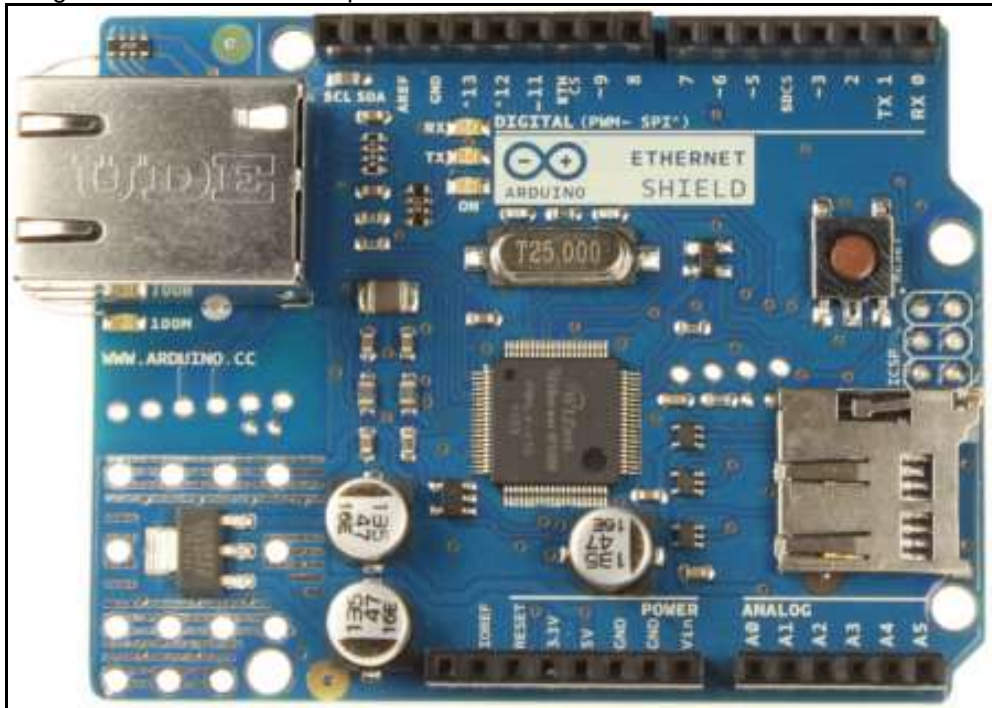
Para programar o Arduino a executar alguma função, é usada o software livre da figura 8, e para usá-lo, é necessário escrever códigos em uma linguagem que é baseada no C/C++ e ainda pode ser estendido por bibliotecas de C++. São esses códigos que são chamados de *sketches* (MCROBERTS, 2011, tradução nossa).

2.3.5 Shields

O Arduino pode ser estendido com outros sistemas embarcados, os *shields* (figura 9), que são placas que contém outras funções de dispositivos, tais como, receptor GPS, display em *Liquid Crystal Display* (LCD), módulos de *Ethernet*,

entre outros, que podem ser simplesmente encaixados na placa de um Arduino para conseguir sua função extra (MCROBERTS, 2011, tradução nossa).

Figura 9 – Shield Ethernet para Arduino



Fonte: Arduino (2013).

Os próprios *shields* estendem os pinos que ficam escondidos debaixo da placa *shield*, por esse motivo, é possível colocar *shields* em cima de outros *shields*, já que a conexão dos pinos do Arduino são estendidas por todos os *shields* (MCROBERTS, 2011, tradução nossa).

Por esse motivo, é comum projetos e protótipos usarem vários *shields*, já que cada um tem sua função específica, em um exemplo, é possível ter um *shield* para ethernet, e em cima deste ter um *shield* para comunicação WiFi e ao longo do último, um *shield* para visor de LCD, justamente o último para ter visibilidade do visor de LCD.

2.4 IDENTIFICAÇÃO POR RADIOFREQUÊNCIA

Processos de identificação automática tornaram-se popular em serviços industriais. O próprio sistema de código de barras trouxe uma revolução aos

sistemas de identificação há pouco tempo atrás. Porém, seu crescimento não pode ser maior devido seu pequeno poder de armazenamento e o fato de não poder ser reprogramável (FINKENZELLER, 2010, tradução nossa).

Assim como o código de barras, o RFID em forma de etiqueta, permite que objetos sejam identificáveis. Porém, de uma forma diferente que os códigos de barras (IGOE, 2012, tradução nossa).

Em geral, RFID é uma tecnologia de ótimo rendimento e pode ser usado como uma ótima ferramenta em vários casos fazendo papel para identificação e até mesmo substituindo a tecnologia de identificação que mais é usada nos dias de hoje, que neste caso, pode ser vista como o sistema de código de barras (SWEENEY II, 2005, tradução nossa).

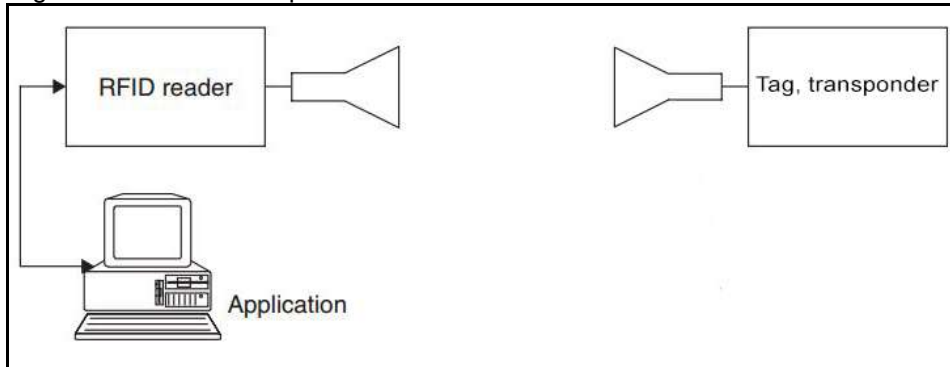
Wal-Mart tem gastado milhões de dólares desde o final da década de 90 pesquisando a eficácia que os sistemas RFID poderiam ter para substituir os códigos de barras. Então, em 1999, com a ajuda de alguns cientistas do *Massachusetts Institute of Technology*, foi formado o Auto-ID Center, que é um centro para pesquisas relacionadas a identificação por rádio frequência. Tiveram ideias inovadoras de como as organizações poderiam identificar e rastrear seus produtos. Hoje, a Auto-ID pode não só rastrear esses objetos, mas também o movimento de produtos, containers, veículo, entre outros que se movimentam em vastas áreas geográficas (SWEENEY II, 2005, tradução nossa).

2.4.1 Componentes

É possível classificar os sistemas RFID em dois tipos, os passivos e ativos. As *tags* de RFID passivos contêm um circuito integrado que possuem um sistema básico de rádio e um pequena quantia de memória não volátil. Tudo isso é alimentado pela energia corrente do sinal que o leitor induz nas antenas. A energia recebida é o bastante para alimentar a *tag* e fazê-la com que transmita suas informações por pelo menos uma vez, mesmo com um sinal relativamente fraco. Na maioria dos sistemas passivos, os leitores podem ler *tags* em uma distância curta de centímetros. Já em um sistema ativo de RFID, a *tag* tem sua própria fonte de eletricidade e transmite o sinal em resposta a mensagem recebida por um leitor.

Sistemas ativos podem transmitir em uma distância muito maior que os sistemas passivos, e são menos propensos a terem erros (IGOE, 2012, tradução nossa).

Figura 10 – Modelo simplificado do sistema RFID



Fonte: Do autor (2013)

Assim como na figura 10, os sistemas de RFID, possuem um leitor que se comunica com *tags* pelo ar com determinada frequência, como qualquer outra comunicação de rádio. O RFID é constituído basicamente por leitores, antenas, *tags*, e frequência. Os leitores funcionam de maneira parecida com os rádios de carros. O leitor produz eletricidade que corre por um fio até chegar na antena, a partir daí essa eletricidade radia o mesmo sinal digital pelo espaço com uma certa frequência e mesmo comprimento de onda. Não apenas gera sinal que vai da antena para o espaço, mas também fica a escuta de uma resposta que será produzida pela *tag*. Dessa forma, o leitor transmite e recebe ondas analógicas e as transforma em informação digital de zeros e uns (SWEENEY II, 2005, tradução nossa).

Na figura 11, é possível ver fisicamente exemplos de um leitor de dados que serão servidos a partir da antena de radiofrequência. Os exemplares de modelos de *tags* estão logo ao lado.

Figura 11 – Um leitor (preto), antena (branco) e 3 tipos de tags.



Fonte: Sweeney II (2005).

Enquanto o sistema leitor realiza transmissões de sinais pelo espaço, há de existir algo para transmitir informações de volta para o leitor realizar o seu trabalho, neste caso, existe a *tag*, que é responsável de mandar as informações de volta para o leitor (SWEENEY II, 2005, tradução nossa).

Sabendo-se da existência desses objetos que constituem o sistema, Finkenzeller (2010, tradução nossa), define os componentes fundamentais de um sistema RFID, estes que serão necessários para o bom funcionamento e integridade do sistema:

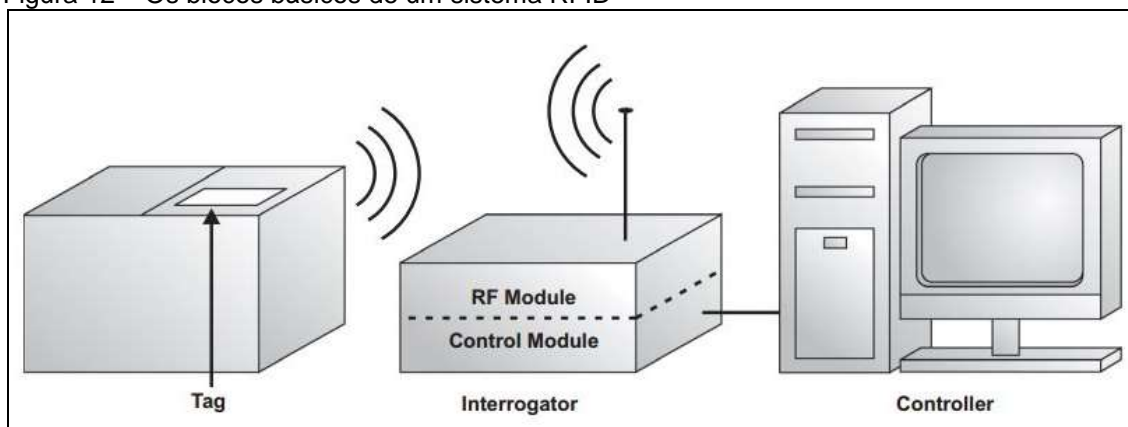
- a) o *transponder*, ou *tag*, que é localizado juntamente com o objeto a ser identificado;
- b) o interrogador, ou leitor, que, dependendo do design e da tecnologia usada, pode ser um dispositivo leitor, ou gravador e leitor.

Mas para Hunt (2007, tradução nossa), existem três componentes fundamentais para um sistema RFID (figura 12):

- a) a *tag* (que algumas vezes, diz ser chamada por *transponder*), que é composta por um chip semicondutor, uma antena e por algumas vezes uma bateria;

- b) um interrogador (que algumas vezes, diz ser chamado de dispositivo leitor ou leitor/gravador), que é composto por uma antena, um módulo eletrônico de rádio frequência, e um módulo controlador de eletrônica;
- c) um controlador (que algumas vezes, diz ser chamado de *host*), que na maioria das vezes é o computador, ou uma estação de trabalho que possui um banco de dados e um programa para controle.

Figura 12 – Os blocos básicos de um sistema RFID



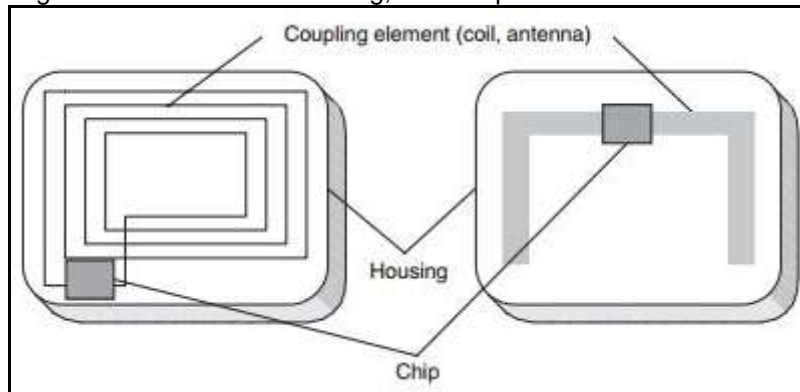
Fonte: Laran RFID (2012).

2.4.2 Tag

Uma *tag* de RFID é basicamente feito de duas partes: um chip de circuito integrado e a antena. O chip é como um pequeno computador que é responsável por guardar uma série de números que irão identificar unicamente a *tag*. É também responsável por ter a lógica implementada do que se fazer quando a *tag* encontrar um leitor por perto. A antena possibilita ativar o chip por meio da eletricidade recebida por ondas e realizar a comunicação, fazendo com que a *tag* realize trocas de informações com o leitor. Algumas *tags* são chamadas de *tags* ativas, que são *tags* que possuem baterias que fornecem energia para elas mesmas, mas na maioria, são mais produzidas as *tags* passivas, que são aquelas que dependem da eletricidade transmitida por ondas de rádio, isso significa que elas somente irão se comunicar quando realmente estiverem perto de um leitor. Estando perto de um leitor significa que estão dentro de um campo magnético, a *tag* alimenta-se da

energia desse campo para ter energia o suficiente para realizar a transmissão de seus dados (SWEENEY II, 2005, tradução nossa).

Figura 13 – Modelo de uma tag, ou transponder.

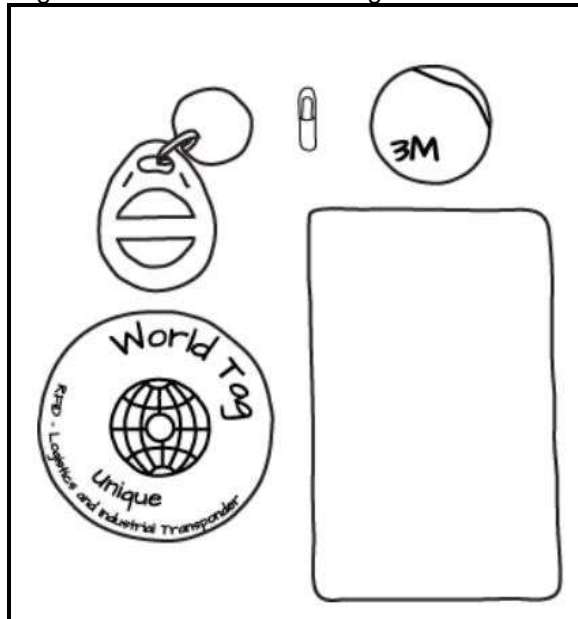


Fonte: Finkenzeller (2010).

Segundo Hunt (2007, tradução nossa), as *tags* (figura 13) de RFID podem ser feitas em diferentes formatos. Por causa do tamanho que o chip e a antena são feitas, que são bastante pequenos, é possível usa-los dentro de etiquetas adesivas ou pulseiras usadas por pessoas. Outros formatos podem ser:

- a) em alguns dos sistemas de RFID mais antigos, eles foram usados para a gestão de gados, e as *tags* eram plastificadas em formatos de “balas de revólver” fixadas nas orelhas do gado;
- b) as *tags* de RFID foram usadas em sistemas de cobrança automática de portagens. Não eram exatamente *tags*, porém, eram cartões de plásticos ou tipos de chaveiros;
- c) nas prisões, as *tags* foram utilizadas para serem incorporadas em pulseiras que eram usadas pelos presos e pelos guardas. De forma parecida com alguns motoristas do FedEx que carregavam em suas pulseiras uma chave de acesso para a van por meio de uma sistema sem chaves e de acesso ao sistema de ignição.

Figura 14 – Variedades de Tags



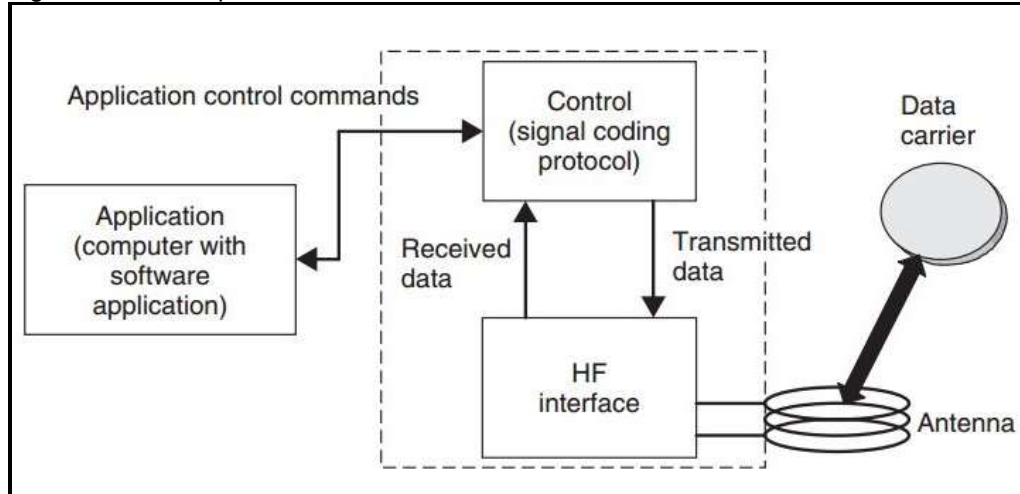
Fonte: Igoe (2011).

Assim como Hunt (2010, tradução nossa), Igoe (2012, tradução nossa) também mostrou os diferentes formatos que as *tags* podem ser feitas como são mostradas na figura 14: *tags* para colar, em moedas, chaveiros, cartões de créditos, e até as de formato para injetar debaixo da pele que são usadas em animais para seu rastreamento, estes não são desenhadas para humanos.

2.4.3 Leitores

Os leitores, basicamente, possuem várias características que os identificam no mundo, que irão fazê-lo bastante único pelas inúmeras combinações dessas características, podendo ser indutivo, ou usar eletromagnetismo, ou suas frequências e distâncias, porém, em seu princípio de operação, irão ser todos similares. Os leitores em todos os sistemas podem ser separados em dois blocos fundamentais de operações (figura 15): o sistema de controle e a interface de rádio frequência, que irá consistir de um transmissor e um receptor (FINKENZELLER, 2010, tradução nossa).

Figura 15 – Exemplo de funcionamento dos leitores



Fonte: Finkenzeller (2010).

Segundo Finkenzeller (2010, tradução nossa), a interface de rádio de frequência irá exercer as seguintes funções:

- a) gerar transmissão de energia com alta frequência para ativar o *transponder* e suprir ele com energia;
- b) modulação do sinal de transmissão para enviar dados ao *transponder*;
- c) recepção e demodulação dos sinais de rádio frequência transmitidos pelo *transponder*.

2.4.4 Caracterização

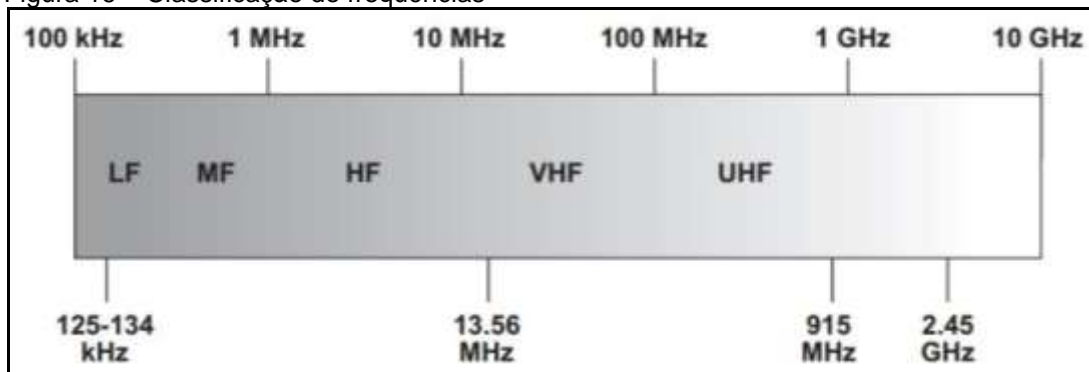
As partes mais importantes que caracterizam, definem e diferenciam um sistema de RFID são sua frequência de operação do leitor, o método usado para desenvolvê-lo e o alcance desse sistema. Existem variadas frequências utilizadas, podendo ser simples ondas de 135 kHz até mesmo micro-ondas de 5.8 GHz. Eletricidade, magnetismo e campos eletromagnéticos são métodos utilizados para a construção de um sistema RFID. E por fim, seu alcance pode variar de poucos milímetros até mesmo 15 metros (FINKENZELLER, 2010, tradução nossa).

2.4.4.1 Frequência, distância e acoplamento

Uma chave importante para o RFID é a frequência em que operam. Como uma televisão pode ser emitida em VHF ou em uma banda UHF, assim, sistemas RFID também podem usar diferentes bandas para se comunicar (HUNT, 2007, tradução nossa).

Na figura 16, é possível verificar a classificação de várias bandas de frequência, as apontadas na parte de baixo, são as frequências mais usadas por sistemas de RFID.

Figura 16 – Classificação de frequências



Fonte: Hunt (2007).

Sobre o tipo de acoplamento entre a tag e o leitor, Finkenzeller (2010, tradução nossa) afirma que sistemas de RFID que usam frequências aproximadamente entre 100 kHz e 30 MHz operam usando acoplamento indutivo. Porém, os que usam acoplamento usando campos eletromagnéticos são sistemas que usam frequências entre 2.45-5.8 GHz.

Sistemas com uma mínima distância, tipicamente aqueles de até 1 cm, são conhecidos como sistemas de acoplamento curto. Para a operação de comunicação ser feita, o *transponder* precisa ser inserido dentro do leitor ou posicionado em uma superfície própria para a distância. Os sistemas de *close-coupling* são unidos usando eletricidade e campos magnéticos e teoricamente podem ser operados em qualquer frequência dada entre DC e 30 MHz porque a operação do *transponder* não depende da radiação dos campos. Esse tipo de sistema é usado primordialmente em aplicações que estão sujeitas a ter restrições

de segurança, mas que não requerem distâncias grandes. São exemplos, sistemas de trancas de portas ou sistemas de pagamentos com cartões inteligentes (FINKENZELLER, 2010, tradução nossa).

Sistemas com gravações e leituras com distâncias de até 1 metro são conhecidas pelo termo coletivo de sistemas de acoplamento remotos. Quase todos desse tipo de sistema são de acoplamento indutivo (magnético) entre o leitor e o *transponder*. Esses sistemas são também conhecidos como sistemas de rádio indutivos. Alguns poucos são de acoplamento capacitivo (elétricos). No mínimo, 90% de todos os sistemas de RFID vendidos são sistemas de acoplamento indutivo. Frequências abaixo de 135 kHz ou 13.56 MHz são usadas como frequências de transmissão (FINKENZELLER, 2010, tradução nossa).

Os sistemas com distâncias de mais de 1 metro são conhecidos como sistemas de longe alcance. Todos os sistemas desse tipo operam com ondas eletromagnéticas na escala de micro-ondas e UHF (FINKENZELLER, 2010, tradução nossa).

A Adafruit, uma empresa norte americana, produz alguns produtos de rádio frequência. Para seus consumidores saberem quais tipos de produtos comprarem, a organização disponibiliza um guia. Na figura 17, é mostrado algumas frequências e suas características gerais que impactam em um sistema RFID.

Figura 17 – Relação de frequências

Frequency	LF 120 ~ 134 kHz	HF 13.56 MHz	UHF 850 ~ 960 MHz
Read Range	0.5 ~ 1 m	< 1 m	> 3 m
Cost	Relatively expensive	Less expensive	Least expensive
Penetration of Materials	Excellent ←————→ Poor		
Affected by Water?	No	To some extent	Yes
Antenna	Coil	Coil	Dipole, Slot
Data Rate	Slower ←————→ Faster		
Reading Multiple Tags	Poor	Good	Very Good
Applications	Immobilisers, industrial-identification	"Pharma", libraries brand protection, tickets, payments, passports	Pallet/case tracking, tolls baggage tracking, PCB tracking

Fonte: Adafruit (2010).

2.4.5 Regulamentação

Justamente por causa dos sistemas gerarem ondas eletromagnéticas, eles são legalmente classificados como sistemas de rádio. O funcionamento de outro serviço de rádio não deve em nenhuma circunstância ser disputado pela operação dos sistemas de RFID. É de importância certificar que os sistemas em RFID não interfiram com rádios e televisões, serviços de sistemas móveis (polícia, serviço de segurança, indústrias), marinha ou serviços de rádio aeronáuticos e serviços de celulares. A existência de frequências ISM é reservada internacionalmente para aplicações em dispositivos que usam radiofrequências, como aplicações industriais, científicas ou medicinais. Exemplos disto são micro-ondas, EDM, ou radioterapia. A ideia original da ISM é que qualquer um poderia usar dispositivos de rádio dentro das bandas de ISM (FINKENZELLER, 2010, tradução nossa).

No Brasil, o órgão responsável por regulamentar as frequências usadas no Brasil, a ANATEL, disponibiliza um documento que regula as frequências ISM, e segundo a ANATEL (2013), as seguintes frequências pertencem à banda ISM:

- a) 13553-13567 kHz (frequência central 13560 kHz);
- b) 26957-27283 kHz (frequência central 27120 kHz);
- c) 40,66-40,70 MHz (frequência central 40,68 MHz);
- d) 902-928 MHz (frequência central 915 MHz);
- e) 2400-2500 MHz (frequência central 2450 MHz);
- f) 5725-5875 MHz (frequência central 5800 MHz);
- g) 24-24,25 GHz (frequência central 24,125 GHz).

2.5 SOFTWARES DE APOIO PARA DESENVOLVIMENTO

Alguns softwares dão apoio ao desenvolvimento de projetos como este, neste caso, Visual Studio que é construído e distribuído pela Microsoft tem a habilidade de construir softwares para computadores. (TROELSEN, 2012, tradução nossa).

Outros softwares, como o Wireshark, tem o poder de dar auxílio em questões específicas, por exemplo, dar informações que sirvam de apoio para resolver problemas de comunicações por ethernet.

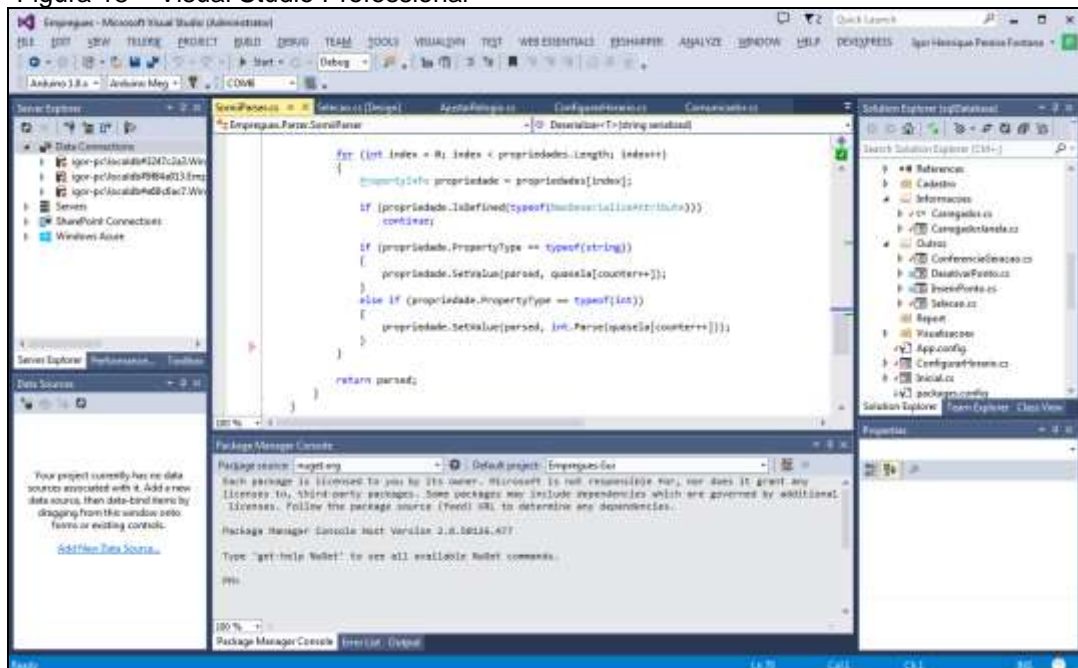
2.5.1 Visual Studio e o .NET Framework

Quando a Microsoft lançou o .NET Framework, 10 anos atrás, ocorreu muita desordem dentro da comunidade de desenvolvedores. Esse framework modernizou ideias de como programar, usando técnicas mais sofisticadas. Uma das linguagens que foram criadas para utilizar esse framework foi o C#. (SUR, 2013, tradução nossa).

Para que o framework da Microsoft, juntamente com a linguagem, tivessem seu uso apropriado, a empresa precisava lançar uma IDE para suportar a

programação utilizando tais recursos. A resposta para tal necessidade foi o Visual Studio (figura 18).

Figura 18 – Visual Studio Professional



Fonte: Do autor

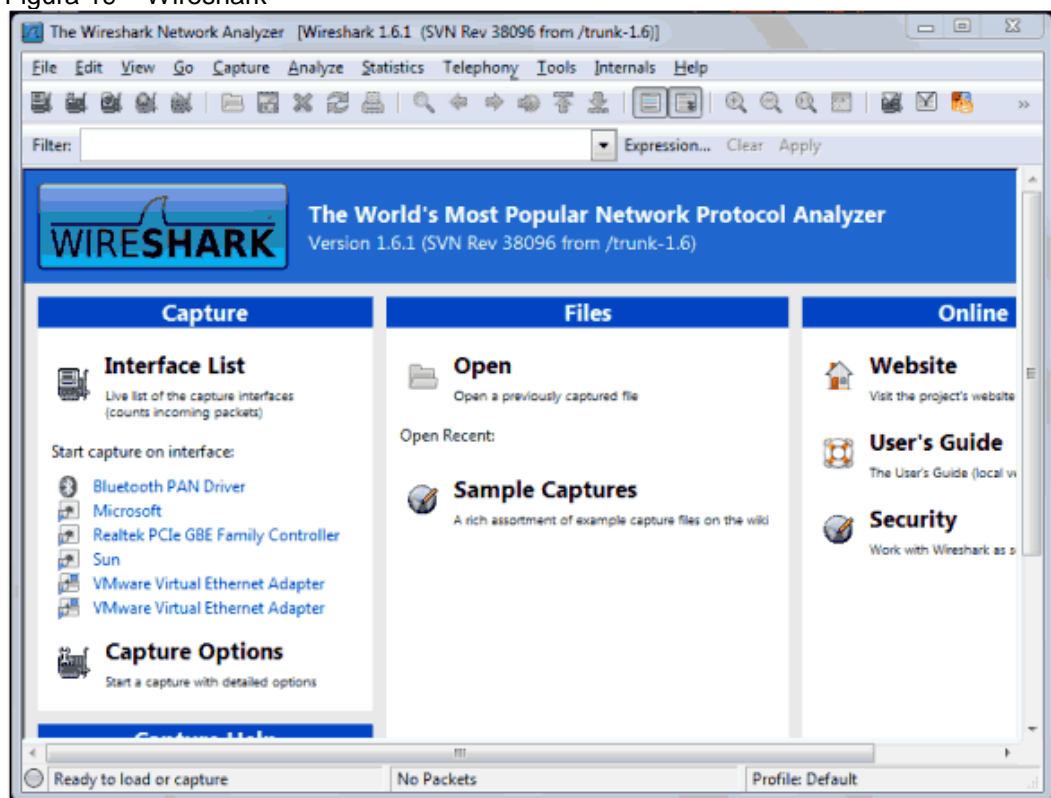
SUR (2013, tradução nossa) explica que existem quatro versões do Visual Studio para que o desenvolvedor escolha como sua IDE, são elas:

- na versão Express o desenvolvedor conta com uma IDE simples e que deseja não gastar capital para construir aplicações simples. A versão é distribuída gratuitamente para qualquer pessoa;
- na versão Professional o desenvolvedor conta com uma IDE mais sofisticada, com ferramentas melhores de *debug* e outros confortos. O preço da ferramenta é razoável;
- a versão Premium é destinada a construção de aplicações de alta qualidade que usam exclusivamente a IDE;
- para o uso da versão Ultimate, é recomendado que seja usado em uma grande equipe. É a versão com todos os recursos possíveis da IDE, incluindo arquitetura de software e modelagem.

2.5.2 Wireshark

SINGH (2013, tradução nossa) afirma que o Wireshark (figura 19) é uma ferramenta de código aberto que oferece análise dos pacotes de uma rede, capturando qualquer tráfego de pacotes de dados pela rede e apresentando-os em uma forma que se possa entender os dados e os analisar. Este software atua como um canivete suíço, pode ser usado de várias maneiras para vários problemas diferentes, tais como auxiliar no conserto de problemas de rede, operações de seguranças e aprendizado de protocolos internos.

Figura 19 – Wireshark



Fonte: Singh (2013)

SINGH (2013, tradução nossa) também aponta alguns benefícios notáveis do uso do Wireshark, são eles:

- a) Wireshark suporta uma grande variedade de protocolos, sendo TCP, UDP, HTTP, entre outros.

- b) Conta com uma interface que ajuda com todas as análises. Oferece opções avançadas, tais como filtro de pacotes e exportação de pacotes.
- c) Pode capturar o tráfico de dados ao vivo e rapidamente gerar informação sobre os protocolos, canais de comunicações, entre outros.
- d) Como o projeto é de código aberto, ele já foi desenvolvido por mais de 500 desenvolvedores ao redor do mundo. Qualquer um pode escrever seu próprio código e submetê-lo para que seja avaliado e aceite para o programa final.

3 TRABALHOS CORRELATOS

A realização do trabalho necessita de uma pesquisa de trabalhos semelhantes, que abordam pelo menos uma característica dos assuntos aqui pesquisados, isso é realizado devido a necessidade de um maior apoio e base para a construção do projeto.

3.1 RFID: ANÁLISE DA VIABILIDADE, VANTAGENS E DESVANTAGENS DA TECNOLOGIA E DESENVOLVIMENTO DE UM SISTEMA PARA DEMONSTRAÇÃO E TESTES

O tem objetivo elaborar um protótipo composto por software e hardware para demonstrar o funcionamento, vantagem, praticidade e viabilidade da utilização da tecnologia RFID em um ponto de venda. Além de, estudar o conceito de comunicação sem fio, o estudo do sistema RFID apresentando as suas características.

3.2 CONTROLE DE PRESENÇA UTILIZANDO RFID: UM ESTUDO DE CASO UTILIZANDO A LINGUAGEM RUBY

Como destaque, é ressaltado que mesmo sendo simples, ainda temos dificuldade para controlar a presença de pessoas em determinados eventos, como chamadas de alunos. Os procedimentos, muitas vezes, provocam atrasos e perdas de tempo. O projeto tem finalidade de simplificar o processo, tornando-o automatizado com o uso de um sistema RFID. Os participantes usariam *tags* para registrar sua presença no sistema na hora da entrada, assim a própria entrada do participante seria sua chamada automatizada. Os resultados indicaram que o processo poderá ser utilizado para agilizar o acesso e o controle de presença de participantes em eventos.

3.3 CONTROLE DO FLUXO DE PESSOAS USANDO RFID

No trabalho é apresentado o desenvolvimento de um aplicativo para o controle de fluxo que utiliza a tecnologia de RFID, ressaltando que a tecnologia pode ser muito útil em empresas, aonde pode ser possível controlar o acesso às áreas restritas da organização, possivelmente também registrando os horários de entradas e saída de cada colaborador. O trabalho apresenta RFID, de forma que possa ser inserida de forma simples em instituições, para comprovar e exemplificar, foi realizado um experimento em ambiente escolar.

4 DESENVOLVIMENTO DO SISTEMA

Este trabalho tem o objetivo de construir um protótipo para ter o funcionamento de um sistema para registrar os pontos de forma eletrônica e que seja possível identificar os colaboradores com um leitor de etiquetas RFID. Há também a criação de um software para gestão do sistema eletrônico e que tenha a habilidade de gerar um relatório dos pontos.

4.1 METODOLOGIA

4.1.1 Aprendizado e compreensão do Arduino

O processo de compreensão sobre desenvolver protótipos com o Arduino requereu alguns componentes. Para o início, foi necessário um kit de desenvolvimento próprio para iniciante (figura 20), esse que ajudaria nas questões de aprendizado.

Figura 20 – Kit de Arduino para iniciantes



Fonte: Arduino.com

Acompanha-se no kit o Arduino UNO, que consiste num modelo simples, próprio para iniciantes. As características desse modelo são básicas, é constituído por 14 pinos de entrada e saída digitais, 6 pinos analógicos, 2 KB de SRAM e 32 KB de memória flash. Seus pinos de entrada e saída são utilizados para a troca de

informação com outros componentes ou até mesmo outro módulos, destes que podem ser módulos destinados a controle de tempo, temperatura ou até mesmo um outro Arduino.

Com base no site oficial do Arduino, onde há vários guias para os iniciantes do uso da plataforma, foi dado os primeiros passos de estudo da plataforma. Um dos exemplos mais básico, é um guia onde aprende-se como utilizar o microcontrolador para controlar um LED. Posteriormente, os guias levam qualquer um a ter conhecimento de como controlar componentes mais complexos, tais como telas de LCD.

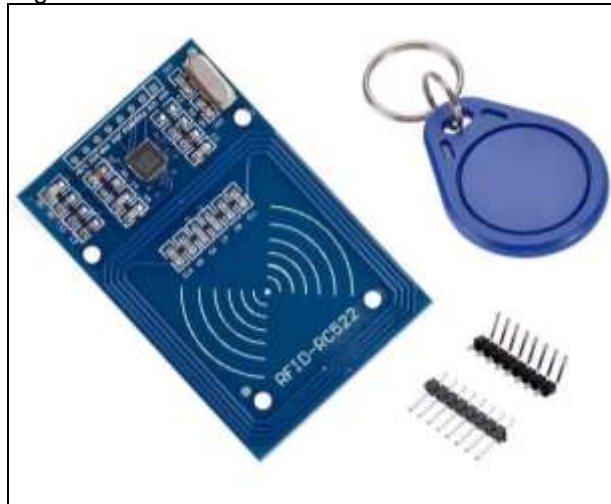
O LCD estudado ocupa muitos pinos do microcontrolador, desta forma conduzindo a uma escassez de pinos disponíveis, isso gera uma dificuldade em conectar outros componentes para a comunicação e controle junto ao Arduino. Para tornar o problema menor, era possível utilizar chips chamados *shift register*, diminuiria a quantidade de pinos necessários para o funcionamento do componente, deixando assim outros pinos livres para os outros, mas mesmo com esse método, não seria possível usar a quantidade desejada de componentes.

Após os estudos sobre os componentes que acompanhavam o kit de iniciante, foi iniciado a seleção e aquisição dos que ainda faltavam para o desenvolvimento do projeto.

4.1.2 Seleção do Hardware necessário

O primeiro item a ser selecionado foi o leitor das etiquetas de RFID. O leitor não precisava ser potente, nem que fosse habilidoso em realizar leituras de longas distâncias, essas características contribuíram para que o leitor escolhido tivesse um valor com baixo custo. RFID-RC522 (figura 21) foi o leitor escolhido, o vendedor forneceu uma etiqueta no formato de chaveiro e uma outra com o formato de cartão de crédito.

Figura 21 – Módulo RFID-RC522



Fonte: image.dhgate.com

Outros dois componentes foram necessários para se comunicar pela rede, dessa forma realizando a troca de dados entre o Arduino e um computador que seria responsável por ter o software gestor e realizar sua gestão, o outro componente deveria ser capaz de persistir dados dos colaboradores, pontos, entre outros.

Com base na requisição, a escolha não poderia ser mais óbvia do que o *Ethernet Shield*, sabendo-se que o Arduino suporta componentes que encaixam-se perfeitamente em sua superfície, neste caso, os shields. Foi possível adquirir esse shield que tinha a capacidade de atender as duas requisições ao mesmo tempo. Para a persistência dos dados, o shield conta com um espaço para usar cartões miniSD.

Devido à complexidade do módulo RFID, o vendedor fornece uma biblioteca para facilitar sua comunicação com o Arduino, basta carregar do site para o computador e utiliza-lo juntamente a programação do sistema. Para utilizar o módulo leitor, foi necessário pesquisar alguns guias de como conectar os pinos do módulo ao Arduino e realizar as primeiras leituras.

O *shield* de Ethernet também possui uma complexidade parecida com a do módulo RFID, mas por este *shield* se tratar de um modelo oficial do Arduino, no site oficial, há uma biblioteca para seu uso com o microcontrolador. Com essa biblioteca, é possível realizar operações de forma descomplicada.

Figura 22 – Ethernet Shield acoplado em um Arduino UNO



Fonte: zugiduino.wordpress.com

Na figura 22, há um Ethernet Shield no topo de um Arduino UNO. É possível ver que um shield para a plataforma estende os pinos originais e não dá indícios de quais pinos são usados pelo *shield*. Diante esses fatos, é necessário averiguar as informações contidas nos guias que há no site oficial do Arduino. O shield usa a comunicação SPI para a comunicação da rede e também para o controle do cartão que for inserido no espaço definido.

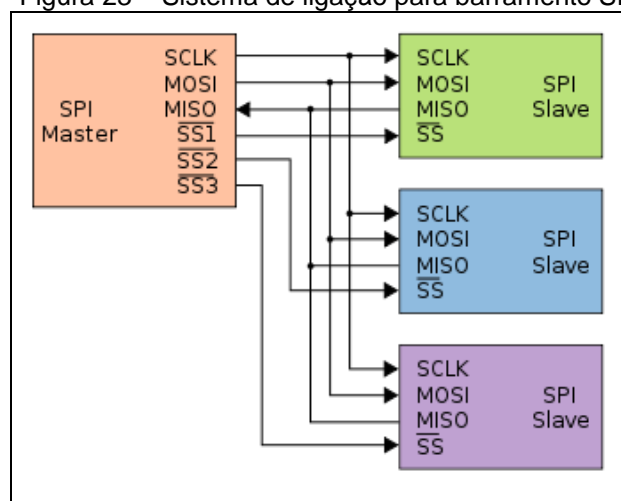
Obteve-se então, após os conhecimentos iniciais, resultados positivos diante os testes de transferência de dados pela rede. A comunicação do cartão miniSD também teve sucesso, sendo possível guardar informações em arquivos dentro do mesmo. As bibliotecas fornecidas para o *shield* oferecem um manuseio facilitado e produtividade.

Houve o aparecimento de um problema no decorrer do desenvolvimento. A tentativa de fazer com que o RFID e o Ethernet Shield trabalhassem conectados ao mesmo tempo. Sem sucesso na tentativa, foi feita uma pesquisa sobre uma possível incompatibilidade dos Shields terem na execução simultânea. A busca indicou que eles não tinham problemas de incompatibilidade, o próximo passo foi pesquisar informações sobre as portas 13, 12 e 11 que o RFID usava, o resultado da pesquisa foi que tais portas eram usados pelo Arduino para o barramento SPI, que tem significado de Serial Peripheral Interface. Justamente por essa ocasião, seria necessário entender o SPI para que os dois trabalhassem simultaneamente, pois os mesmos não continham incompatibilidade como o pensando anteriormente.

4.1.3 Barramentos de comunicações utilizadas pelo Arduino

Podendo ser chamado simplesmente de SPI ou barramento SPI. Com uma explicação simples, é uma interface que permite a comunicação de um microcontrolador com diversos outros componentes, formando uma rede de periféricos.

Figura 23 – Sistema de ligação para barramento SPI



Fonte: coocox.org

Na figura 23, é possível identificar um componente *Master*, que no caso do projeto é o Arduino MEGA, e os seus *Slaves*, ou seja, os componentes extras que realizam funções extras.

O pino SCLK (Serial Clock) é responsável por compartilhar o Clock do mestre e se encontra no pino 13 do Arduino, MOSI (Master out Slave in) é responsável por transferir dados do *Master* para o *Slave*, que é o pino 11, o MISO (Master in Slave out) é responsável pelas transferências dos dados do *Slave* para o *Master*, que é o pino 12, e o SS (*Slave Select*) é responsável por selecionar o *Slave* que trocará informações com o Microcontrolador quando pedido.

Uma outra interface também é usada no Arduino e esta é destinada a uma maior facilidade em controlar os componentes extras, ela é chamada de TWI mas o seu funcionamento é o mesmo do I2C.

O barramento IIC (Inter-Integrated Circuit) ou I2C, tem a mesma finalidade do SPI que é integrar vários componentes extras a um mesmo microcontrolador. No

I2C ao invés de 3 fios e mais um extra para cada componente, ele usa apenas 2 fios, independentemente da quantidade de componentes extras. O I2C é chamado de TWI (Two Wired Interface) no Arduino, mas tem a mesma funcionalidade, apenas com um nome diferente. É usado os pinos A4 e A5 no Arduino UNO para este barramento.

Os estudos sobre os barramentos de comunicações usados pelo Arduino é muito importante, isso habilita o uso de vários componentes usando pouquíssimos pinos para o funcionamento dos mesmos. Com a ajuda dos barramentos, foi possível conectar o Ethernet Shield juntamente com o leitor do RFID na comunicação via SPI.

Mesmo com o sucesso de conectar os dois componentes ao Arduino e realizar operações básicas utilizando esses barramentos, o módulo de RFID apresentou problemas durante o funcionamento do mesmo, como dificuldade de leitura de algumas etiquetas ou até mesmo não realizar leitura alguma.

4.1.4 Seleção dos componentes para uso do barramento I2C

Diante do problema do módulo atual do RFID, foi necessário buscar outro módulo RFID, com mais qualidade mas não necessariamente com mais funções. Com a requisição, foi decidido que o módulo de RFID (figura 24) construído pela ElecFreaks seria escolhido. Este novo módulo apresenta melhor qualidade de produção e também há a possibilidade de definir o tipo de comunicação do módulo que pode ser SPI, I2C ou UART.

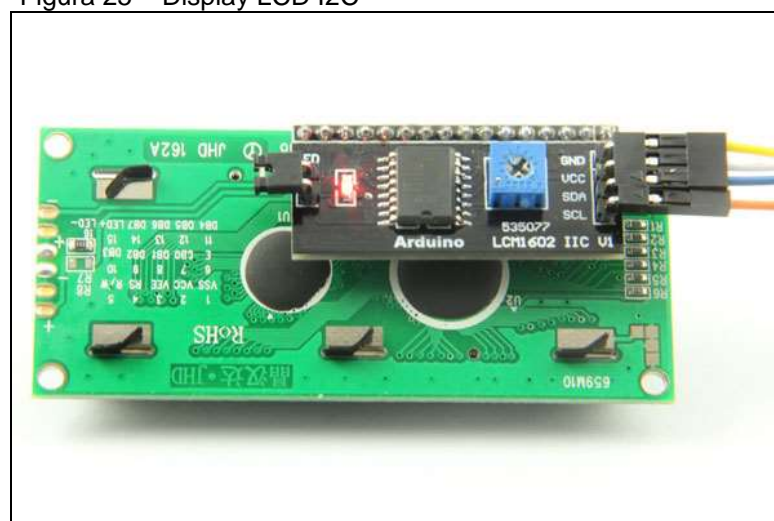
Figura 24 – Módulo RFID/NFC ElecFreaks



Fonte: elecfreaks.com

Para o melhor aproveitamento das comunicações em barramento até então citadas, o ideal é que a maioria dos componentes utilizados no projeto pudessem realizar suas respectivas comunicações usando esses barramentos. Foi selecionado um display LCD que é compatível com I2C (figura 25) e também um Real Time Clock (RTC) com a mesma compatibilidade (figura 26), dessa forma o projeto ganha espaço para utilizar componentes que usem o restante de pinos do Arduino.

Figura 25 – Display LCD I2C



Fonte: geetech.com

Figura 26 – Módulo RTC I2C



Fonte: tinyosshop.com

Com o uso de todos os componentes utilizando os barramentos de comunicações do Arduino, é possível utilizar o módulo de RFID, LCD e RTC juntamente com apenas dois pinos e o Ethernet Shield e miniSD com mais quatro pinos.

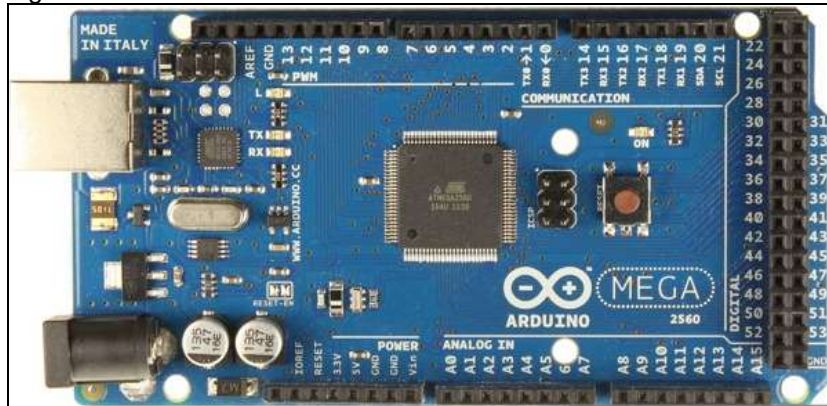
Para que todas as peças escolhidas fossem validadas para o projeto, foi necessário realizar um teste onde as mesmas seriam conectadas ao Arduino, importar todas as bibliotecas necessárias em um projeto de Arduino, programar alguma função básica para cada componente.

Apesar do UNO ser um modelo para suportar vários projetos, suas capacidades não foram suficientes para o projeto em questão e teve que ser substituído pelo seu irmão maior, o Arduino MEGA 2560.

4.1.5 Mudança do Arduino para versão MEGA e testes necessários

Ao início do teste, foi constatado que o modelo UNO do Arduino não é compatível com o tamanho do projeto uma vez que apenas as bibliotecas ocupam toda sua memória de programa. Foi necessário realizar a aquisição de um modelo melhor do Arduino, sua versão MEGA (figura 27) fornece oito vezes mais memória de programa, desta forma, habilitando mais código escrito.

Figura 27 – Arduino MEGA



Fonte: adafruit.com

Algumas diferenças são notáveis quando é usado o Arduino MEGA para realizar algum trabalho com peças externas, a maneira que é conectado os *shields* são diferentes, alguns módulos devem ser conectados em pinos com números diferentes dos que são usados no UNO.

No início do novo teste que até então não foi possível realizar, houve o aparecimento de um problema que foi necessário muito tempo para solucioná-lo. Quando todos componentes foram ligados e após a programação dos mesmos, o módulo que controla o tempo real fornecera horários estranhos, que eram resultados de um mal funcionamento interno.

A solução do problema não foi encontrada em nenhuma fonte externa, contudo, quando o módulo era tocado por mãos humanas em determinados locais, o módulo voltava a fornecer os números corretos de data e horário. Foi constatado que a linha de comunicação de dados do módulo estava com tensões altas, e para diminuir estas tensões foi necessário colocar à linha um resistor pequeno para que a tensão caísse, e apenas desta maneira, o módulo forneceu a tensão correta para a linha de comunicação.

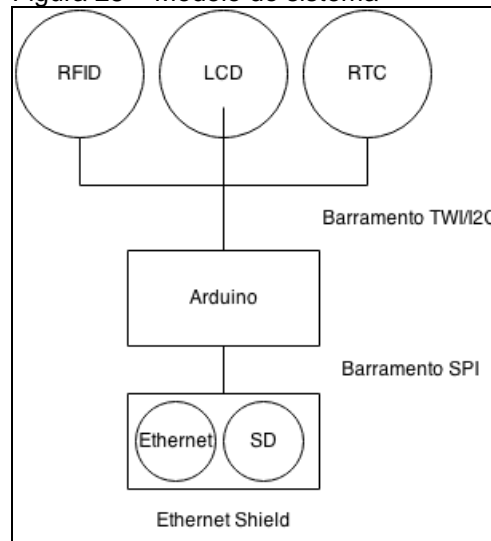
Para que a conclusão de testes tivesse êxito, foi necessário fazer com que cada componente realizasse uma determinada função, sendo possível avaliar o funcionamento do hardware e também se as escolhas das peças eram de fato definitivas.

Um pequeno sistema de demonstração foi construído, onde ao passar uma etiqueta pelo leitor, o sistema realizava um envio da informação para o

computador via rede e ao mesmo tempo que esse envio era efetuado, foi mostrado na tela LCD e gravado no cartão miniSD. O teste foi concluído e o sistema funcionara com sucesso.

O modelo do sistema finalmente foi definido como na figura 28, os dois barramentos seriam usados, SPI para construir a interface entre o Arduino e o Ethernet Shield que contém em seu interior a parte de Ethernet e também do controle do microSD que será utilizado para armazenar todos os dados do sistema.

Figura 28 – Modelo do sistema



Fonte: Do autor

Com o hardware definido e montado, o desenvolvimento lógico foi iniciado, este que consiste em controlar o Arduino e seus módulos e também um gestor para o Windows que tenha finalidade de realizar o controle do sistema.

4.1.6 Desenvolvimento do código do Arduino e seu gestor para Windows

4.1.6.1 Organização de código

Para a simplificação do desenvolvimento da parte lógica do sistema, foi realizado separações na construção dos códigos.

4.1.6.1.1 Estrutura para o código do Arduino

Na construção do código para o Arduino, a primeira parte construída foi um simples código destinado a controlar alguns arquivos que são salvos no microSD.

A segunda parte construída foi destinada a troca de informações pela ethernet, uma classe que tem o poder de trocar as informações de uma maneira descomplicada e que garante os dados em um formato correto de leitura, dessa forma, ao invés de ler todos os caracteres de forma descontrolada quando vinda

pela rede, é definido um ponto máximo de leitura, onde é empacotado uma quantidade de informações e feito sua interpretação, logo após, é continuado a leitura dos próximos dados e o ciclo recomeça.

Na terceira e última parte da construção do programa, é a parte principal, onde é chamado todas as outras partes e é realizado o uso dos códigos que foram escritos. É definido quando e onde serão feitas as leituras e respostas da rede local e também quando usados os métodos de gravação e leitura dos dados do microSD.

4.1.6.1.2 Estrutura para o código do gestor

Para o desenvolvimento do programa de computador de gerência dos dados, tais como cadastrar um novo colaborador ou empregador, incluir pontos esquecidos ou excluir pontos que foram marcados errados e gerar seus relatórios, foi usado a IDE disponibilizada pela Microsoft, o Visual Studio 2013.

A construção do programa do Windows também foi realizada em partes, este se encontra um pouco mais complexo por tratar de vários mecanismos diferentes, tais como funções para preparar pacotes, comunicar-se através da ethernet, serializar e desserializar um objeto, controle de banco de dados, geração de relatórios e controle de dados de execução.

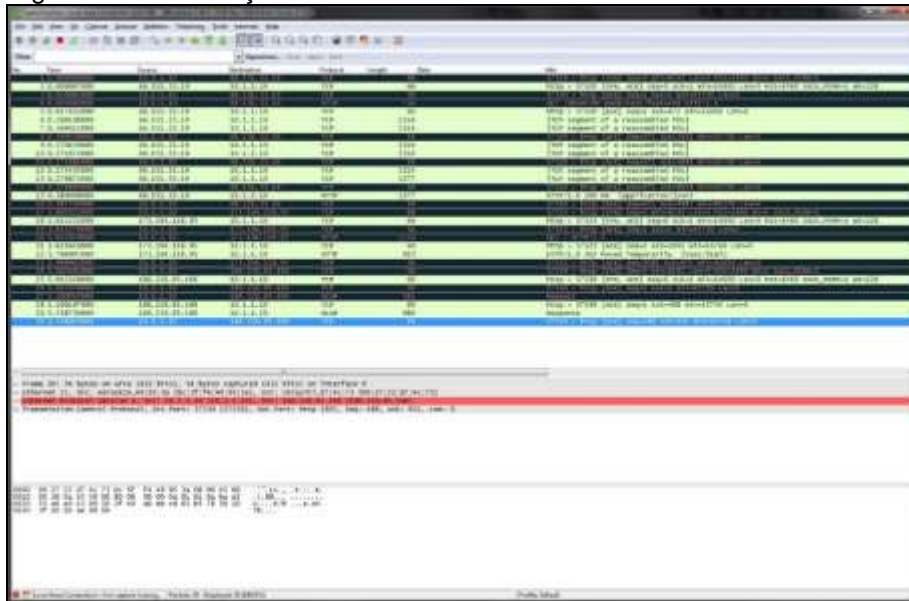
4.1.6.2 Definição da metodologia de desenvolvimento de código

Para que o desenvolvimento pudesse ser testado ao longo do mesmo, foi necessário que ambas partes, tanto quanto o software para Windows e o sistema com o Arduino, fossem construídas juntas, caso contrário, não seria possível saber se a comunicação pela rede funcionaria.

A construção do projeto teve início pela parte do desenvolvimento onde é realizada a comunicação da rede, utilizou-se de uma técnica para separar os dados necessários em pedaços que são enviados como pacotes, obteve-se vários problemas de comunicação e controle dos dados que eram enviados. Devido a esse problema, foi necessário o uso de um software chamado Wireshark.

O Wireshark (figura 29) é um programa que facilita o controle dos dados que são recebidos e enviados pelo computador que o possui. É possível ver e analisar todos os dados que estão sendo trocados com qualquer outro dispositivo que comunica-se através dos protocolos suportados.

Figura 29 – Detecção da rede utilizando o Wireshark



Fonte: Do autor

A comunicação com o Arduino é feita através do protocolo TCP/IP, como o Wireshark oferece suporte para este protocolo, foi possível realizar as análises dos pacotes trocados pela rede, e consequentemente, reestruturar o código necessário para a comunicação de rede, desta forma, fazendo com que os pacotes fossem escritos de maneira eficiente e correta, o mesmo para a classe que realiza a comunicação que utiliza esses pacotes.

4.1.6.3 Transmissão dos dados entre o gestor e o Arduino

Com o objetivo de otimizar e simplificar a transmissão de dados pela rede, foi feita a tentativa de utilizar uma serialização de dados que é usada em vários softwares. JavaScript Object Notation (JSON), é um método para transformar objetos dentro da programação em um pequeno texto em que é possível um humano ler de forma compreensível e ao mesmo tempo ter a habilidade de voltar ao

seu estado original ou transformar-se em um objeto de qualquer outra linguagem, desde que esta tenha a capacidade de realizar tal processo.

Com a existência de um pequeno framework de JSON para .NET construído por James Newton-King, é possível realizar transformações de objetos em C# para cadeias de caracteres que podem ser transferidas como dados para o Arduino através da rede.

Há também uma biblioteca em C++, que pode ser utilizada no Arduino e a mesma faz as transformações necessários em JSON para que cadeias de caracteres sejam transformadas em objetos do C++. Escrita por Marcus Nowotny, pode ser facilmente implementada em qualquer projeto Arduino.

Na utilização das duas bibliotecas, foi feita a transferência de dados pela rede utilizando os benefícios do JSON, com a existência do padrão, existe uma facilidade grande em obter bibliotecas em várias linguagens que fornecem métodos de serialização, ou seja, a transformação dos objetos em cadeias de caracteres e também a desserialização, o oposto da serialização, transforma cadeias de caracteres em objetos da linguagem em uso.

Figura 30 – Classe exemplar de uma pessoa

```

1 public class Pessoa {
2     public string Nome { get; set; }
3     public int Idade { get; set; }
4 }

```

Fonte: Do autor

Uma classe representando uma pessoa (figura 30) fora implementado em C# com intuito de testar a comunicação com a ajuda do JSON. Uma instância da classe foi criada como na figura 29.

Figura 31 – Instância da classe pessoa

```

1 Pessoa pessoa = new Pessoa();
2 pessoa.Nome = "Igor";
3 pessoa.Idade = 21;

```

Fonte: Do autor

A serialização da instância do objeto pessoa da figura 31, resultou em uma cadeia de caracteres como da figura 32.

Figura 32 – Objeto serializado em JSON

```
1 { "Nome" : "Igor", "Idade" : 21 }
```

Fonte: Do autor

Os resultados obtidos forneceu grande facilidade no controle dos dados para serem transmitidos durante a comunicação da rede, mas resultou em uma carga muito grande na quantidade de caracteres quando transmitidos objetos mais complexos envolvendo vários campos ou até mesmo objetos que se encontravam dentro de outros objetos, resultando em uma imensa cadeia de caracteres.

Essa extensa informação que era transmitida pela rede requeria também um processamento maior do Arduino para transforma-los em objetos e só então era possível usar as informações para realizar as funções do sistema.

Há também um problema escrito por Benoit Blanchon em seu blog, ele diz que tal biblioteca abusa da técnica de alocar memória dinâmica sempre que preciso, e isso pode ocasionar erros durante a execução do sistema. É oferecido uma outra biblioteca em seu blog, entretanto, tal biblioteca não oferece recursos suficientes para todas as operações de JSON.

Com base nas tentativas, o projeto necessitava de outra técnica, algo que tivesse simplicidade e ao mesmo tempo capacidade de não requerer tanto processamento do Arduino.

Na próxima tentativa, da qual ficou como definitiva, foi separar consecutivamente as propriedades dos objetos separados por ponto e vírgula. Com o novo método, o mesmo objeto da figura 31, em sua forma serializada ficaria de tal forma como na figura 33.

Figura 33 – Objeto serializado com método próprio

```
1 Igor;21
```

Fonte: Do autor

Com a cadeia de caracteres gerada, é possível notar que as propriedades são serializadas de tal forma que correspondem consecutivamente suas ordens naturais. Na serialização da figura 31, primeiro vem o nome da pessoa e logo após

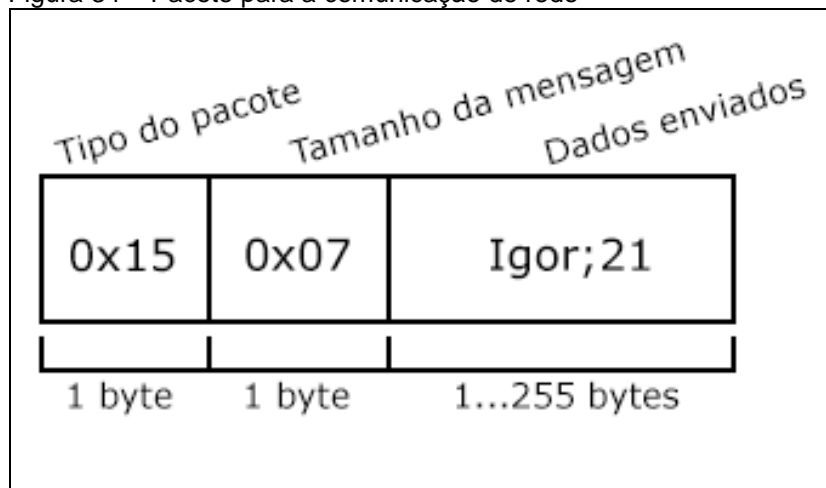
sua idade, dessa forma, quando é feito a desserialização, é obrigatório que a primeira propriedade seja o nome e a segunda sua idade.

Neste novo método, o Arduino não precisava de muitos recursos, já que não usava nenhum processo de busca de propriedades, simplesmente transformava-as sequencialmente. É também descartado apenas o ponto e vírgula ao contrário de caracteres como as aspas, espaços, chaves, pontos duplos e vírgulas como no JSON.

A maneira de como os dados são transmitidos pela rede não significa que o Arduino saberá o que fazer com estes dados, para isso é preciso de uma maneira para que o gestor avise-o sobre o que fazer com esta informação recebida.

Um protocolo básico foi criado, e pode ser explicado com a ajuda da figura 34, sendo esta figura um exemplo de comando para cadastrar a pessoa serializada da figura 33. O formato dos dados para o tipo do pacote e tamanho da mensagem estão em hexadecimal, os dados enviados também são enviados em hexadecimal, mas foram postos em caracteres para uma melhor leitura da imagem.

Figura 34 – Pacote para a comunicação de rede



Fonte: Do autor

Na figura 34, encontra-se um pacote de comunicação para que o comando seja enviado ao Arduino, este que irá manusear os dados recebidos e realizar alguma ação. O tipo de pacote nada mais é que uma identificação deste pacote, no caso citado é informado, através do tipo, que os dados ali contidos são destinados a um cadastro de uma nova pessoa.

A leitura dos dados da rede são efetuadas sequencialmente e infinitamente, ou seja, o Arduino lê todos os *bytes* recebidos através do seu módulo de rede, desta forma, para restringi-lo a ler apenas os bytes que pertencem ao pacote que está a receber, foi necessário dizer-lhe esta quantidade. O dever do pacote então é de enviar juntamente consigo seu próprio tamanho, citado como tamanho da mensagem na figura 34.

4.1.6.4 Persistência dos dados para o Arduino

O sistema agora precisava de maneiras para persistir dados. No Ethernet Shield, há um espaço para colocar um cartão miniSD, onde é possível salvar arquivos usando a biblioteca disponibilizada pela equipe do Arduino.

Para a distribuição dos arquivos, foi definido que eles se encontrarão dentro de pastas que os identificam, pois na abertura de arquivos, é preciso rastrear uma pasta para obter os nomes dos arquivos, já que esses nomes serão criados de forma dinâmica. Por se tratar de um microcontrolador que não possui alto desempenho, é necessário que essa parte do sistema também seja minimizada para não criar processos longos, dessa forma, um sistema de pastas foi criada para o problema.

Há no cartão, uma pasta para os dados de colaboradores, para os dados de cartões RFID e outra para os pontos registrados durante o uso do sistema. Na pasta responsável pelos pontos, o sistema irá criar uma pasta para os anos e na pasta de cada ano, haverá uma pasta para cada mês do ano, dessa forma, haverá um controle simples e lógico, conseguindo minimizar os processos de buscas. Quando necessitar de um relatório de um mês específico de um ano, o sistema não precisará realizar nenhuma busca, simplesmente irá ler todos os arquivos dentro da pasta que pertence a aquele mês.

Foram feitas as primeiras gravações de pontos, após o sucesso do cadastramento de funcionários e suas respectivas etiquetas para identificação. Os pontos são gravados no miniSD, e futuramente precisam dos mesmos para gerar relatórios durante o mês, este serviço é servido exclusivamente pelo programa do Windows.

Durante o desenvolvimento de relatórios, sendo que era necessário consolidar todos os pontos de um único mês para que posteriormente fosse impresso em uma espécie de grade de horários realizados neste período, foi constatado que o espaço de desenvolvimento usando o Visual Studio era complexo ao ponto de preferir métodos mais simples e menos robusto para o desenvolvimento.

Foi realizado o download de uma biblioteca para o .NET que oferece suporte para a criação de documentos no formato PDF. Desenvolvido pela empresa Empira, distribuído gratuitamente, o PDFsharp pode ser usado para este objetivo.

A biblioteca consiste em oferecer a possibilidade de realizar impressões de arquivos PDF com suporte a formatação de texto, tabelas, títulos e outras funcionalidades. Todo o processo é feito via código e nenhuma interface é usada durante a construção de impressões, diferentemente do sistema de relatórios que já existe no Visual Studio, que conta com algumas interfaces visuais para a criação de relatórios.

4.1.7 Materiais utilizados

Na tabela 1 é descrito todos os itens com seus valores em moeda brasileira, o real. Alguns dos itens foram compradas com moedas estrangeiras, dessa forma o valor é atualizado com a cotação no momento da compra. Estes itens são aqueles que definem o projeto final e não é incluído nenhum item que fora comprado para testes ou aprendizado.

Tabela 1 – Preços dos componentes utilizados no projeto final

Descrição do componente	Qtd.	Valor unitário (reais)	Valor total (reais)
Módulo RFID/NFC ElecFreaks	1	67,68	67,68
Ethernet Shield	1	31,56	31,56
Arduino MEGA 2560	1	46,25	46,25
Display LCD I2C	1	16,47	16,47
Módulo RTC	1	7,89	7,89
Total			169,85

Fonte: Do autor

Os itens que foram destinados ao aprendizado, testes e outros fins, podem ser encontrados na tabela 2.

Tabela 2 – Preços de componentes extras para aprendizado

Descrição do componente	Qtd.	Valor unitário (reais)	Valor total (reais)
Kit iniciante para Arduino	1	116,42	116,42
RFID-RC522	2	28,65	57,30
Total			173,72

Fonte: Do autor

Conforme visto na tabela 1, o valor total dos itens necessários para produzir um sistema como este, retirando os custos de qualquer mão de obra, foi de 169,85 reais.

4.2 RESULTADOS OBTIDOS

O trabalho desenvolvido teve os objetivos atendidos como definidos. O REP construído tem a finalidade de persistir os dados de funcionários, do empregador e dos pontos marcados, esses pontos são persistidos através da identificação de um funcionário já cadastrado juntamente ao número da etiqueta para que o módulo RFID o identifique.

O estudo sobre Arduino e RFID possibilitou ter conhecimento a respeito de como desenvolver o sistema por inteiro, assim sendo possível também definir quais funções tem cada componente no sistema.

Para o último objetivo, um sistema para o Windows, utilizando como ferramenta de desenvolvimento o Visual Studio, foi construído para realizar a gestão do REP, habilitando o usuário a cadastrar os funcionários, definir o empregador, visualizar os pontos marcados, e quando gerar os relatórios, excluir os pontos que foram marcados de forma errada ou incluir pontos que foram esquecidos.

O sistema como um todo teve seu comportamento de forma simples como pensado. O funcionário não precisa realizar nenhum evento complexo para que seu ponto seja marcado no sistema, ele apenas precisa ter uma etiqueta para sua identificação no momento em que seu turno tiver início e fim. Para isso, é preciso aproximar seu identificador cadastrado ao sistema, o mesmo deve acontecer ao final do seu turno.

Para a usabilidade do sistema ser possível para qualquer empregador doméstico, foi definido que seu custo final deveria ser baixo, sendo que uma maior gama de nível social pudesse adquirir tal produto, pois os custos finais seriam mais baixos do que os praticados no mercado.

5 CONCLUSÃO

Este protótipo desenvolvido é um sistema que facilita o modo de controle das marcações de ponto, para o início do projeto foi definido que esse fosse destinado a funcionários domésticos, porém o sistema pode facilmente ser integrado em outros ambientes de trabalho.

O projeto conta com um sistema totalmente funcional, mas para seu uso ser válido para a justiça, o mesmo deveria ser avaliado e homologado pelo Ministério do Trabalho. Para essa homologação, o sistema deve contar com um módulo adicional. Contudo o uso não é proibido, sendo livre para o uso doméstico, dessa forma habilitando o empregador controlar normalmente os horários de seus funcionários domésticos.

Com o protótipo não podendo ser homologado pelo Ministério do Trabalho, não é possível compará-lo como um produto final diante os outros disponíveis no mercado.

Em geral, o protótipo fornece gestão total para o empregador, podendo ao final do mês ter uma visibilidade de todos os pontos marcados pelo funcionário doméstico, e assim sendo possível gerar um relatório com facilidade para que faça o pagamento do colaborador.

Como todo projeto, várias dificuldades surgiram durante seu desenvolvimento, algumas foram superadas com muito estudo de grandes livros da área, outras com muitas tentativas e erros.

5.1 TRABALHOS FUTUROS

Durante a construção do projeto como um todo, foi pensado várias implementações, mas por se tratarem de ideias de funcionalidades maiores, foram deixadas como trabalhos futuros. É destacável a ideia de ter avisos para o empregador quando o funcionário doméstico tivesse seu ponto marcado, seja ele de entrada, saída ou para intervalos, esses avisos poderiam ser para um smartphone, caso tivesse, se não, através de SMS. Foi levantado também, como trabalho futuro,

ter a implementação de um sistema para enviar os relatórios diretamente para a contabilidade do empregador.

REFERÊNCIAS

- ANATEL. **Plano de Atribuição, Destinação e Distribuição de Faixas de Frequências no Brasil**. 2013. p. 170. Disponível em: <<http://www.anatel.gov.br/Portal/exibirPortalInternet.do>>. Acesso em: 4 nov. 2013.
- BANZI, Massimo. **Getting Started with Arduino**. 2 ed. Sebastopol: O'Reilly, 2011.
- BRASIL. Constituição (1988). Emenda constitucional n.º 72, de 2 de abril de 2013. Altera a redação do parágrafo único do art. 7 da constituição federal.
- _____. Decreto-Lei n.º 5.859, de 11 de dezembro de 1972. Dispõe sobre a profissão de empregado doméstico e dá outras providências.
- _____. Decreto-Lei n.º 7.855, de 24 de outubro de 1989. Altera a consolidação das leis do trabalho.
- _____. Ministro de Estado do Trabalho e Emprego. Portaria n.º 1.510, de 21 de agosto de 2009.
- _____. **Trabalhador Doméstico**. Brasília, 2013. p. 22. Disponível em: <http://portal.mte.gov.br/trab_domestico/trabalho-domestico.htm>. Acesso em: 3 nov. 2013.
- CLEMENTE, Graziela Candici; DESCALZO, Daniel; SILVA, Flávia Andressa Jacinto da. RFID: Um estudo de tecnologias disponíveis para análise de viabilidade e utilização e montagem de protótipo para demonstração e testes. 2011. 92 f. Trabalho de Conclusão de Curso (Graduação) - Curso Superior de Tecnologia em Eletrônica - Comunicações e Tecnologia em Sistemas de Telecomunicações, Universidade Tecnológica Federal do Paraná (UTFPR), Curitiba. Disponível em: <http://repositorio.roca.utfpr.edu.br/jspui/bitstream/1/882/1/CT_COTEL_2011_1_01.pdf>. Acesso em: 16 nov. 2012.
- FINKENZELLER, Klaus. **RFID Handbook: Fundamentals and Applications in Contactless Smart Cards, Radio Frequency Identification and Near-Field Communication**. 3 ed. Chichester: John Wiley and Sons, 2010.
- GRIFFITHS, Ian; ADAMS, Matthew; LIBERTY, Jesse. (2001). **Programming C# 4.0: Building Windows, Web and RIA Applications for .NET with C# 4.0**. 6. Sebastopol: O'Reilly, 2010. 856 p.
- HUNT, V et al. **RFID: A guide to radio frequency identification**. New Jersey: John Wiley and Sons, 2007.
- IBRAHIM, Dogan. **Using LEDs, LCDs, and GLCDs in microcontroller projects**. Chichester: John Wiley and Sons, 2012.

IGOE, Tom. **Getting Started with RFID**. Sebastopol: O'Reilly, 2011.

JOHNSON, Bruce. **Professional Visual Studio 2012**. Canada: Wrox, 2012. 1104 p.

MACKEY, Alex; TULLOCH, William; KRISHNAN, Mahesh. **Introducing .NET 4.5**: Discover what's new in the newest version of microsoft's .NET Framework. 2012. [S. l.]: Apress, 2012. 408 p.

MARGOLIS, Michael. **Arduino Cookbook**. 2. ed. Sebastopol: O'Reilly, 2011.

MAZIDI, M et al. **The AVR microcontroller and embedded system**: using assembly and c. New Jersey: Prentice Hall, 2011.

MCROBERTS, Michael. **Beginning Arduino**: Create simple but practical projects with arduino using step-by-step instructions and easy-to-follow diagrams, no experience necessary. 2 ed. [S. l.]: Apress, 2013.

NAGEL, Christian et tal. **Professional C# 4 and .NET 4**. Canada: Wiley, 2010. 1852 p.

_____. **Professional C# 2012 and .NET 4.5**. Canada: Wiley, 2013. 1588 p.

NETTO, Luiz Ferraz. Resistores. 1999. Disponível em: <http://www.feiradeciencias.com.br/sala12/12_T02.asp>. Acesso em: 11 maio 2014

SARMENTO, Eduardo Mendes. Controle de presença utilizando RFID: um estudo de caso utilizando a linguagem Ruby. 2011. 25 f. Trabalho de Conclusão de Curso (Graduação) - Universidade Tecnológica Federal do Paraná, Medianeira, 2011. Disponível em: < <http://repositorio.roca.utfpr.edu.br/jspui/handle/1/610> >. Acesso em: 9 nov. 2012.

SCHMIDT, Maik. **Arduino: A Quick-Start Guide**. [S. l.]: Pragmatic Programmers, 2011.

SHELDON, Bill et tal. **Professional Visual Basic 2012 and .NET 4.5 Programming**. Canada: Wiley, 2013. 1588 p.

SINGH, Abhinav. **Wireshark Starter**: A quick and easy guide to getting started with network analysis using Wireshark. Reino Unido: Packt Publishing, 2013. 55 p.

SOUZA, David. **Desbravando o PIC**: Ampliado e atualizado para PIC16F628A. 8 ed. São Paulo: Editora Érica, 2003.

STELLMAN, Andrew; GREENE, Jennifer. **Head First C#, 2nd Edition**: A Learner's Guide to Real-World Programming with Visual C# and .NET. Canada: O'Reilly, 2010. 848 p.

SUR, Abhishek. **Visual Studio 2012 and .NET 4.5 Expert Development Cookbook**: Over 40 recipes for successfully mixing the powerful capabilities of .NET 4.5 and Visual Studio 2012. Reino Unido: Packt Publishing, 2012. 380 p.

SWEENEY II, Patrick. **RFID For Dummies**. Indianapolis: Wiley Publishing, 2005.

TEIXEIRA, Tiago. Controle de Fluxo de Pessoas Usando RFId. 2011. 73 f. Trabalho de Conclusão de Curso (Graduação) - Curso de Tecnologia em Sistemas de Telecomunicações, Instituto Federal de Santa Catarina, São José, 2011. Disponível em: <http://wiki.sj.ifsc.edu.br/wiki/images/f/fa/TCC_TiagoTeixeira.pdf>. Acesso em: 14 nov 2012.

TROELSEN, Andrew. **Pro C# 5.0 and the .NET 4.5 Framework**: Exploring the .NET universe using curling brackets. 6. [S. l.]: Apress, 2012. 1560 p.

APÊNDICES

APÊNDICE A – Artigo

Desenvolvimento de Sistema Para Controle de Horários de Funcionários Domésticos com Identificação por RadiofrequênciaSérgio Coral¹, Igor Henrique P. Fontana²¹Professor Especialista – Ciência da Computação – Universidade do Extremo Sul Catarinense (UNESC)²Ciência da Computação - Universidade do Extremo Sul Catarinense (UNESC)

sergiocoral@unesc.net, igor.fontana@hotmail.com

***Abstract.** As new laws for domestic workers were introduced, new obligations emerged in order to equalize workers rights, turning this kind of job into one easier to be supervised using appropriate electronic devices. However, it may be expensive to be used in the domestic sphere. This work aims to build an electronic system for this specific scenario.*

***Resumo.** Novas legislações surgiram para o empregado doméstico, igualando seus direitos. O empregador recebeu direito de controlar a jornada de trabalho do trabalhador, possibilitando usar qualquer sistema para este fim. Os sistemas que estão no mercado tem um custo alto para os empregadores, dessa forma este trabalho tem o objetivo de construir um sistema eletrônico de marcação de ponto com um custo acessível.*

1. Introdução

A alteração da lei que promove os direitos trabalhistas permitiu que o empregado doméstico passe a ter novos direitos (BRASIL, 2013). Desta forma, um trabalhador doméstico tem os direitos iguais aos outros trabalhadores. Os empregadores também têm direitos iguais, e um desses é o de fiscalizar e assegurar que o trabalhador esteja cumprindo seus deveres (BRASIL, 1989). Com a ideia de tornar a jornada de trabalho do trabalhador doméstico fiscalizada, a tecnologia permite que de forma automatizada, sua jornada seja acompanhada de forma eletrônica por meio de um sistema eletrônico para registros de pontos.

Existem várias formas de construir um sistema para o registro de pontos, principalmente para a forma de como é identificado o empregado, podendo ser por senha, leitura biográfica, identificação por radiofrequência (RFID), entre outros. A tecnologia de identificação por radiofrequência surgiu para um propósito similar ao do código de barras, porém pode ser aplicado em inúmeras ideias compatíveis.

Identificação por rádio frequência é um sistema que permite a comunicação de dados sem fios que permite os usuários identificar objetos ou seres vivos que terão etiquetas únicas (HUNT, 2007, tradução nossa). Podendo por exemplo, em um supermercado, saber quais produtos o cliente tem em seu carrinho de compras, e automaticamente mostrar o resultado da compra. (FINKENZELLER, 2010, tradução nossa).

Segundo Finkenzeller (2010, tradução nossa), um sistema de identificação por rádio frequência necessita de um controlador para consumir as informações lidas pelo sistema, pois o mesmo apenas realiza as leituras das etiquetas e entrega para alguma aplicação responsável.

Segundo Banzi (2011, tradução nossa), o funcionamento de componentes eletrônicos e suas teorias são de difícil entendimento, o que levaria muito tempo para compreendê-las e como utilizá-las. Com um simples kit de desenvolvimento de circuitos, é possível realizar trabalhos e protótipos com maior facilidade. Por esses motivos, e muitos outros, Arduino foi construído como um projeto aberto que nada mais é do que uma plataforma de computação física baseada em um sistema simples de entrada e saída em uma placa.

2. Justificativa

O desenvolvimento de sistemas usando a plataforma Arduino oferece uma facilidade enorme, tanto para iniciantes, ou para pessoas experientes, pois é justamente uma plataforma que é construída para tornar o processo de prototipagem de sistemas mais rápido e para proporcionar uma maior facilidade para o aprendizado de eletrônica para iniciantes (BANZI, 2011, tradução nossa). Arduino, como uma plataforma open-source, é um dos mais utilizados ultimamente para projetos caseiros, e também projetos acadêmicos. Isso mostra que com o uso do Arduino, grandes conhecimentos podem ser alcançados com prática de desenvolvimento.

RFID, como um sistema de identificação única, é possível utilizá-lo em inúmeras aplicações, em estacionamentos para computação do tempo permanecido, em supermercados para identificar todos os itens em um carrinho de compras (IGOE, 2012). Esse tipo de sistema mostra facilidade de aplicá-lo, e uma enorme comodidade para as pessoas que o utilizam, pois ao se tratar de uma pequena etiqueta que identifica os objetivos, ela pode ser usada praticamente em qualquer lugar em qualquer formato (FINKENZELLER, 2010, tradução nossa).

O Arduino possui uma capacidade de receber *shields* acoplados para terem um funcionamento adicional, esses *shields* possuem várias finalidades, e alguns deles possuem a funcionalidade de atuarem como leitores de RFID. Esses componentes extras, normalmente, oferecem maior facilidade do que componentes comuns voltados à microcontroladores em geral, pois para o Arduino, existem bibliotecas prontas que facilitam a programação do sistema para o manuseio das informações de RFID (IGOE, 2012, tradução nossa).

Com um sistema RFID aplicado ao Arduino, é possível desenvolver um sistema que será capaz de realizar registros de entradas e saídas, bem como armazená-las em uma memória permanente e com um computador será possível realizar relatórios diários e também mensais, fornecendo uma maneira para efetuar pagamento do funcionário. O sistema deve ser simples e proporcionar facilidade para os funcionários domésticos, deve ser desenvolvido desta forma para evitar problemas com pessoas que possuem dificuldades com equipamentos e tecnologias recentes.

3. Fundamentação Teórica

3.1 Legislação e Lei

Existem várias legislações para trabalhadores, empregadores e ultimamente temos as novas normas para os empregados domésticos que até então não tinham direitos iguais aos outros

trabalhadores. O conhecimento sobre as regras são importantes para o desenvolvimento do projeto, pois também existem legislações para os meios de fiscalizações.

3.1.1 Lei do Empregado Doméstico

No início de Abril de 2013, houve uma alteração na redação do parágrafo único do art. 7 da Constituição Federal pela emenda constitucional n.º 72. Até então, pessoas que trabalhavam em residências prestando serviços não possuíam direitos trabalhistas iguais aos outros trabalhadores urbanos e rurais. A partir da alteração, foi estabelecido essa igualdade, de que trabalhadores domésticos teriam igualdade de direitos entre os outros tipos de trabalhadores. Com a alteração, todo empregador deve fornecer os direitos para o trabalhador doméstico que agora surgiram, bem como tem seus direitos de controle e fiscalização (BRASIL, 2013).

Segundo a lei n.º 5.859, os empregados domésticos são aqueles que prestam serviços de natureza não-econômica para uma pessoa ou uma família, e estes serviços são do tipo residencial (BRASIL, 1972). Conforme os termos, são exemplos de ocupações dos empregados domésticos: empregado, cozinheiro, governanta, babá, lavadeira, faxineiro, vigia, motorista particular, jardineiro, acompanhante de idosos, dentre outros (BRASIL, 2013).

3.1.2 Lei da Jornada de Trabalho

Segundo a lei n.º 7.855, a jornada de trabalho pode ser fiscalizada por meios de registros manuais, mecânicos ou eletrônicos, porém, as instruções dos métodos utilizados devem ser expedidos pelo Ministério do Trabalho. Contudo, o principal é ter anotação de cada entrada e saída. Para estabelecimentos com mais de dez trabalhadores, algum dos métodos citados devem ser obrigatoriamente implantados para uso (BRASIL, 1989).

Conforme a lei, o método eletrônico é possível ser usado para controlar a jornada de trabalho dos colaboradores, o método pode ser usado em qualquer ocasião. Em um exemplo, é possível ter um empregado doméstico sem ter sua jornada de trabalho fiscalizada, já que a casa apenas teria este único colaborador, contudo, fica a disponibilização do uso do método.

3.2 Arduino

Uma placa Arduino é um pequeno computador que é possível programá-lo para processar entradas e saídas de dados entre dispositivos e componentes externos que é conectado à placa. Arduino é conhecido como uma plataforma em sistemas embarcados. Sendo possível usá-lo para ligar uma lâmpada por um período de tempo, ou após pressionar um botão. Este mesmo conceito poderia ser estendido para algo ainda maior e fazê-lo com que detecte a presença de alguém em uma sala, e fazer com que a lâmpada dessa sala seja acessa pelo sistema (BANZI, 2011, tradução nossa).

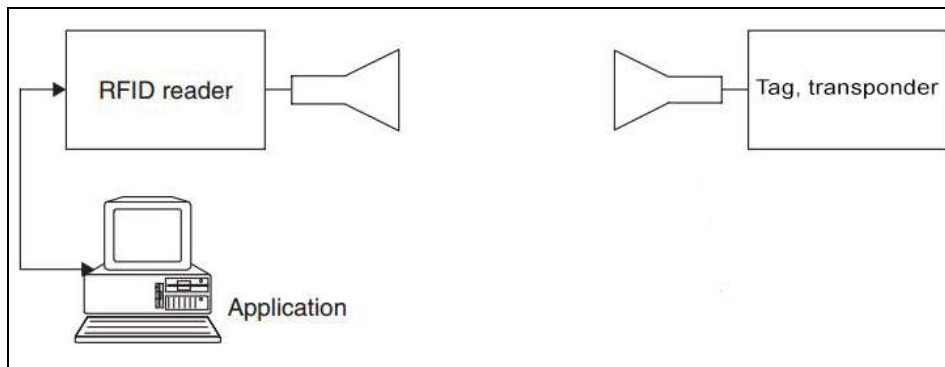
3.2.1 Shields

O Arduino pode ser estendido com outros sistemas embarcados, os *shields*, que são placas que contém outras funções de dispositivos, tais como, receptor GPS, *display* em *Liquid Crystal Display* (LCD), módulos de *Ethernet*, entre outros, que podem ser simplesmente encaixados na placa de um Arduino para conseguir sua função extra (MCROBERTS, 2011, tradução nossa).

3.3 Identificação por Radiofrequência

Assim como o código de barras, o RFID em forma de etiqueta, permite que objetos sejam identificáveis. Porém, de uma forma diferente que os códigos de barras (IGOE, 2012, tradução nossa).

Em geral, RFID é uma tecnologia de ótimo rendimento e pode ser usado como uma ótima ferramenta em vários casos fazendo papel para identificação e até mesmo substituindo a tecnologia de identificação que mais é usada nos dias de hoje, que neste caso, pode ser vista como o sistema de código de barras (SWEENEY II, 2005, tradução nossa).



2 Figura 1. Exemplo de sistema RFID

Os sistemas de RFID (Figura 1), possuem um leitor que se comunica com *tags* pelo ar com determinada frequência, como qualquer outra comunicação de rádio. O RFID é constituído basicamente por leitores, antenas, *tags*, e frequência. Os leitores funcionam de maneira parecida com os rádios de carros. O leitor produz eletricidade que corre por um fio até chegar na antena, a partir daí essa eletricidade radia o mesmo sinal digital pelo espaço com uma certa frequência e mesmo comprimento de onda. Não apenas gera sinal que vai da antena para o espaço, mas também fica a escuta de uma resposta que será produzida pela *tag*. Dessa forma, o leitor transmite e recebe ondas analógicas e as transforma em informação digital de zeros e uns (SWEENEY II, 2005, tradução nossa).

4. Desenvolvimento

O trabalho teve objetivo principal de construir um protótipo para ter o funcionamento de um sistema para registrar os pontos da jornada do trabalhador de forma eletrônica, sendo possível identificar os colaboradores com um sistema de leitura de etiquetas com radiofrequência. Para controlar o sistema embarcado, um software construído com o .NET Framework tem o poder de gestão.

4.1 Metodologia

4.1.1 Aprendizado e Compreensão do Arduino

O processo de compreensão sobre desenvolver protótipos com o Arduino requereu alguns componentes. Para o início, foi necessário um *kit* de desenvolvimento próprio para iniciante (Figura 2), esse que ajudaria nas questões de aprendizado.



3 Figura 2. Kit de Arduino para iniciantes

Com base no site oficial do Arduino, onde há vários guias para os iniciantes do uso da plataforma, foi dado os primeiros passos de estudo da plataforma. Um dos exemplos mais básico, é um guia onde aprende-se como utilizar o microcontrolador para controlar um pequeno emissor de luz. Posteriormente, os guias levam qualquer um a ter conhecimento de como controlar componentes mais complexos, tais como telas de LCD.

4.1.2 Hardware Necessário

O primeiro item a ser selecionado foi o leitor das etiquetas de RFID. O leitor não precisava ser potente, nem que fosse habilidoso em realizar leituras de longas distâncias, essas características contribuíram para que o leitor escolhido tivesse um valor com baixo custo. RFID-RC522 (Figura 3) foi o leitor escolhido, o vendedor forneceu uma etiqueta no formato de chaveiro e uma outra com o formato de cartão de crédito.

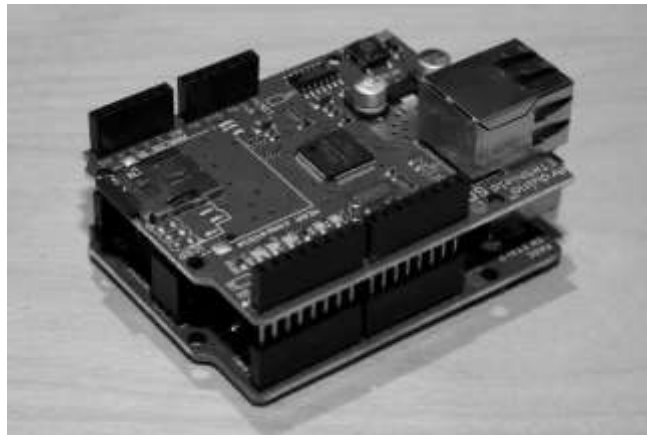
Devido à complexidade do módulo RFID, o vendedor fornece uma biblioteca para facilitar sua comunicação com o Arduino, basta carregar do site para o computador e utiliza-lo juntamente a programação do sistema. Para utilizar o módulo leitor, foi necessário pesquisar alguns guias de como conectar os pinos do módulo ao Arduino e realizar as primeiras leituras.



4 Figura 3. Módulo RFID-RC522

Outros dois componentes foram necessários para se comunicar pela rede, dessa forma realizando a troca de dados entre o Arduino e um computador que seria responsável por ter o software gestor e realizar sua gestão, o outro componente deveria ser capaz de persistir dados dos colaboradores, pontos, entre outros.

Com base na requisição, a escolha não poderia ser mais óbvia do que o Ethernet Shield, sabendo-se que o Arduino suporta componentes que encaixam-se perfeitamente em sua superfície, neste caso, os *shields*. Foi possível adquirir esse shield que tinha a capacidade de atender as duas requisições ao mesmo tempo. Para a persistência dos dados, o *shield* conta com um espaço para usar cartões *miniSD*.

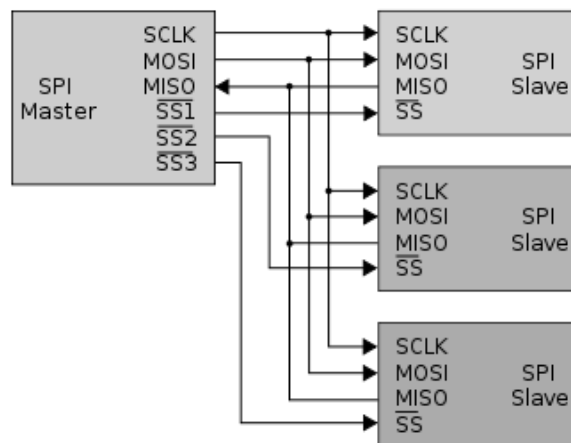


5 Figura 4. Ethernet Shield acoplado em um Arduino UNO

Todos esses componentes necessitavam de um conhecimento sobre os barramentos utilizados no Arduino.

4.1.3 Barramentos de Comunicações

O Serial Peripheral Interface (SPI) é uma interface que permite a comunicação de um microcontrolador com diversos outros componentes, formando uma rede de periféricos como no exemplo da figura 5. A principal vantagem do barramento é a economia do uso portas de entrada e saída de dados.



6 Figura 5. Rede de componentes no barramento SPI

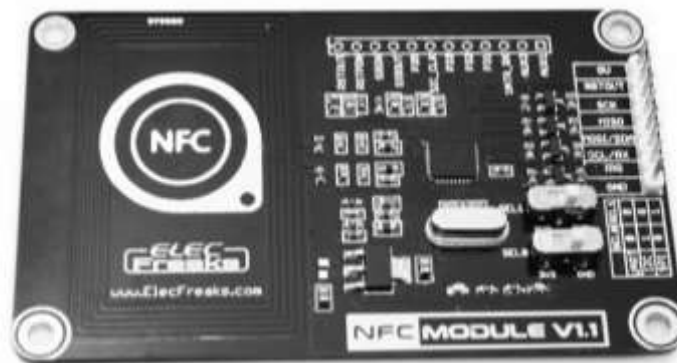
Há a existência de outro barramento na plataforma do Arduino. O *Inter-Integrated Interface* (IIC) ou *Two Wired Interface* (TWI) tem uma funcionalidade bastante semelhante a SPI. A diferença está na economia de portas, sendo que é superior a economia feita pelo SPI.

Os estudos sobre os barramentos de comunicações usados pelo Arduino é muito importante, isso habilita o uso de vários componentes usando pouquíssimos pinos para o funcionamento dos mesmos. Com a ajuda dos barramentos, foi possível conectar o Ethernet Shield juntamente com o leitor do RFID na comunicação via SPI.

Mesmo com o sucesso de conectar os dois componentes ao Arduino e realizar operações básicas utilizando esses barramentos, o módulo de RFID apresentou problemas durante o funcionamento do mesmo, como dificuldade de leitura de algumas etiquetas ou até mesmo não realizar leitura alguma.

4.1.4 Componentes Para o Barramento I2C

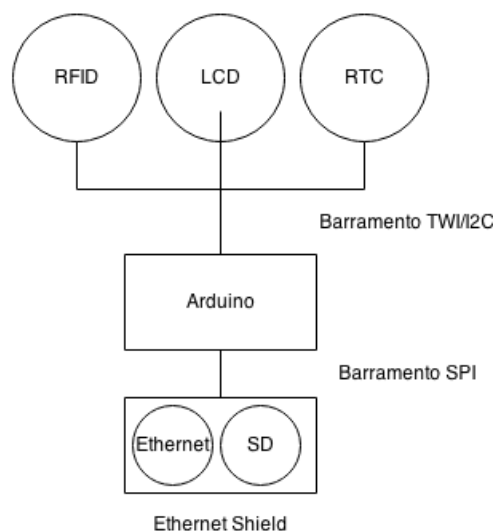
Diante do problema do módulo atual do RFID, foi necessário buscar outro módulo RFID (Figura 6), com mais qualidade mas não necessariamente com mais funções.



7 Figura 6. Módulo RFID/NFC ElecFreaks

Foi selecionado um display LCD, o LCM11602 que é compatível com I2C e também um Real Time Clock (RTC) com a mesma compatibilidade.

Com todos os componentes em mãos, o modelo do sistema foi definido (Figura 7).



8 Figura 7. Modelo do sistema

4.1.5 Construção Lógica do Sistema e do Software

4.1.5.1 Comunicação de Dados

Com o objetivo de simplificar a transmissão de dados pela rede, foi realizada uma tentativa para utilizar de uma serialização de dados muito usada em sistemas atuais, o *JavaScript Object Notation* (JSON).

Seu uso ofereceu grande facilidade, mas trouxe grande perda de desempenho para o sistema embarcado. Sistemas como estes tem limitações, e é necessário respeitá-las. Uma forma alternativa foi implementar um método próprio para serialização dos objetos transferidos pela rede local.

Foi utilizado como base o estilo de serialização *Comma-separated values* (CSV) e realizado algumas mudanças. No sistema é possível escolher qual o caractere correspondente a separação dos valores e dessa forma foi escolhida o ponto e vírgula como padrão, justamente para não dar conflitos quando valores conter vírgulas.

A maneira de como os dados são transmitidos pela rede não significa que o Arduino saberá o que fazer com estes dados, para isso é preciso de uma maneira para que o gestor avise-o sobre o que fazer com esta informação recebida.

Um protocolo básico foi criado. Primeiramente é enviado o tipo do pacote, sendo que aponta para a função será executada em ambos os lados, após o envio deste primeiro *byte* é enviado um outro *byte* contendo o tamanho de *bytes* a serem lidos que corresponderão aos dados utilizados pela função. Como o tamanho é apenas um *byte*, o tamanho de dados fica limitado a 255 *bytes*.

4.1.5.2 Persistência dos Dados e Relatórios

O sistema agora precisava de maneiras para persistir dados. No Ethernet Shield, há um espaço para colocar um cartão miniSD, onde é possível salvar arquivos usando a biblioteca disponibilizada pela equipe do Arduino.

Foi realizado uma distribuição de arquivos em pastas descritivas para os mesmos arquivos, como o mês que se refere aos pontos que ali dentro estão persistidos. Para cada entidade é criado uma pasta, os nomes dos arquivos são seus números de identificação e seu interior contém os valores.

Todos os dados são gerenciados pelo software criado com o Visual Studio. É possível cadastrar todos os empregados, associar uma *tag* ao empregado, visualizar todos os pontos marcados durante um período, e por fim gerar os relatórios das marcações de pontos, bem como corrigi-las informando sempre um motivo. Não é possível excluir pontos marcados pelo sistema, apenas marca-los como inativos, tal funcionalidade foi desenvolvida dessa forma para diminuir a chance de negligência.

Para que fosse possível gerar relatórios com resultado em .PDF imprimíveis foi utilizada uma biblioteca desenvolvida pela empresa Empira, distribuído gratuitamente com o nome PDFSharp.

4.1.6 Materiais utilizados

Estes itens da tabela 1, são aqueles que definem o projeto final e não é incluído nenhum item que fora comprado para testes ou aprendizado.

Descrição do componente	Qtd.	Valor unitário (reais)	Valor total (reais)
Módulo RFID/NFC ElecFreaks	1	67,68	67,68
Ethernet Shield	1	31,56	31,56
Arduino MEGA 2560	1	46,25	46,25
Display LCD I2C	1	16,47	16,47
Módulo RTC	1	7,89	7,89
Total			169,85

9 Tabela 1. Preços dos componentes do protótipo final

5. Conclusão

Este protótipo desenvolvido é um sistema que facilita o modo de controle das marcações de ponto, para o início do projeto foi definido que esse fosse destinado a funcionários domésticos, porém o sistema pode facilmente ser integrado em outros ambientes de trabalho.

O projeto conta com um sistema totalmente funcional, mas para seu uso ser válido para a justiça, o mesmo deveria ser avaliado e homologado pelo Ministério do Trabalho. Para essa homologação, o sistema deve contar com um módulo adicional. Contudo o uso não é proibido, sendo livre para o uso doméstico, dessa forma habilitando o empregador controlar normalmente os horários de seus funcionários domésticos.

Com o protótipo não podendo ser homologado pelo Ministério do Trabalho, não é possível compará-lo como um produto final diante os outros disponíveis no mercado.

Em geral, o protótipo fornece gestão total para o empregador, podendo ao final do mês ter uma visibilidade de todos os pontos marcados pelo funcionário doméstico, e assim sendo possível gerar um relatório com facilidade para que faça o pagamento do colaborador.

7. Referências

Banzi, M. (2011). “Getting Started with Arduino”. 2 ed. Sebastopol: O’Reilly.

Brasil (1988). Constituição. Emenda constitucional n.º 72, de 2 de abril de 2013. “Altera a redação do parágrafo único do art. 7 da constituição federal.”

____ (1972). Decreto-Lei n.º 5.859, de 11 de dezembro de 1972. “Dispõe sobre a profissão de empregado doméstico e dá outras providências.”

____ (1989). Decreto-Lei n.º 7.855, de 24 de outubro de 1989. “Altera a consolidação das leis do trabalho.”

Finkenzeller, K. (2007). “RFID Handbook: Fundamentals and Applications in Contactless Smart Cards, Radio Frequency Identification and Near-Field Communication”. 3 ed. Chichester: John Wiley and Sons, 2010.

Hunt, V. et al (2007). “RFID: A guide to radio frequency identification”. New Jersey: John Wiley and Sons.

Igoe, T. (2011). “Getting Started with RFID”. Sebastopol: O’Reilly.

- McRoberts (2013), M. (2013). "Beginning Arduino: Create simple but practical projects with arduino using step-by-step instructions and easy-to-follow diagrams, no experience necessary". 2 ed. Apress.
- Sweeney II, P. (2005). "RFID For Dummies". Indianapolis: Wiley Publishing.