



Fundação Educacional do Município de Assis
Instituto Municipal de Ensino Superior de Assis - IMESA

SAURO MARTINS REBESCHINI

**SISTEMA DE SEGURANÇA POR CÂMERAS E SENSORES
CONTROLADOS POR DISPOSITIVO REMOTO**

Assis

2012



Fundação Educacional do Município de Assis
Instituto Municipal de Ensino Superior de Assis - IMESA

SAURO MARTINS REBESCHINI

SISTEMA DE SEGURANÇA POR CÂMERAS E SENSORES CONTROLADOS POR DISPOSITIVO REMOTO

Trabalho de Conclusão de Curso
apresentado ao Instituto Municipal de
Ensino Superior de Assis como
requisito do curso de Análise e
Desenvolvimento de Sistemas.

Orientador: Prof^o.Esp. Guilherme de Cleva Farto

Área de Concentração: Informática

Assis
2012

FICHA CATALOGRÁFICA

REBESCHINI , Sauro M.

SISTEMA DE SEGURANÇA POR CÂMERAS E SENSORES CONTROLADOS
POR DISPOSITIVO REMOTO / Sauro Martins

Rebeschini. Fundação Educacional do Município de Assis – FEMA –
Assis, 2012.

58p.

Orientador: Prof^o.Esp. Guilherme de Cleve Farto .
Trabalho de Conclusão de Curso -
Instituto Municipal de Ensino Superior
de Assis – IMESA.

1. Câmeras. 2. Sensores. 3. Arduino.

CDD:001.42

Biblioteca FEMA



Fundação Educacional do Município de Assis
Instituto Municipal de Ensino Superior de Assis - IMESA

SISTEMA DE SEGURANÇA POR CÂMERAS E SENSORES CONTROLADOS POR DISPOSITIVO REMOTO

SAURO MARTINS REBESCHINI

Trabalho de Conclusão de Curso apresentado ao
Instituto Municipal de Ensino Superior de Assis do
Curso de Análise e Desenvolvimento de Sistema,
analisado pela seguinte comissão examinadora:

Orientador: Prof^o. Esp. Guilherme de Cleve Farto

Analisador: Prof^o. Ms. Osmar Aparecido Machado

Assis
2012

DEDICATÓRIA

Dedico esse trabalho primeiramente a Deus e aos meus pais Luiza Cristina Martins Rebeschini e Luiz Antonio Rebeschini, juntamente com minha vó Maria Nazarethi da Silva Martins, meu vô por parte de pai já falecido João Rebeschini e a minha vó Adelaide Perez Rebeschini. Sou grato por todos acreditarem em mim e por estarem sempre orando pelo meu progresso.

AGRADECIMENTOS

Obrigado Deus Oshieoyá Samá(Patriarca da igreja PL) e aos mestres da igreja PL de Marília, mestre Obuch e mestre Mirtes obrigado pelas orações e pelas orientações.

A minha orientadora, Prof^a. Regina Fumie Etó, pela orientação e ajuda segura durante este trabalho. Muito obrigado por todo apoio, paciência, disponibilidade, esforço e conhecimento prestados à minha carreira profissional.

Agradeço ao meu orientador Guilherme de Cleve Farto, que pegou o trabalho pela metade, mas já fez um grande progresso, me ajudou a concluir a minha monografia com êxito, com a sua sabedoria e paciência.

Aos meus pais, meu irmão e a minha vó, por sempre estarem orando por mim e pelos meus estudos.

A meu colega de sala e futuro sócio Cesar Henrique Guadagnino pela ajuda e por ser o meu melhor amigo e por me aguentar todos esses anos.

A minha namorada Letícia Arruda, que está sempre ao meu lado, me dando forças para continuar o projeto.

A Fundação Educacional do Município de Assis –(FEMA), por tornar-me capacitado para desenvolver este trabalho.

Aos professores do Curso de Análise e Desenvolvimento de Sistemas da FEMA, pelos valiosos ensinamentos durante a minha passagem pela instituição.

Aos amigos de curso e família, pelo apoio, amizade e demonstração de companheirismo.

A todos que direta ou indiretamente, contribuíram para a realização deste trabalho.

RESUMO

O objetivo deste trabalho é desenvolver um software juntamente com um protótipo de projeto com dispositivos digitais, câmera e sensores, baseando-se em conhecimento de eletrônica para realizar a montagem dos equipamentos. O desenvolvimento da programação do software será realizado por meio das tecnologias C# (C Sharp) e Web Asp .Net utilizando-se a plataforma Visual Studio 2012, assim como integrar a plataforma Arduino, responsável pelo controle dos dispositivos eletrônicos. A implementação desse projeto visa contribuir com um sistema Web responsável pelo controle de câmeras e sensores com o intuito de melhorar a segurança de residências e quaisquer outros locais onde possa ser instalado.

A aplicação final pode ser gerenciada via Web, tanto por computador quanto por um dispositivo móvel, desde que a autenticação do usuário e senha sejam validados. Apesar de o sistema apresentar uma usabilidade simplificada, é possível realizar o monitoramento, por exemplo, de uma residência ou estabelecimento comercial sem estar necessariamente presente no local.

Por ser um trabalho acadêmico, espera-se que trabalhos futuros possam contribuir e melhorar, ainda mais, o que fora desenvolvido durante este projeto, para que novas funcionalidades, dispositivos e sensores possam ser integrados, tornando o sistema de segurança mais completo.

Palavra chave: ARDUINO, SENSORES, CÂMERAS.

ABSTRACT

The aim of this work is to present the Assembly of a prototype project with digital devices, camera and sensors, based on knowledge of electronics to perform Assembly of equipment.

The development of software programming will be carried out by means of C # (C Sharp) and Asp .net using the Visual Studio platform 2012, as well as integrate the Arduino platform, responsible for the control of electronic devices.

The implementation of this project aims to contribute with a Web system responsible for the control of cameras and sensors to improve the safety of homes and any other locations where it can be installed.

The final application can be managed via the Web, both by computer and a mobile device, provided that the user and password authentication to be validated. Although the system prompts a simplified usability, it is possible to perform the monitoring, for example, of a residence or commercial establishment without being necessarily present on site.

To be an academic work, it is expected that future work could contribute and improve even more, which had developed during this project, so that new features can be integrated sensors and devices, making it the most comprehensive security system.

Key Word: ARDUINO , SENSOR, CAMERA.

LISTA DE ILUSTRAÇÕES.

Figura 1- Sensor de Presença	12
Figura 2- Câmera.....	13
Figura 3- Imagens Modelos de Arduino	18
Figura 4- Diagramas de Blocos	21
Figura 5- Arquitetura Básica de um Arduino (SILVA, 2010).	22
Figura 6- Dispositivos Importantes do Arduino (COSTA, 2010).	25
Figura 7- Pinagem Atmega168.	26
Figura 8- Diagrama de Blocos	26
Figura 9- Ambiente Gráfico do Arduino (ENDEL B, 2008).....	28
Figura 10- Arquitetura de Programação no Arduino.	31
Figura 11- Modelo lógico do Banco de Dados	40
Figura 12- Arquitetura do Sistema de Cadastro.	41
Figura 13- Diagrama de Caso de Uso.	42
Figura 14- Diagrama de Classe.	42
Figura 15- Diagrama de Sequência.	43
Figura 16- Diagrama de integração do computador com o sistema.....	52
Tela 1- Sistema de Cadastro de Usuário	44
Tela 2- Página Principal.....	45
Tela 3- Login e Senha.	46
Tela 4- Entrada do Sistema.....	47
Tela 5- Câmeras e Gravações.....	48
Código1- Programação no Arduino Sensores.....	49
Código2- Comunicação do C# via porta serial com Arduino.....	50

LISTA DE TABELAS

Tabela 1 - Aplicações do reconhecimento de padrões (JAIN et al, 2000).....	18
Tabela 2- Principais Diferenças Entre os Arduinos (SILVEIRA, 2001).....	21
Tabela 3 - Comandos do Toolbar e Suas Funções (adaptado de ENDEL C, 2008).....	32
Tabela 4 - Comandos do Menu Sketch e suas funções (adaptado de ENDEL C, 2008).	32
Tabela 5 - Comandos do Menu Tools e Suas Funções (adaptado de ENDEL C, 2008)	33

Sumário

1. INTRODUÇÃO.....	12
1.1 – OBJETIVOS.....	15
1.2 – JUSTIFICATIVAS.....	15
1.3 - MOTIVAÇÃO.....	15
1.4 - ESTRUTURA DO TRABALHO.....	15
2. FUNDAMENTAÇÃO TEÓRICA BÁSICA.....	16
2.1- RECONHECIMENTO DE PADRÕES.....	16
2.2- CÂMERAS.....	18
2.3- SENSORES.....	18
2.3.1- Hardware.....	21
2.3.2- Características do Arduino	22
2.4- Ambiente de Programação para o Arduino	27
2.4.1- IDE Arduino.....	27
2.4.2- Linguagem de programação para Arduino.....	32
2.4.3- Linguagem de referência.....	32
2.4.4- Funções.....	33
2.4.5- Bibliotecas.....	35
2.4.6- Comunicação (redes e protocolos).....	36
2.4.7- Sensoriamento.....	37
2.4.8 - Geração de Frequência e de Áudio.....	37
2.4.9- Temporização.....	37
2.4.9.1- Utilidades.....	38
2.4.9.2- Comunicação Entre o Arduino e o C#.....	38
3. DESENVOLVIMENTO DO PROJETO.....	38
3.2- MODELAGEM DO PROBLEMA.....	39
3.3- IMPLEMENTAÇÃO.....	40
3.3.1- Criação do banco de dados.....	40
3.3.2- Sistema de Cadastramento.....	42
3.3.2.1- Especificação.....	43
3.3.3- Telas do Sistema e Códigos.....	46
3.3.3.4- Material Utilizado.....	52
4. CONCLUSÃO.....	53

1. INTRODUÇÃO

O século XXI chegou com várias modificações e tendências sociais e tecnológicas para trazer conforto e comodidade à população. Pode-se apontar como uma tendência a domótica ou seja a automatização residencial.

Um dos percalços dos atuantes da área de domótica é quanto às instalações interativas de ambiente, que englobam a configuração de plantas residenciais levando-se em conta as instalações elétricas e eletrônicas envolvidas no processo de automação residencial. Como exemplo, possível citar a instalação de câmeras e sensores no ambiente para dar todo o rastreamento de movimentos.

Será implementado um software Web para que usuários sejam autenticados e autorizados à utilizar o sistema de segurança em qualquer local ou momento.

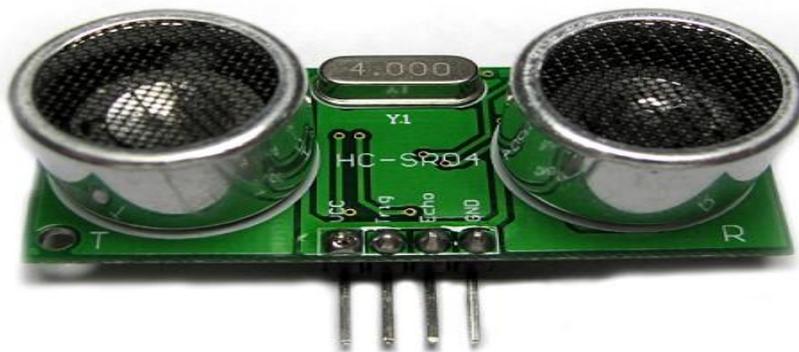


Figura 1 – Sensor de Presença

O projeto final desenvolvido durante este trabalho acadêmico pode ser implantado em qualquer local que necessite de acesso restrito ou apenas para monitoração.

A partir do uso de tal produto é possível as movimentações em cada perímetro em que for configurado, mas isso após uma movimentação no local designado.

O projeto foi idealizado devido a suspeitas de desaparecimento de equipamentos e entrada de pessoas não autorizadas no local de implantação do sistema de monitoramento remoto.

O intuito deste trabalho é o de desenvolver um sistema de segurança e monitoramento de ambiente, tornando-o de fácil acesso a uma maior comunidade em geral e de proporcionar um gerenciamento do laboratório de eletrônica para fins didáticos.

Na figura 2 pode-se observar a câmera que irá fazer as gravações do Sistema de Segurança.



Figura 2 – Lic 100

O sistema será composto por um conjunto de sensores, que serão ativados por dispositivo remoto cujas funções básicas são:

- Ativar câmeras de segurança remotamente por meio da aplicação Web;

- Gravar a movimentação do local ou simplesmente sem realizar o registro;

Existem diversas maneiras de se monitorar um local ou recinto. Porém, o meio tecnológico mais utilizado atualmente é fazendo-se uso de câmeras e sensores. Por este motivo, o sistema deste trabalho fora desenvolvido com a plataforma Arduino, realizando-se a comunicação entre câmeras e sensores Web desenvolvida com a tecnologia Microsoft.Net. Para fins de teste o projeto foi implantado em uma maquete didática, disponibilizada pela FEMA/IMESA junto ao laboratório de eletrônica. As funcionalidades deste projeto serão apresentadas e detalhadas nas próximas seções.

Exemplos:

- Cadastro realizado somente pelo Administrador do sistema Web;
- Monitoramento por Website com autenticação de usuário e senha, onde somente o usuário autenticado terá acesso às câmeras de monitoramento;

Atualmente, os sistemas de segurança por câmeras e sensores são uma excelente maneira de monitorar o ambiente de uma residência ou qualquer outro local. No futuro, provavelmente irão se tornar uma parte da vida para a maioria das pessoas, além da automação realizada em outros dispositivos como chaves, cartões de banco e senhas.

Neste projeto foi desenvolvido um dispositivo de captura de imagem por câmeras e sensores de movimentação utilizando-se Arduino, que é baseado em um micro controlador Atmega e dessa forma é logicamente programável, ou seja, torna-se possível de programa-las, utilizando uma linguagem própria baseada em C, C++, C# ou tecnologia Java, quando já implementada faz com que o hardware execute certas ações. O Arduino faz parte do conceito de *hardware* e *software* livre e está aberto para uso e contribuição de toda sociedade. (FONSECA e BEPPU, 2010)

1.1 - OBJETIVOS

O objetivo é o de criar um ambiente domótico, sendo que o estudo do ambiente físico a ser trabalhado e da especificação das posições de dispositivos que irão compor o ambiente automatizado terá o enfoque primário. Como segundo objetivo, será focado no micro controlador da ATmega e da família Arduino, que realizará o gerenciamento dos dispositivos acoplados ao ambiente de monitoramento e se possível utilizar um sistema remoto para controle do ambiente.

1.2 - JUSTIFICATIVAS

Os sensores com câmeras que podem ser integradas ao dispositivo Arduino surgiram a partir do século XX e há pouco trabalho técnico e acadêmico provendo informações sobre tais projetos de automação residencial ou industrial. Este sistema é um desafio, pois para se desenvolver é necessário conhecer eletrônica, hardware e software. Todo este projeto tem o acoplado hardware, ou seja, Arduino que usa uma plataforma Open Source. Com outros componentes como câmeras e sensores de presença.

1.3 - MOTIVAÇÕES

A motivação para o desenvolvimento deste projeto é que o Arduino se tornou uma realidade em vários projetos de automação por ter uma fácil linguagem de desenvolvimento e um forte poder de processamento em pequenos e grandes projetos. Com isso, despertou a curiosidade em conhecer a sua programação e o funcionamento no tratamento de imagem e integração com sensores e câmeras, juntamente com a plataforma Visual Studio 2010 e linguagem de programação C # Web.

1.4 - ESTRUTURA DO TRABALHO

A estrutura do trabalho é apresentada em quatro capítulos. Sendo o primeiro a introdução. No segundo capítulo, serão apresentadas as fundamentações teóricas Básicas. No terceiro capítulo, serão apresentados os desenvolvimentos gerais. No quarto capítulo, é apresentada a conclusão.

2. FUNDAMENTAÇÃO TEÓRICA BÁSICA

Neste capítulo será realizada a fundamentação teórica acerca das tecnologias utilizadas para o desenvolvimento deste projeto acadêmico. Para isso, serão abordados os temas como reconhecimento de padrões, câmeras, sensores, entre outros tópicos necessários para a fase de construção do projeto final.

2.1 – RECONHECIMENTO DE PADRÕES

O Reconhecimento de Padrões (RP) é a ciência que tem por objetivo a classificação de objetos em categorias ou classes. Desde os primórdios da computação, a tarefa de implementar algoritmos, tem se apresentado como uma das mais intrigante e desafiadora.

As técnicas de reconhecimento de padrões apresentam um vasto leque de aplicações nas áreas científicas e tecnológicas, principalmente na área de informática. O interesse na área de reconhecimento de padrões tem aumentado nos últimos anos, devido a novas aplicações que não são só um desafio, mas também computacionalmente mais exigentes.

Estas aplicações incluem *data mining*, ou mineração de dados que identificam:

- Um padrão ou uma relação entre milhões de modelos; a classificação de documentos, muito útil para procurar documentos de texto;
- Previsões financeiras; organização e recuperação de bancos de dados multimídia e biometria, que é a identificação pessoal baseada em vários atributos físicos ou comportamentais;

A tabela 1 mostra algumas aplicações do reconhecimento de padrões.
(JAIN et al, 2000)

Domínio do Problema	Aplicação	Padrão de Entrada	Classes de Padrão
Bioinformática	Análise de Sequência	DNA/Sequência de proteínas	Tipos conhecidos de genes/padrões
Mineração de dados	Busca por padrões significantes	Pontos em um espaço multidimensional	Compactar e bem separar grupos
Classificação de documentos	Busca na Internet	Documento texto	Categorias semânticas (negócios, esportes e etc.)
Análise de documentos de imagem	Máquinas de leitura para cego	Documento de imagem	Palavras e caracteres alfanuméricos
Automação industrial	Inspeção de circuito impresso em placas	Intensidade ou alcance de imagem	Produto defeituoso/não defeituoso
Recuperação de base de dados multimídia	Busca na Internet	Vídeo clipe	Gêneros de vídeos
Reconhecimento biométrico	Identificação pessoal	Face, íris, impressão digital	Usuários autorizados para controle de acesso
Sensoriamento remoto	Prognóstico da produção de colheita	Imagem multiespectral	Categorias de aproveitamento de terra, desenvolvimento de padrões de colheita
Reconhecimento de voz	Inquérito por telefone sem assistência de operador	Voz em forma de onda	Palavras faladas

Tabela 1 - Aplicações do reconhecimento de padrões (JAIN et al, 2000).

A busca para projetar e construir sistemas capazes de realizar o reconhecimento automático de padrões de forma precisa e confiável é imensamente útil. Durante a resolução dos inúmeros problemas necessários para construir esse sistema, a compreensão de como é realizado o reconhecimento de padrões no mundo real se faz necessária, pois algumas aplicações como o reconhecimento da fala e o reconhecimento facial, influenciam na maneira como esses sistemas são estruturados.

Existe ainda uma nova aplicação para o reconhecimento de padrão, chamado computação afetiva, que dá a um computador a capacidade de reconhecer e expressar emoções e empregar mecanismos que contribuem para a tomada de decisão racional (JAIN et al, 2000).

A técnica de comparação é realizada pelo método do qual se deseja obter o reconhecimento, que é depois conferido com o padrão armazenado. A comparação entre os modelos exige bastante processamento, mas a disponibilidade de processadores mais rápidos a preços acessíveis, tem tornado esta abordagem mais viável. Os processos de tratamento das imagens são bem complexos e exige um amplo conhecimento das técnicas e métodos de processamento digital de imagens. Esta complexidade envolve a escolha adequada das técnicas e métodos para obter uma solução razoável para o problema em questão. (SERRANO, 2010).

2.2 CÂMERAS

As câmeras possuem uma grande importância no sistema de segurança e para o desenvolvimento deste trabalho. Será utilizada uma do modelo WebCam Lic 100 da marca LG, podendo estar ser conectada ao computador e realizar a comunicação por meio da porta serial.

As câmeras possuem um zoom limitado, porém serão colocadas em pontos estratégicos para obterem as melhores imagens da movimentação.

Apesar de a câmera ser bem simples e por o objetivo do trabalho ser o desenvolvimento de um protótipo, servirá muito bem para efeito de teste. Porém para a realização projetos futuros poderá ser substituída por uma câmera mais potente e de qualidade superior.

2.3 SENSORES

O sensor, como ilustrado na **Figura 1**, é um dispositivo de fácil integração à plataforma Arduino.

É um sensor de presença responsável pelo monitoramento de pessoas em um ambiente que apesar de ter um alcance limitado, sendo bem instalado em um perímetro poderá ter um bom foco da pessoa ou objeto em movimento.

2.3.1 – ARDUINO

O Arduino é uma placa controladora composta e alguns outros componentes eletrônicos montados numa pequena placa de circuito impresso com uma interface serial para comunicação com um computador padrão PC. Por meio dessa placa é possível utilizar existem também alguns conectores onde podem ser ligados outros circuitos externos, como sensores, *LEDs*, chaves, relés e pequenos motores. O Arduino também conta com uma ferramenta para criação de protótipos de eletrônica baseada no conceito de *software* e *hardware* livres. Isso quer dizer que esses projetos podem ser copiados e modificados por outras pessoas conforme suas necessidades e depois podem ser colocados de volta ao domínio público de modo que outros usuários possam usufruir dessas mudanças em seus próprios projetos. A figura 3 mostra alguns modelos de Arduino. (SILVEIRA 2011)

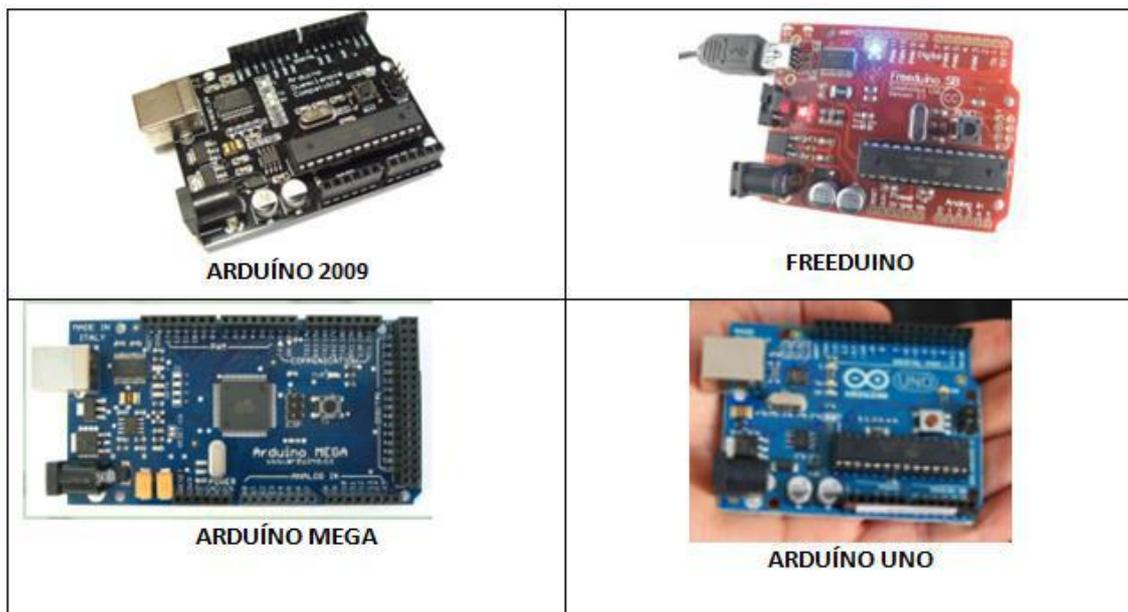


Figura 2 - Modelos de Arduino

O Arduino é composto por duas partes principais: um *hardware*, a placa de circuito impresso com o micro controlador, e um *software*, o *bootloader*, um aplicativo residente na memória de programas desse micro controlador. Externamente existe também uma interface gráfica, um programa que roda em ambiente Windows ou Linux num computador PC ou numa máquina Apple com o Mac OS X. É nessa interface gráfica ou ambiente de desenvolvimento integrado (*Integrated Development Environment* ou *IDE*) onde implementa-se os programas que serão carregados no Arduino. São esses programas, chamados de *sketches*, que vão dizer ao *hardware* o que deve ser feito (SILVEIRA 2011).

Na verdade, o Arduino é um kit de desenvolvimento capaz de interpretar variáveis no ambiente e transformá-las em sinal elétrico correspondente, por meio de sensores ligados aos seus terminais de entrada, e atuar no controle ou acionamento de algum outro elemento eletroeletrônico conectado ao terminal de saída. Ou seja, é uma ferramenta de controle de entrada e saída de dados, que pode ser acionada por um sensor (por exemplo, um resistor dependente da luz - LDR) e que, logo após passar por uma etapa de processamento, o micro controlador, poderá acionar um atuador (um motor, por exemplo). Como podem perceber, é como um computador, que têm como sensores de entrada como o mouse e o teclado, e de saída, impressoras e caixas de som, por exemplo, só que ele faz interface com circuitos elétricos, podendo receber ou enviar informações/tensões neles.

2.3.2 – Hardware

O *hardware* do Arduino é baseado nos microcontroladores AVR da Atmel, em particular nos modelos ATmega8, ATmega168, ATmega328 e no ATmega1280. O Arduino recebe um codinome em italiano dependendo do microcontrolador utilizado. A tabela 2 mostra as principais diferenças entre os Arduinos em relação ao microcontrolador que ele recebe.

ARDUINO	Diecimila	Duemilanove 168	Duemilanove 328	Mega
Processador	ATmega8	ATmega168	ATmega328	ATmega1280
Memória <i>flash</i>	8 k	16 K	32 K	128 K
Memória RAM	1 K	1 K	2 K	8 K
Memória EEPROM	512 bytes	512 bytes	1 K	4 K
Pinos digitais	14	14	14	54
Pinos analógicos	6	6	6	16
Saídas PWM	3	6	6	14

Tabela 2-Principais diferenças entre os Arduinos

O projeto original do Arduino foi baseado no circuito básico com um microcontrolador ATmega8, cujo diagrama em blocos é mostrado na figura 4.

Entretanto, uma fonte de uma fonte de alimentação simples com o regulador LM7805, um circuito de conversão para comunicação serial RS-232 e alguns conectores para controle externo e entrada de sensores também foram incorporados

a este mesmo projeto. Além disso, foi incorporado posteriormente ao projeto, um conector para a programação do microcontrolador no circuito, o ICSP ou *In-Circuit Serial Programming*.



Figura 4 – Diagrama de bloco

2.3.3 – Características do Arduino

Para a utilização do Arduino é necessário um bom conhecimento das características básicas de *hardware* para que sejam explorados ao máximo todos os recursos disponíveis. A figura 5 mostra a arquitetura básica de um Arduino.

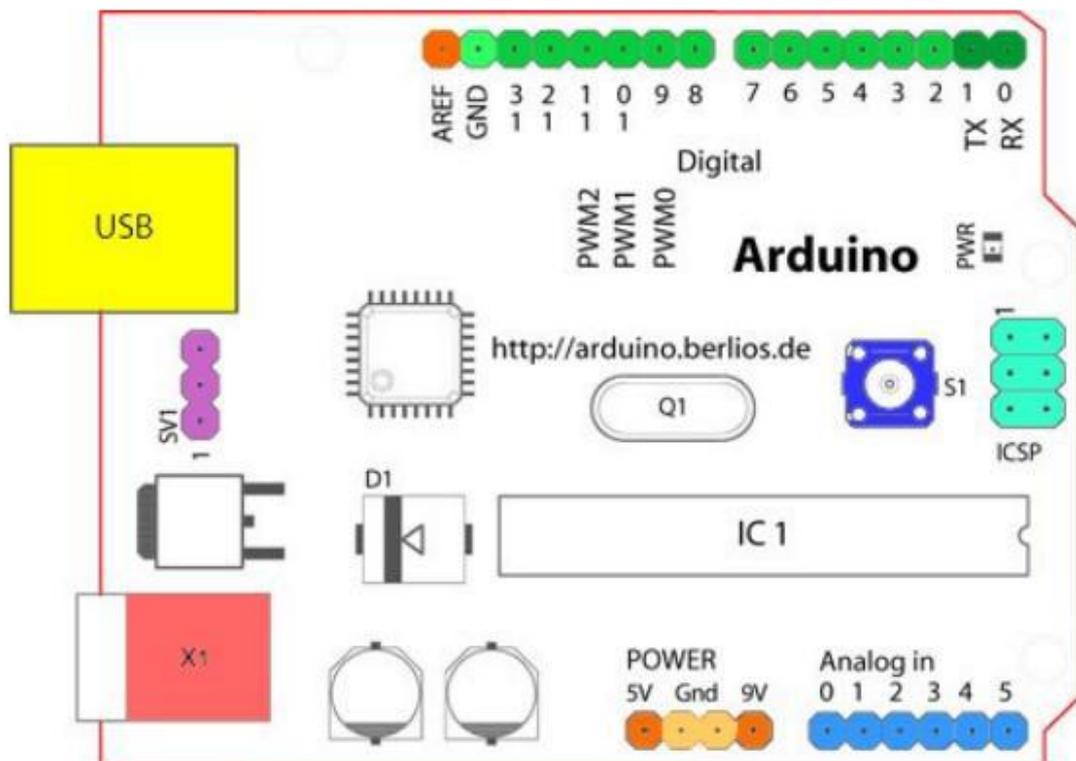


Figura 5 – Arquitetura básica de um Arduino

O *hardware* do Arduino é uma placa baseada no microcontrolador ATmega. Tem 14 pinos de entrada ou saída digital (dos quais 6 podem ser utilizados como saídas

PWM), 6 entradas analógicas, um oscilador de cristal 16MHz, controlador USB, uma tomada de alimentação, um conector ICSP, e um botão de reset. Para sua utilização basta conectá-lo a um computador com um cabo USB ou ligá-lo com um adaptador AC para DC ou bateria.

O Arduino pode ser alimentado pela conexão USB ou por qualquer fonte de alimentação externa. A fonte de alimentação é selecionada automaticamente. De acordo com (FONSECA e BEPPU, 2010), os pinos de alimentação são:

- 5V: a fonte de alimentação utilizada para o microcontrolador e para outros componentes da placa. Pode ser proveniente do pino 9V através de um regulador *on-board* ou ser fornecida pelo USB ou outra fonte de 5V;
- X1: suprimento externo de energia In (9-12VDC);
- 9V: entrada de alimentação para a placa Arduino quando uma fonte externa for utilizada. Pode fornecer alimentação por este pino ou, se usar o conector de alimentação, acessar a alimentação por este pino;
- GND (*ground*): pino terra;
- SV1: Jumper que determina a alimentação da USB ou externa X1.

Cada um dos 14 pinos digitais do Arduino pode ser usado como entrada ou saída usando as funções de *pinMode()*, *digitalWrite()* e *digitalRead()*. Eles operam com 5 V. Cada pino pode fornecer ou receber um máximo de 40 mA e tem um resistor *pull-up* interno (desconectado por padrão) de 20-50k .Além disso, de acordo com (FONSECA e BEPPU, 2010), alguns pinos têm funções especializadas:

- Digital Pins 0-1/Serial In TX/RX (Verde Escuro) – estes pinos não podem ser usados para digital i/o (digital Read e digital Write) se estiver usando comunicação serial (Serial.begin)
- PWM: 3,5,6,9,10,e11. Fornecem uma saída analógica PWM de 8-bit com a função *analogWrite()*;
- SPI: 10(SS), 11(MOSI), 12(MISO), 13(SCK). Estes pinos suportam comunicação SPI, que embora compatível com o hardware, não está incluída na linguagem do Arduino;
- LED: 13. Há um LED já montado e conectado ao pino digital13;

□ AREF: Analog Reference pin-AREF, referência de tensão para entradas analógicas. Usados com *analogReference()*.

O arduino tem um fusível que protege a porta USB do seu computador contra curto circuito e sobrecarga de corrente. Apesar da maioria dos computadores possuírem proteção interna própria o fusível proporciona uma proteção extra. Se mais de 500mA forem aplicados na porta USB, o fusível irá automaticamente interromper a conexão até que o curto ou a sobrecarga seja removida.

□ USB: Usada para gravar os programas; Comunicação serial entre placa e computador; Alimentação da placa (Amarelo).

Algumas versões anteriores do Arduino requerem um reset físico (pressionando o botão de reset na placa) antes de carregar um sketch (o programa a ser compilado). Os Arduino baseados em microcontrolador ATmega168 ou ATmega328 foram projetados de modo a permitir que isto seja feito através do software que esteja rodando no computador conectado. Uma das linhas de controle de hardware (DTR) do FT232RL está conectada ao reset do ATmega328 via um capacitor de 100microFaraday. Quando esta linha é colocada em nível lógico baixo, o sinal cai por tempo suficiente para reiniciar o chip. O software Arduino usa esta característica para permitir carregar o programa simplesmente pressionando o botão *upload no ambiente Arduino*. Isto significa que o *bootloader* pode ter um *timeout* mais curto, já que a ativação do DTR (sinal baixo) pode ser bem coordenada com o início do *upload*.

□ S1: Botão de Reset

O arduino tem 6 entradas analógicas, cada uma delas está ligada a um conversor analógico-digital de 10bits, ou seja, transformam a leitura analógica em um valor dentre 1024 possibilidades. Por padrão, elas medem de 0 a 5 Volts, embora seja possível mudar o limite superior usando o pino AREF e um pouco de código de baixo nível.

□ Analog In Pins 0-5

Além disso, tem o ICSP que nada mais é do que um conector.

□ ICSP: In-circuit Serial Program

A figura 6 mostra outros dispositivos importantes no Arduino.

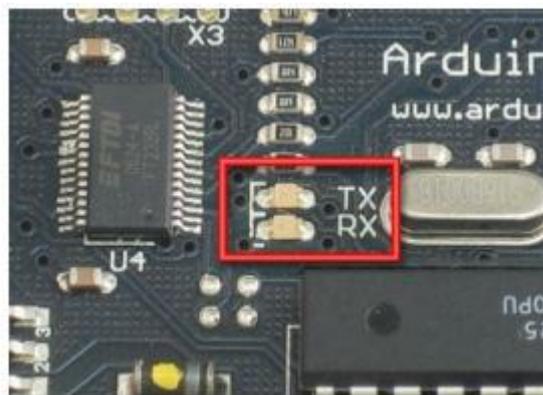


Figura 6 – Dispositivos importantes do Arduino

A comunicação com um computador, com outro Arduino ou com outros microcontroladores é muito simplificada. O Arduino permite comunicação serial no padrão UARTTTL (5V), que está disponível nos pinos digitais 0(RX)e1(TX). Um chip FTDI232RL na placa encaminha esta comunicação serial através da USB e os drivers FTDI (incluído no software do Arduino) fornecem uma porta virtual para o software no computador. O software Arduino inclui um monitor serial que permite que os dados de texto sejam enviados e recebidos à placa Arduino. Os LEDs RX e TX da placa piscam quando os dados estão sendo transferidos ao computador pelo chip FTDI e há conexão USB (mas não quando há comunicação serial pelos pinos 0 e 1).

- Led Serial: 0 (RX) e 1 (TX). Usados para receber (RX) e transmitir (TX) dados seriais TTL. Estes pinos são conectados aos pinos correspondentes do chip serial FTDI USB-to-TTL;
- Chip FTDI (comunicação serial USB)

A figura 7 mostra a disposição dos pinos do Atmega168

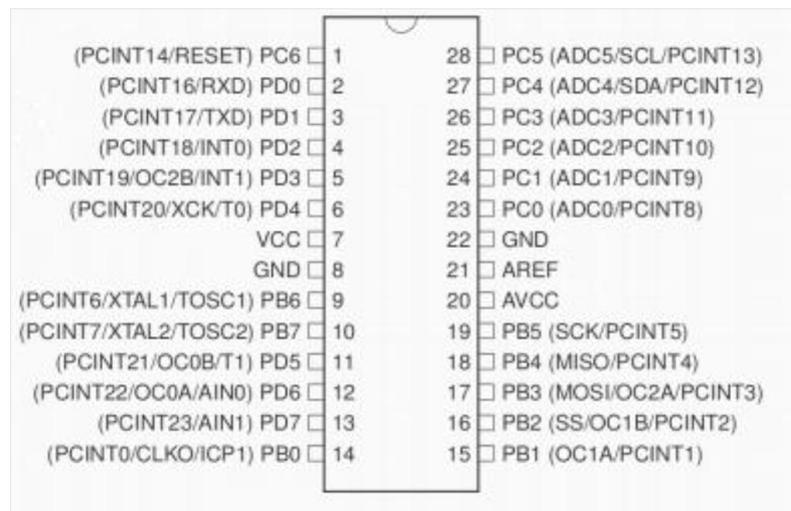


Figura 7 - Pinagem Atmega168.

Saber sobre os pinos é importante para não queimar o circuito e serve para saber fazer as ligações corretamente.

A Figura 8 mostra o diagrama de blocos interno do Atmega 168:

considerada a linguagem do Arduino. A linguagem *Process* é similar às linguagens C e C++, contudo possui uma série de restrições. O software Arduino tem um ambiente gráfico muito semelhante ao do *Processing*.

Um programa escrito em *process* é denominado sketch e deve ser escrito no Ambiente de desenvolvimento Arduino. O ambiente Arduino é escrito na linguagem Java e assim está disponível para diversos sistemas.

Porém o sistema que desenvolvido utiliza com linguagem de programação C++ baseando-se na comunicação com o C# Web.

Quando um sketch é compilado, o ambiente Arduino *percorre* o código fonte fazendo algumas substituições e transforma o código *Process* em linguagem C/C++ que é compilado efetivamente pelo compilador da GNU.

A figura 9 mostra o ambiente gráfico do Arduino.

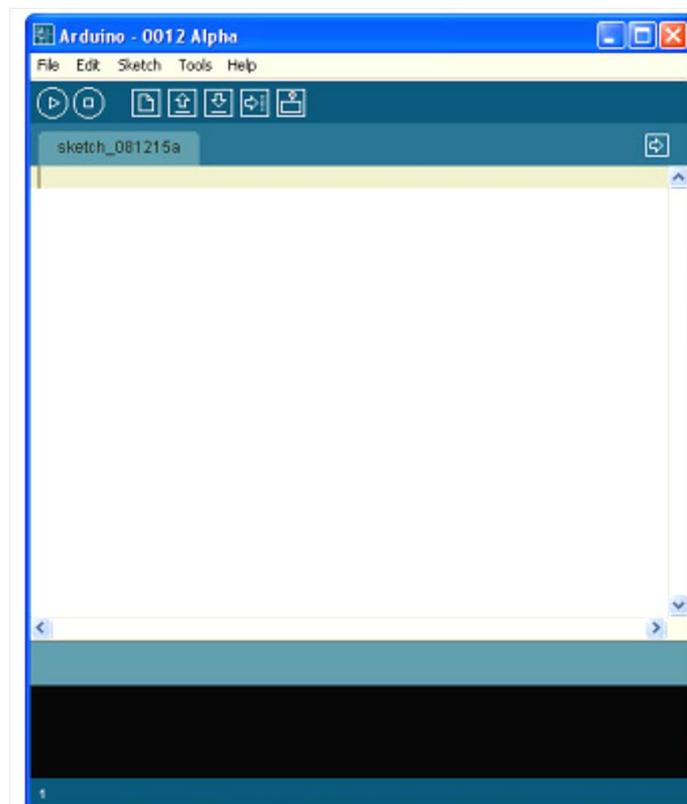


Figura 9 - Ambiente Gráfico do Arduino (ENDEL B, 2008).

O ambiente é constituído pelo Toolbar, Tab Menu e Menus (*File*, *Edit*, *Sketch*, *Tools* e *Help*). A tabela 3 mostra os vários botões com funções distintas.

	Verifica se o código tem erros
	Para o serial monitor ou desativa outros botões
	Cria um novo documento
	Abre uma lista de documentos salvos e abre o que for selecionado
	Salva o documento
	Envia código para o Arduino
	Mostra a informação enviada pela placa do arduino
	Cria nova Tab

Tabela 3 - Comandos do Toolbar e Suas Funções (adaptado de ENDEL C, 2008).

O Tab Menu permite gerir documentos com mais do que um ficheiro, cada um aberto num tab independente. Esses ficheiros podem ser ficheiros normais de código Arduino (sem extensão), ficheiros C (extensão .c), C++ (.cpp) ou *header files* (.h). Os Menus *File*, *Edit* e *Help* são semelhantes em todos os programas, e não será feita uma descrição detalhada de cada uma delas. A tabela 4 mostra o menu *Sketch*.

Verify/Compile	Verifica se o código tem erros
Import Library	Adiciona bibliotecas ao código, o que lhe confere novas possibilidades
Show Sketch Folder	Abre a pasta do ficheiro no Ambiente de Trabalho
Add File...	Adiciona outro ficheiro ao ficheiro inicial. Esse ficheiro aparece numa nova aba

Tabela 4 - Comandos do Menu Sketch e suas funções (adaptado de ENDEL C, 2008).

O menu *Tools* é composto por outras funções que são intrínsecas a este programa. A tabela 5 mostra os comandos do menu tools e suas funções.

Auto Format	Formata o código para uma melhor leitura, alinhando as chaves e indentando o seu conteúdo.
Copy for Discourse	Copia o código para o clipboard, de forma que seja possível colocá-lo num fórum e manter a distinção da sintaxe por cores.
Board	Seleciona a placa que estamos a usar, o que controla a forma como o código é compilado e o modo como é feito o upload, assim como o comportamento dos itens do menu Burn Bootloader
Serial Port	Contém todas as portas série que o computador possui, permitindo escolher a que está a ser usada. Para usar portas USB é necessário instalar um driver ³ . No Mac, a porta série USB tem uma apresentação semelhante a esta: <i>/dev/tty.usbserial-1B1</i> e no Windows a esta: <i>COM1</i> , variando os números finais.
Burn Bootloader	Os itens deste menu permitem gravar um bootloader na placa <i>Arduino</i> com variados programas, mas não é necessário para um uso normal do <i>Arduino</i> .

Tabela 5 - Comandos do Menu Tools e Suas Funções (adaptado de ENDEL C, 2008)

A figura 10 mostra a arquitetura de programação no Arduino.

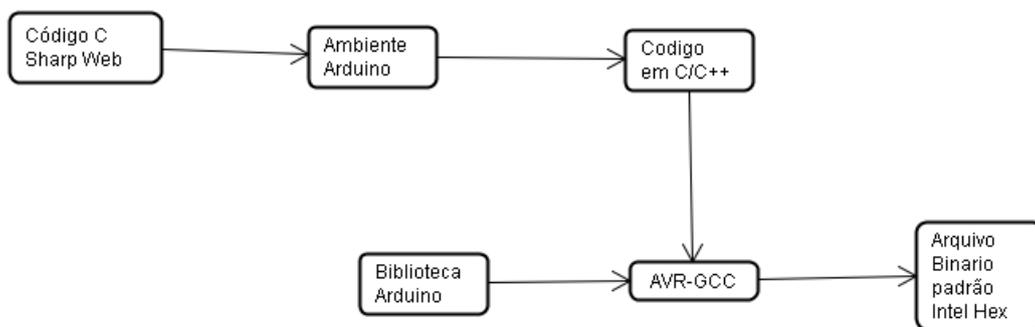


Figura 10 - Arquitetura de Programação no Arduino.

No ambiente Arduino são realizadas algumas transformações no código C Sharp Web, o resultado é um código C/C++. No Processo GCC é compilado o código C/C++ e junta às bibliotecas para controle dos recursos do microcontrolador, tais como `Serial.printf()` e `digitalWrite()`. No fim de todo esse processo é gerado um arquivo binário que será gravado na memória do microcontrolador.

2.4.2 – Linguagem de programação para Arduino

Nesta seção será apresentada toda a parte de linguagem de programação, objetivando citar comando e especificações.

2.4.2 - Linguagem de referência

As funções são referências essenciais para o desenvolvimento de um projeto sendo o Arduino, principalmente para os iniciantes no assunto. Essas funções já implementadas e disponíveis em bibliotecas direcionam e exemplificam as funcionalidades básicas do microcontrolador. Segundo (FONSECA e BEPPU, 2010) as funções básicas e de referências são:

- Estruturas de controle (if, else, break, ...);
- Sintaxe básica (define, include, ; , ...);
- Operadores aritméticos e de comparação (+, -, =, ==, !=, ...);
- Operadores booleanos (, ||, !);
- Acesso a ponteiros (*,);
- Operadores compostos (++ , - , += , ...);
- Operadores de bits (|, ^ ; ...).

- Valores de referências**

- Tipos de dados (byte, array, int , char , ...);
- Conversões(char(), byte(), int(), ...);
- Variável de escopo e de qualificação (variable scope, static, volatile, ...);

- Utilitários (sizeof(), diz o tamanho da variável em bytes)

É bom citar que o software que vem no Arduino já provê várias funções e constantes para facilitar a programação que são:

- setup();
- loop();
- Constantes (HIGH | LOW , INPUT | OUTPUT , ...);
- Bibliotecas (Serial, Servo, Tone, etc.).

2.4.3- Funções

As funções são referências essenciais para o desenvolvimento de um projeto usando o Arduino, principalmente para os iniciantes no assunto. Essas funções já implementadas e disponíveis em bibliotecas direcionam e exemplificam as funcionalidades básicas do microcontrolador. Segundo (FONSECA e BEPPU, 2010) as funções básicas e de referências são:

- Digital I/O - pinMode() digitalWrite() digitalRead();
- Analógico I/O - analogReference() analogRead() analogWrite() – PWM;
- Avançado I/O - tone() noTone() shiftOut() pulseIn();
- Tempo - millis() micros() delay() delayMicroseconds();
- Matemática - min() max() abs() constrain() map() pow() ***só do C/C++ sqrt() ***só do C/C++;
- Trigonométrica - sin() ***só do C/C++ cos() ***só do C/C++ tan() ***só do C/C++;
- Números aleatórios - randomSeed() random();

- Bits e Bytes - lowByte() highByte() bitRead() bitWrite() bitSet() bitClear() bit();

- Interrupções externas -attachInterrupt() detachInterrupt();
- Interrupções - interrupts() noInterrupts();
- Comunicação Serial.

2.4.4 – Bibliotecas

O uso de bibliotecas nos proporciona um horizonte de programação mais amplo e diverso quando comparado à utilização apenas de estruturas, valores e funções. Isso é perceptível quando analisamos os assuntos que são abordados por cada biblioteca em específico. Lembrando sempre que, para utilizar uma biblioteca esta já deve estar instalada e disponível na sua máquina. As bibliotecas de referencias são:

- EEPROM - leitura e escrita de "armazenamento" permanente;
- Ethernet - para se conectar a uma rede Ethernet usando o Arduino Ethernet Shield;
- Firmata - para se comunicar com os aplicativos no computador usando o protocolo Firmata;
- LiquidCrystal - para controlar telas de cristal líquido (LCDs);
- Servo - para controlar servo motores;
- SPI - para se comunicar com dispositivos que utilizam barramento Serial Peripheral Interface (SPI);
- SoftwareSerial - Para a comunicação serial em qualquer um dos pinos digitais;
- Stepper - para controlar motores de passo;
- Wire - Dois Wire Interface (TWI/I2C) para enviar e receber dados através de uma rede de dispositivos ou sensores. (FONSECA e BEPPU, 2010)

Além dessas referencias, existem algumas bibliotecas mais específicas. Isso é de extrema importância, pois é possível utilizar o arduíno com um enfoque em uma determinada área. De acordo com (FONSECA e BEPPU, 2010) essas bibliotecas são.

2.4.5 - Comunicação (redes e protocolos)

Aqui tem alguns comandos para serem usados na interface do Arduino:

- Messenger - Para o processamento de mensagens de texto a partir do computador;
- NewSoftSerial - Uma versão melhorada da biblioteca SoftwareSerial;
- OneWire - Dispositivos de controle que usam o protocolo One Wire;
- PS2Keyboard - Ler caracteres de um PS2 teclado;
- Simple Message System - Enviar mensagens entre Arduino e o computador;
- SSerial2Mobile - Enviar mensagens de texto ou e-mails usando um telefone celular;
- Webduino - Biblioteca que cria um servidor Web (para uso com o Arduino Ethernet Shield);
- X10 - Envio de sinais X10 nas linhas de energia AC;
- XBee - Para se comunicar via protocolo XBee;
- SerialControl - Controle remoto através de uma conexão serial.

2.4.6 - Sensoriamento

- Capacitive Sensing - Transformar dois ou mais pinos em sensores capacitivos;
- Debounce - Leitura de ruídos na entrada digital.

2.4.7 - Geração de Frequência e de Áudio

- Tone - Gerar ondas quadradas de frequência de áudio em qualquer pino do microcontrolador.

2.4.8 - Temporização

- DateTime - Uma biblioteca para se manter informado da data e hora atuais do software;
- Metro - Ajuda ao programador a acionar o tempo em intervalos regulares;
- MsTimer2 - Utiliza o temporizador de 2 de interrupção para desencadear uma ação a cada N milissegundos.

2.4.9 - Utilidades

- TextString (String) - Manipular strings
- PString - uma classe leve para imprimir em buffers.
- ▣ Streaming - Um método para simplificar as declarações de impressão.

2.5 – Comunicação Entre o Arduino e o C#

O. Net é a nova plataforma de desenvolvimento da Microsoft que tem como foco principal o desenvolvimento de Serviços Web baseados em XML.

Para a comunicação das câmeras, utilizasse as bibliotecas Aforge.Vídeo e Aforge.Show, sendo possível integra-las facilmente a um projeto em C#.

E com a biblioteca Webduino pode-se utilizar uma placa Ethernet Shield para a comunicação com o código Web por meio um programa em Arduino.

A aplicação Web contém uma tela de autenticação de usuário e senha, integrando o projeto Arduino com a plataforma de desenvolvimento C#, possibilitando aos usuários de visualizar a câmera em tempo real.

3. DESENVOLVIMENTO DO PROJETO

O software foi desenvolvido com a plataforma Visual Studio 2010 para o Web site e a câmera foi usado a plataforma do Arduino mesmo.

No desenvolvimento do Web site foi usado algumas bibliotecas foram adicionadas no Visual Studio como a Aforge.Show, Aforge.Video.directshow elas fazem a parte da captura da imagem.

Ja na parte do Arduino foi desenvolvido o software para o sensor de presença, não foi acrescentado nenhuma biblioteca, apenas foi implementado usando as portas digitais pin6, pin7, Vcc(5 Volts para alimentar o sensor) e o pino GND(negativo) para o sensor Ultrasonic(sensor sonorizador).

A junção dos dois sistemas formam o Sistema de Segurança Controlado por Dispositivo Remoto via Web, a comunicação entre os dois software serem feitas serialmente.

O Arduino possui uma porta serial que vareia de Arduino para Arduino,com a porta COM2, COM3, COM6,COM7 e assim por diante. Por ela passa todas as informações adquiridas durante a transação dos dados reseptores e emissores.

3.1 – DESCRIÇÃO DO PROBLEMA

O limite ou restrição do acesso de pessoas a um ambiente muitas vezes pode ser fraco, tornando necessário novos meios de gerenciar a detecção de movimentos em determinados perímetros. Tudo é de acordo com o local onde o sistema for implantado.

3.2 – MODELAGEM DO PROBLEMA

Nesta seção, será apresentada a modelagem do sistema geral do projeto. O modelo tem as câmeras juntamente com os sensores de presença, o Arduino será programado para receber as imagens via Web em tempo real, juntamente com o sinal de cada sensor do local e se caso forem ativados, será gravado um vídeo pelo programa.

O computador servirá possuirá acesso à internet por meio de uma aplicação desenvolvida em C# com a plataforma Visual Studio 2010 fazendo a comunicação com o Arduino.

O desenvolvimento do projeto foi dividido em módulos para facilitar a sua implementação.

Módulo 1: Sistema de cadastramento Web;

Módulo 2: Desenvolvimento do dispositivo de câmeras e sensores;

Módulo 3: Integração do dispositivo com o Arduino;

3.3 – IMPLEMENTAÇÃO

Será apresentado a implementação de cada módulo do projeto para a ativação das câmeras e sensores utilizando o Arduino.

3.3.1 – Criação do banco de dados

Com a finalidade de armazenar as informações do projeto, tornou-se necessário modelar e utilizar um banco de dados relacional a ser utilizado pela aplicação Web desenvolvida em C#.

A Figura 11 demonstra o modelo lógico do banco de dados utilizado para o gerenciamento das informações:

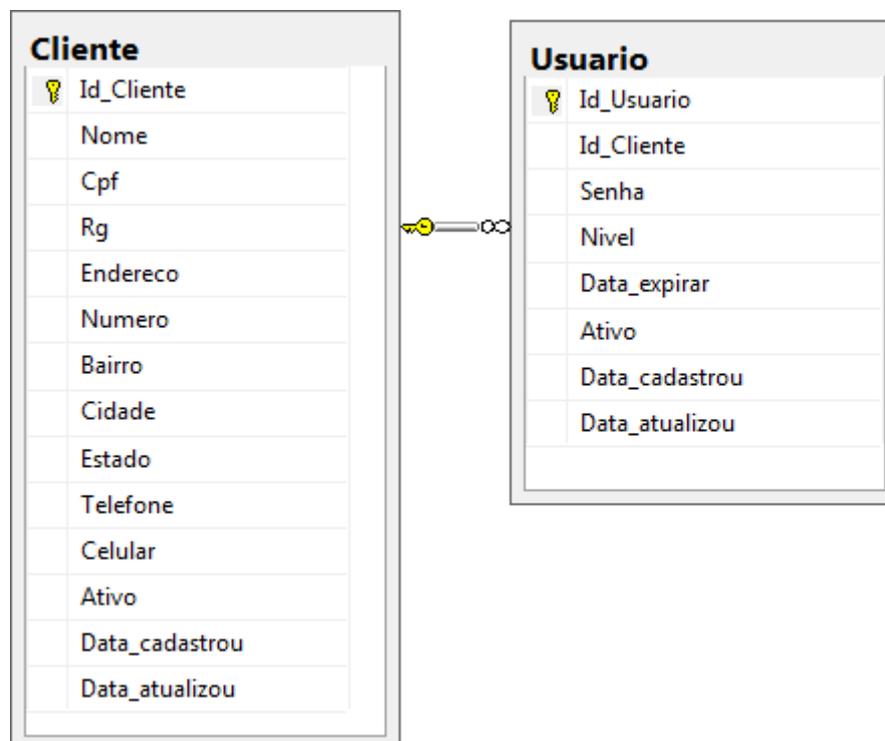


Figura 11 - Modelo Lógico do Banco de Dados.

3.3.2 – Sistema de Cadastramento

A tarefa de cadastrar novos usuários é realizada pelo administrador do sistema, possibilitando que novas pessoas possam utilizar a aplicação Web.

A figura 12 mostra a arquitetura do sistema de cadastramento.

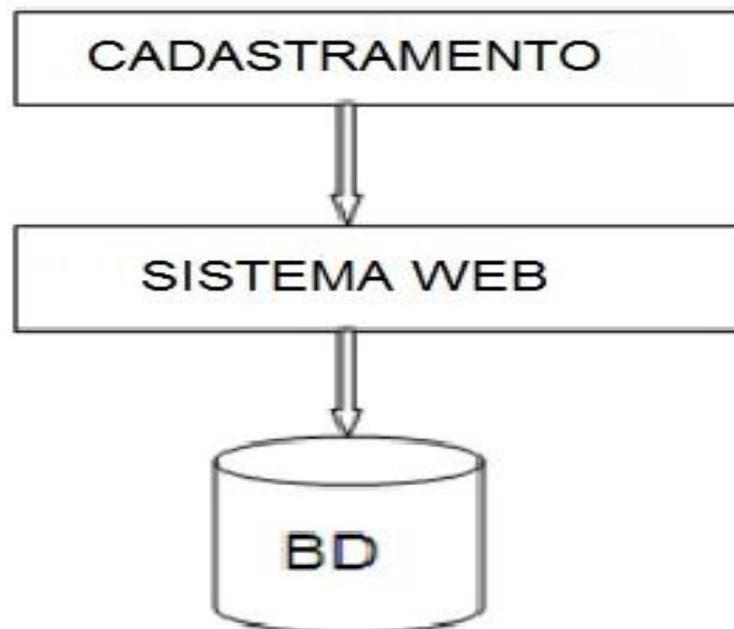


Figura 12 - Arquitetura do Sistema de Cadastro.

3.3.2.1 – Especificação

Para a especificação do sistema de cadastro foi desenvolvido apenas um o diagrama de caso de uso. A figura 13 mostra o diagrama de caso de uso para realizar o cadastro de um novo usuário.

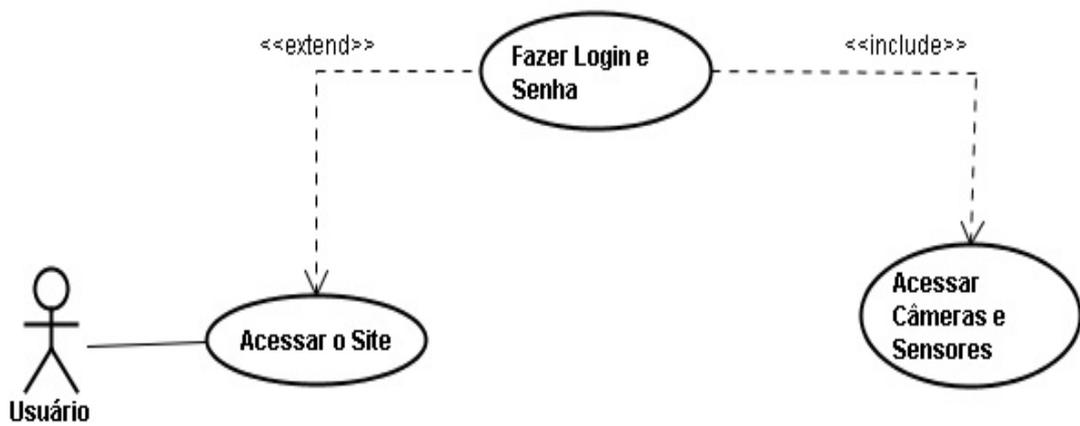


Figura 13 - Diagrama de Caso de Uso.

A figura 14 mostra o diagrama de classes do sistema, que tem apenas para cadastrar o código, nome, cidade, estado e CPF para confirmação do acesso ao sistema Web.

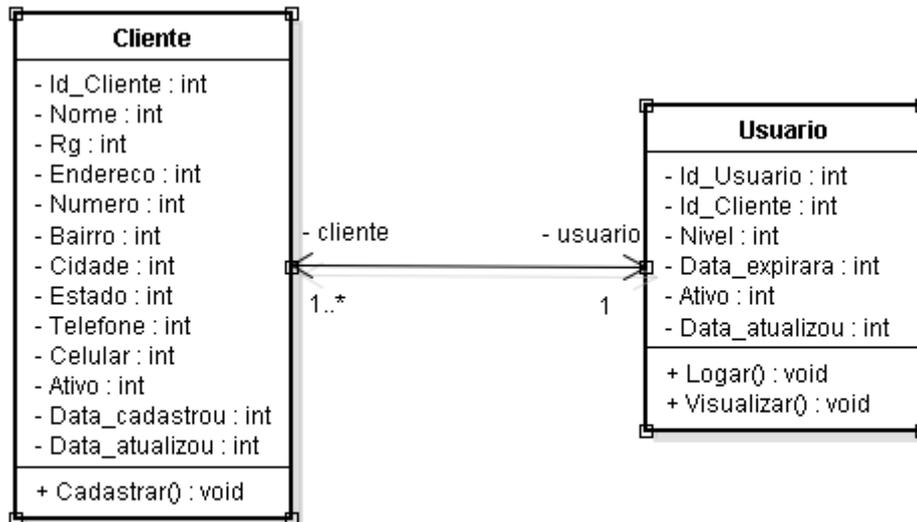


Figura 14 - Diagrama de Classe.

A figura 15 mostra o diagrama de sequência do sistema de cadastro construído a partir do diagrama de classe.

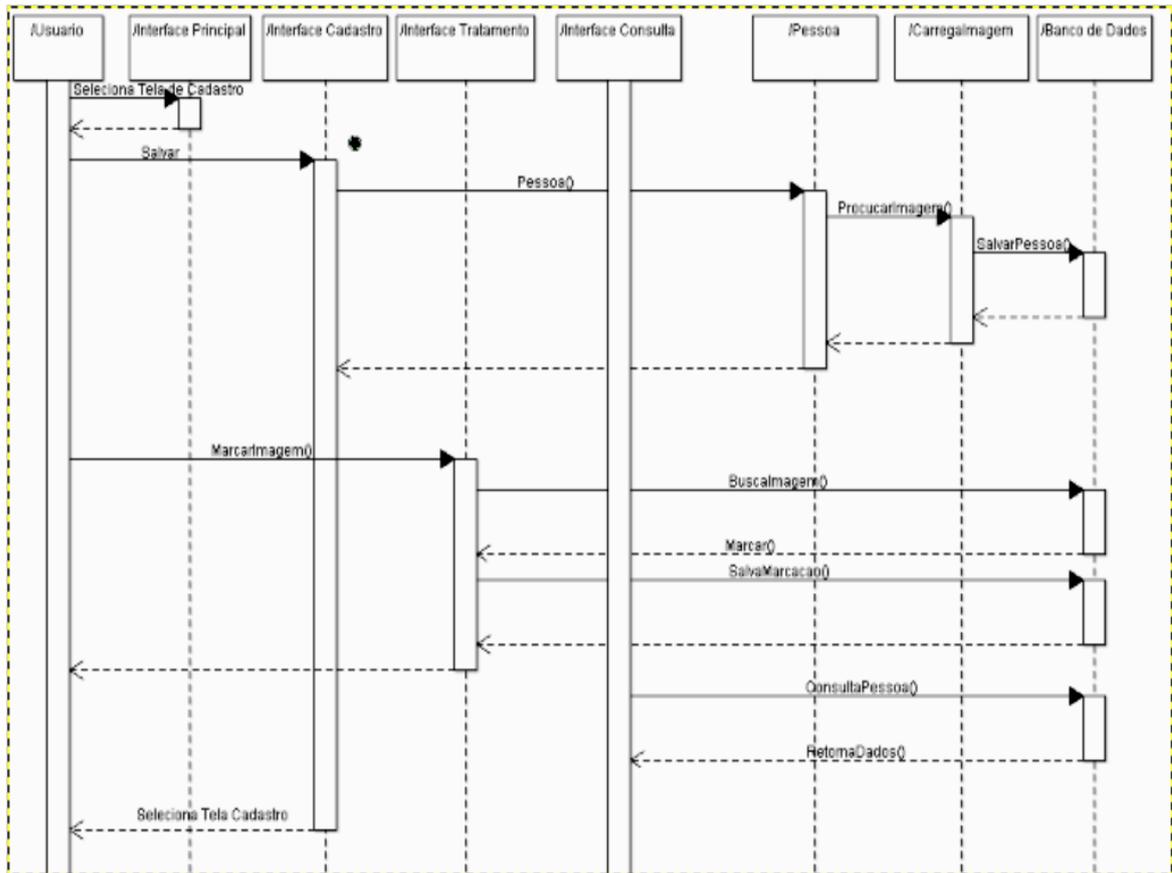


Figura 15 - Diagrama de Sequência.

3.3.3.3- Telas do Sistema e Códigos

localhost:1095/1CC%20F0KMA1AD01/ADM/cadastro_de_usuario.aspx

SSV.WEB

Cadastro de Usuário Complete

Cadastro de Usuário

Nome: *

Senha: *

Confirmação da Senha:

E-mail:

Pergunta Secreta:

Resposta Secreta:

Tela 1- Sistema de Cadastro de Usuário

- Sistema de Cadastro de Usuário

O sistema de cadastro é por meio do ASP.NET Configuration, sendo assim o banco do cadastro será manipulado pelo administrador. Seu cadastro terá mais segurança, por que só uma pessoa terá acesso dificultando alguma tentativa de invadir o sistema. Para realizar o cadastro de um novo usuário a pessoa deverá entrar em contato com o Administrador para ter seu login e senha.

localhost:1095/TCC%20FORMATADO1/Default2.aspx

SSV.WEB

Seja bem vindo(a) ao site do SSV.WEB-SISTEMA DE SEGURANÇA VIA WEB.

[Login](#)
Clique aqui para efetuar login caso seja cadastrado.

Tecnologias usadas pelo sistema:

Arduino AtMEGA;
Sensore de presença Ultrasonic;
WebCam.

Imagens dos dispositivos:

Arduino



Sensor Ultrasonic



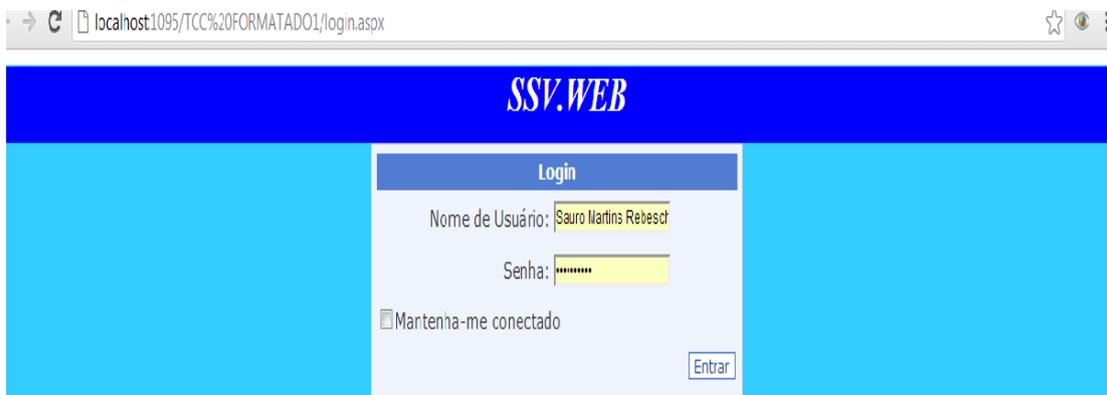
WebCam



Tela 2- Página Principal

- **Página Principal**

Nesta tela o usuário do sistema poderá efetuar autenticação login para acessar as gravações feitas pelas movimentações contidas no local, onde as câmeras estão instaladas.



Tela 3- Login e Senha

- Login e Senha

Para acessar as gravações ou visualizar a câmera em tempo real, o usuário deverá se autenticar informando sua conta de *login* e senha.

localhost:1095/TCC%20FORMATADO1/Default.aspx

SISTEMA DE MONITORAMENTO SSV.WEB

Gravações

Olá, Sauro Martins Rebeschini

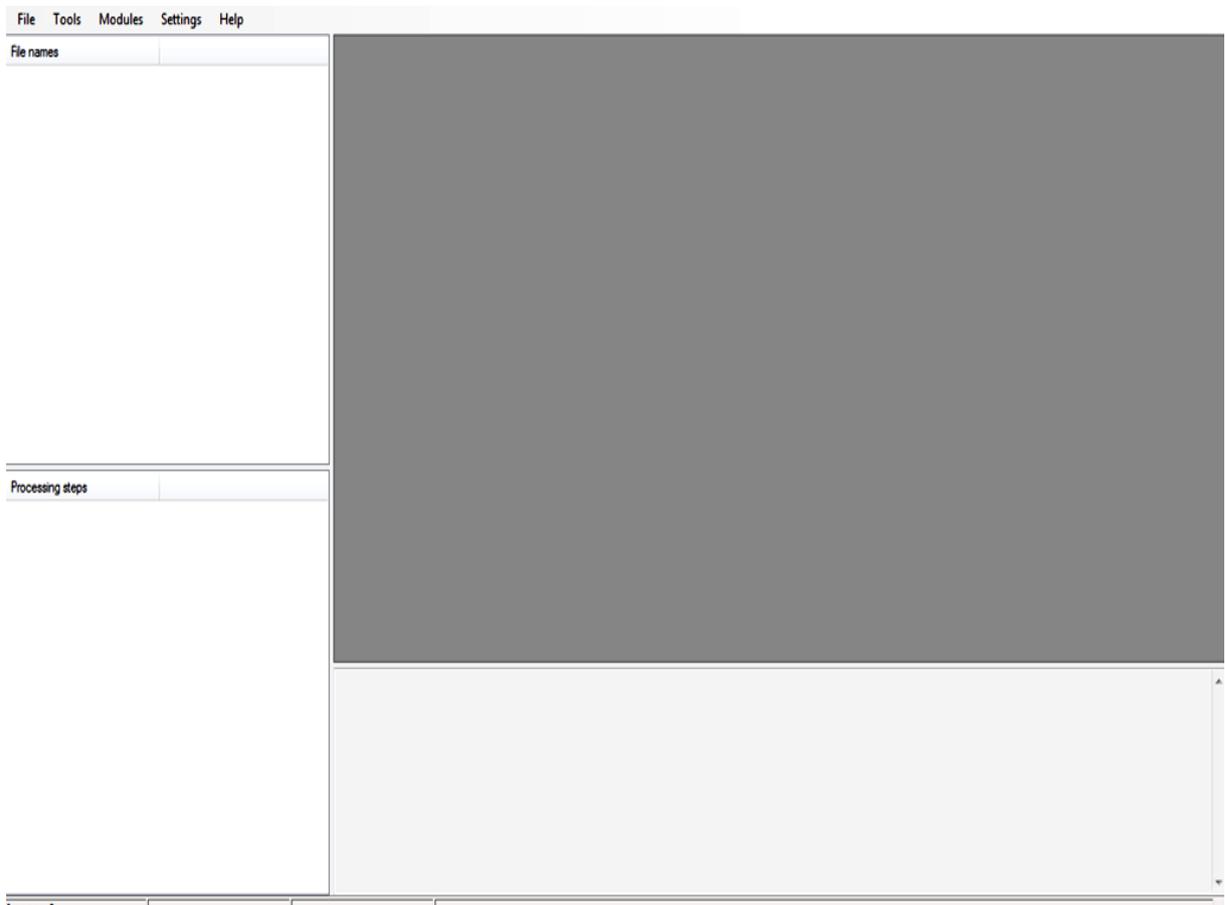
Seja bem vindo(a) ao Sistema de Monitoramento via Web.
A sua segurança esta aqui.



Tela 4- Entrada do Sistema

- Entrada do Sistema

Após a autenticação, o usuário validado terá acesso às gravações realizadas pelas câmeras de segurança, conforme ilustrado na Figura XXX:



Tela 5- Câmeras e Gravações

- Câmeras e Gravações

Juntamente com o arquivo de gravação será exibido a data e hora de filmagem possibilitando uma filtragem do momento em que se deseja visualizar a imagem.

Aqui mostra a tela de interface do Arduino e o código fazendo a comunicação serial.

```

void loop(){
  long cm,cml;
  cm = distancia(trig,echo);
  cml = distancia(trigl,echol);
  Serial.print( cm);
  Serial.print( " ");
  Serial.println( cml);
  if (cm <= 30) { //se cm menor ou igual a 30, liga o LED
    digitalWrite(ledPin, HIGH);
  }
  else {
    digitalWrite(ledPin, LOW);
  }
  delay(500);
}

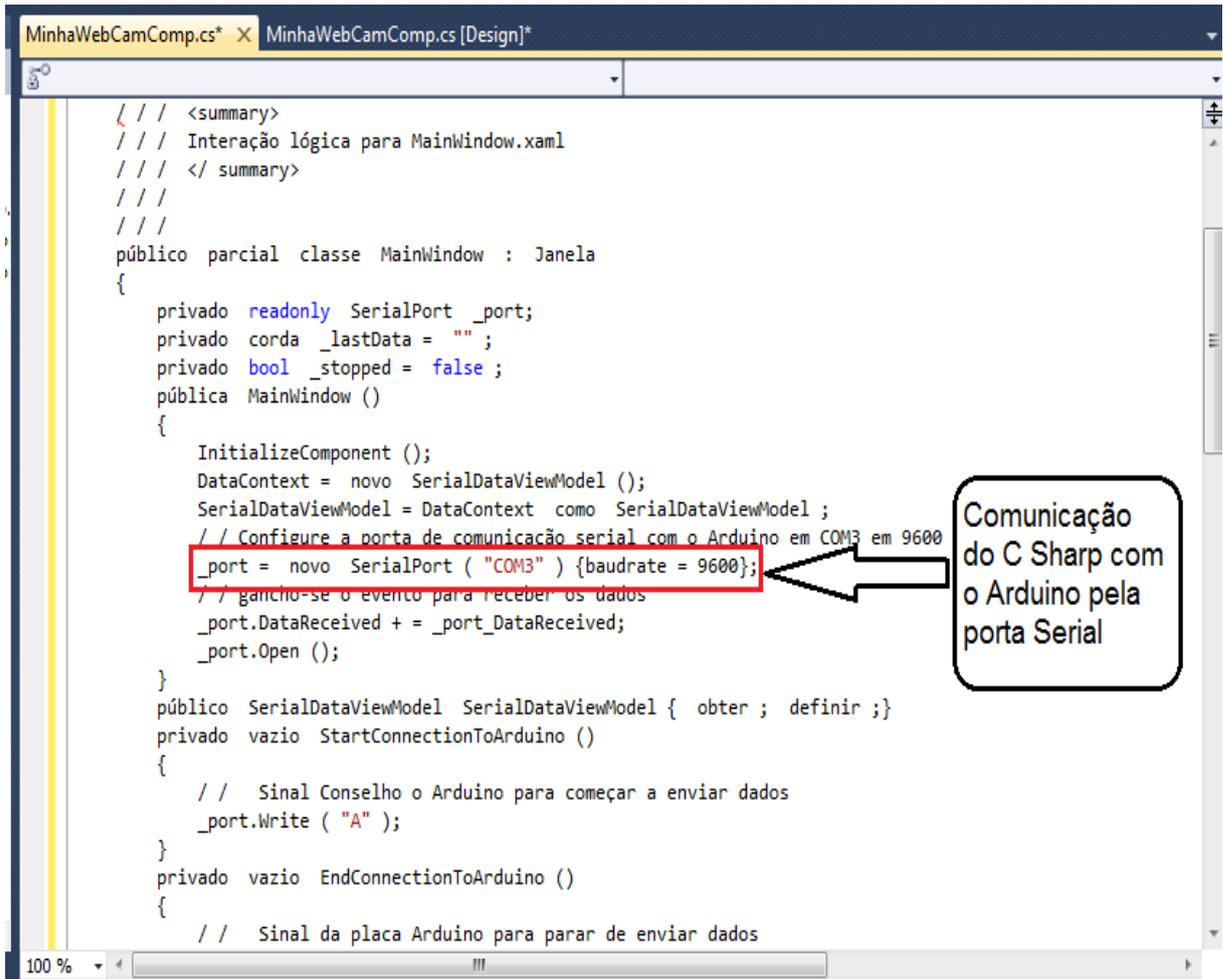
long distancia(int trig,int echo){
  long duration, cm;
  //comandos para enviar um pulso do sonar e armazenar o tempo de resposta
  digitalWrite(trig, LOW);
  delayMicroseconds(2);
  digitalWrite(trig, HIGH); //ativa o sonar por 10 microsegundos
  delayMicroseconds(10);
  digitalWrite(trig, LOW);
  delayMicroseconds(2);
  duration = pulseIn(echo, HIGH); //duration recebe o tempo de ida e volta do sinal
  cm = duration/29/2; //converte duration em centímetros
  return cm;
}

```

Função do sensor que capta a movimentação no local.

Código 1- Programação no Arduino Sensores

- Programação no Arduino Sensores
Toda a movimentação esta sendo captada por meio de uma dessa função em C que, apesar de simples, é bem objetiva para a funcionalidade desejada.



```
MinhaWebCamComp.cs* x MinhaWebCamComp.cs [Design]*  
  
{ // <summary>  
  // Interação lógica para MainWindow.xaml  
  // </summary>  
  //  
  //  
  público parcial classe MainWindow : Janela  
  {  
    privado readonly SerialPort _port;  
    privado corda _lastData = "" ;  
    privado bool _stopped = false ;  
    pública MainWindow ()  
    {  
      InitializeComponent ();  
      DataContext = novo SerialDataViewModel ();  
      SerialDataViewModel = DataContext como SerialDataViewModel ;  
      // Configure a porta de comunicação serial com o Arduino em COM3 em 9600  
      _port = novo SerialPort ( "COM3" ) {baudrate = 9600};  
      // gancho-se o evento para receber os dados  
      _port.DataReceived += _port_DataReceived;  
      _port.Open ();  
    }  
    público SerialDataViewModel SerialDataViewModel { obter ; definir ;}  
    privado vazio StartConnectionToArduino ()  
    {  
      // Sinal Conselho o Arduino para começar a enviar dados  
      _port.Write ( "A" );  
    }  
    privado vazio EndConnectionToArduino ()  
    {  
      // Sinal da placa Arduino para parar de enviar dados
```

Comunicação
do C Sharp com
o Arduino pela
porta Serial

Código 2-Comunicação do C# via porta serial com Arduino

- Comunicação do C# via porta serial com Arduino

Para realizar a comunicação entre Arduino e a aplicação C# tornou-se necessário criar uma classe responsável pela troca de informações por meio da porta serial, possibilitando que sinais analógicos e digitais pudessem ser enviados e recebidos, acessando a câmera e sensor de presença.

3.3.3.4– Material Utilizado

Será feita uma descrição dos materiais utilizados para o desenvolvimento do sistema de segurança controlado por dispositivo remoto.

□ Arduino ATmega Arduino 328.

Para a construção de um sistema é preciso uma placa de Arduino que nesse equipamento será manuseado os códigos onde captará o sinal de presença.

□ Câmeras

As especificações da câmera:

- WebCam Lic 100
- aparelhos eletrônicos

A comunicação do Arduino com o dispositivo é feita por um programa desenvolvido em C/C++, tendo em vista que a sintaxe da linguagem é baseada na biblioteca *Webduino* (baseado em C# Web). A figura 16 mostra o diagrama de integração do computador com o sistema de segurança por dispositivo remoto.

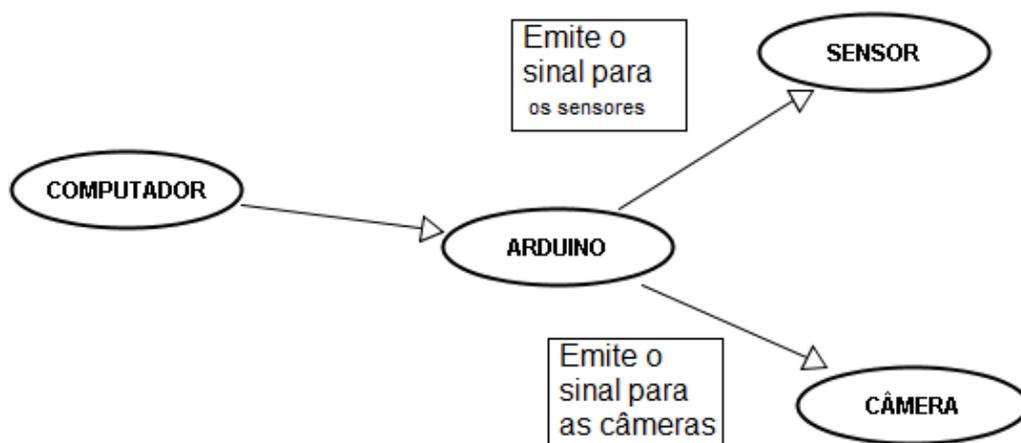


Figura 16 – Diagrama o Circuito.

4. CONCLUSÃO

Ao concluir este projeto, pode se dizer que tudo começa pela motivação em realizar um projeto que poderá beneficiar as pessoas dando a elas mais segurança. Alguns motivos foram de grande ajuda, como a falta de monitoramento em residências ou departamentos.

Outras formas de se concluir tal sistema é o conhecimento de outras plataformas de programação o Arduino tem um simples entendimento, mas com grande capacidade de implementação, sendo uma plataforma open source pode ser feito vários experimentos desde ligar uma lâmpada por sensor de presença como automatizar sua residência inteira.

Durante esse projeto teve muitas adversidades e dificuldades durante o desenvolvimento do software, por tanto muitas coisas foram descobertas e adquiridas através de vídeo aulas e alguns livros.

É interessante comentar sobre a comunicação do Arduino e C#, que por sua vez não foi fácil sua implementação, pois foi usado algumas bibliotecas diferentes como a AForge.Video e até assistir vídeo aula para poder entender mais sobre o sistema.

O mais importante é ter aprendido sobre várias tecnologias podendo contribuir um pouco com a segurança dos perímetros e por deixar em aberto a possibilidade de melhorar o projeto atribuindo mais coisas a ele.

REFERÊNCIAS

CAPUANO, G.F. e IDOETA, I.V., **Elementos de eletrônica digital**, Editora Erica, 2001.

COSTA, S.M.F., **Classificação e verificação de impressões digitais**, Dissertação de Mestrado, Escola Politécnica da Universidade de São Paulo, 2001.

De La ROCHA, F.R., **Curso Introdutório de Microcontroladores – Plataforma Arduino**, 2010.

ENDEL A, http://www.lojasmartsec.com.br/produto.php?cod_produto=51184.

ENDEL B, Arduino Language Reference, <http://arduino.cc/en/Reference/HomePage>, 2008. Acesso em maio de 2011.

ENDEL C, http://produto.mercadolivre.com.br/MLB-199915725-max7219-driver-de-display-7-segmentos-pic-8051-arduino-avr_JM. Acesso em setembro de 2011.

ENDEL D, <http://www.ptc.gda.pl/english/products-services/products,catalogue,product,biometric-fingerprint-sensor-usb-zk4000.html>. Acesso em março de 2011.

ENDEL E, <http://www.sparkfun.com/datasheets/Components/General/COM-09622-MAX7219-MAX7221.pdf>. Acesso em setembro de 2011.

ENDEL F, <http://multilogica-shop.com/driver-para-display-de-led-max7219cng>. Acesso em setembro de 2011.

ESPINOSA-DURÓ, Virginia. **Minutiae Detection Agoritmo for FingerPrint Recognition**. IEEE Aerospace and Eletronics System Magazine, Vol. 17, 2002. 59

FARIA, D.R., **Reconhecimento de Impressões Digitais com Baixo Custo Computacional para um Sistema de Controle de Acesso**, Dissertação de Mestrado, UFP, 2005.

FERNANDES, C.C. e LOPES, G.T., **Introdução ao Arduino**, [http://pt.scribd.com/doc/35379935/ Introducao-ao-Arduino-Get-Starter-com-arduino](http://pt.scribd.com/doc/35379935/Introducao-ao-Arduino-Get-Starter-com-arduino) acesso em maio de 2011.

FONSECA, E.G.P. e BEPPU, M.M., **Apostila Arduino**, Universidade Federal Fluminense, Rio de Janeiro, 2010.

FUTURLEC, <http://www.futurlec.com/LED/LEDM88RGCC.shtml>. Acesso em setembro de 2011.

HARRIS, T., **HowStuffWorks - Como Funcionam os Leitores de Impressões Digitais**. <http://informatica.hsw.uol.com.br/leitores-de-impressoes-digitais2.htm>. Acesso em maio de 2011.

JAIN, Anil; DUIN, Robert; MAO, Jianchang. **Statistical Pattern Recognition: A Review**. 2000. 34p. IEEE Transactions on Pattern Analysis and machine Intelligence, Vol. 22, 2000.

KERNIGHAN B.W. e RITCHIE, D.M., **The C programming language** , 2th Edition, 1988.

LEMAY, L. e PERKINS, C.L., **Teach yourself – Java in 21 days**, Sams.net Publishing, 1996.

MAZI, R.C. e DAL PINO JR, A., **Identificação Biométrica Através da Impressão Digital Usando Redes Neurais Artificiais**, Anais do XIV ENCITA, 2009.

POTTS, A. e FRIEDEL JR, D. **Java programming language handbook**, CH., Coriolis Group Books, 2004.

SCHILDT, H., **C Completo e Total**, 3ª Edição, Makron Books do Brasil Editora Ltda, 1996.

SILVA, L.M.G., Mini-Curso de Arduino, 2010. Disponível em www.brasilrobotics.blogspot.com. Acesso em maio de 2011.

SILVEIRA, J.A., **O Arduino**, <http://www.ordemnatural.com.br>, Acesso em maio 2011.

ZURADA, Jackek M., **Introduction to Artificial Neural System** West Publishing Company, 1992 110p.

TROELSEN, Andrew, **C# and .NET Platform**.

Site da câmera:

http://www.google.com.br/imgres?q=lic+100+lg&num=10&hl=pt-BR&biw=1360&bih=677&tbnm=isch&tbnid=T2WPGaWRal4OsM:&imgrefurl=http://produto.mercadolivre.com.br/MLB-440071690-webcam-lg-lic-100-com-prendedor-quebrado-_JM%3FredirectedFromParent%3DMLB432026888&docid=5EkM8q7PrDLrRM&imgurl=http://bimg2.mlstatic.com/webcam-lg-lic-100-com-prendedor-quebrado_MLB-F-2937080808_072012.jpg&w=900&h=1200&ei=e916UL-wBcuo0AHUsYDgAw&zoom=1&iact=hc&vpx=1125&vpy=122&dur=45&hovh=259&hovw=194&tx=160&ty=127&sig=117674056675204068049&page=1&tbnh=144&tbnw=105&start=0&ndsp=27&ved=1t:429,r:8,s:0,i:93

Site do sensor:

<http://www.webtronico.com/sensor-de-distancia-ultrasonico-hc-sr04.html>