CENTRO UNIVERSITÁRIO NOSSA SENHORA DO PATROCINIO

FACULDADE DE INFORMÁTICA

CURSO DE SISTEMAS DE INFORMAÇÃO

ANA PAULA VIEIRA VIANA

GILMAR DE CAMARGO PICCIN

**Sistema de alarme residencial utilizando microcontrolador e ferramenta de desenvolvimento de código aberto.**

ITU

NOVEMBRO/2012

ANA PAULA VIEIRA VIANA

GILMAR DE CAMARGO PICCIN

**Sistema de alarme residencial utilizando microcontrolador e ferramenta de desenvolvimento de código aberto.**

Trabalho de conclusão do curso de Sistemas de Informação do Centro Universitário Nossa Senhora do Patrocínio como requisito para a obtenção do título de Bacharel em Sistemas de Informação.

Orientador Prof. Ricardo Roberto Leme

ITU

NOVEMBRO/2012

“Agradecemos a Deus, pois sem ele não teríamos força para essa longa jornada.”

“No que diz respeito ao empenho, ao compromisso, ao esforço, à dedicação, não existe meio termo. Ou você faz uma coisa bem feita ou não faz.”

Ayrton Senna

**LISTA DE ILUSTRAÇÕES**

[Figura 1: Celular Ericsson MTA (Fonte: http://worldcom.net.br/blog-world/o-telefone-movel-mais-antigo-da-historia/) 15](#_Toc336811483)

[Figura 2: Dyna TAC 8000x (Fonte: http://infomais.blog.br/celulares-e-sua-evolucao/) 16](#_Toc336811484)

[Figura 3: Nokia MobiraTalkman (Fonte: http://blog.dialaphone.co.uk/2008/10/01/vintage-era-of-mobile-phones) 16](#_Toc336811485)

[Figura 4: Motorola Micro TAC (Fonte: http://www.retrobrick.com/motomicro.html) 17](#_Toc336811486)

[Figura 5: Motorola Phone Bag 2900 (Fonte:http://www.maximumpc.com/article/features/tin\_cans\_touchscreens\_40\_most\_important\_phones\_history?page=0,4) 17](#_Toc336811487)

[Figura 6: Motorola Star Tac (Fonte: http://www.maximumpc.com/article/features/tin\_cans\_touchscreens\_40\_most\_important\_phones\_history?page=0,4) 18](#_Toc336811488)

[Figura 7: Divisão da tecnologia Celular (GSM) (Fonte: http://www.teleco.com.br/tecnocel.asp) 20](#_Toc336811489)

[Figura 8: Celular por Tecnologia no Brasil. Fonte:(http://www.teleco.com.br/tecnocel.asp). 21](#_Toc336811490)

[Figura 9: Sensação de insegurança dos brasileiros. Fonte: (IBGE) 23](#_Toc336811491)

[Figura 10: Uso de dispositivo de segurança. Fonte: (IBGE) 24](#_Toc336811492)

[Figura 11: Percentual de domicílios com cada tipo de dispositivo de segurança, no total de domicílios particulares permanentes (%). Fonte: (IBGE, Pesquisa Nacional por Amostra de Domicílios 2009). 25](#_Toc336811493)

[Figura 12: Dispositivos de Segurança por Estado (Região Sudeste). Fonte: (IBGE, Pesquisa Nacional por Amostra de Domicílios 2009). 27](#_Toc336811494)

[Figura 13: Dispositivos de Segurança. Fonte: (IBGE, Pesquisa Nacional por Amostra de Domicílios 2009). 28](#_Toc336811495)

[Figura 14: Sensação de Segurança. Fonte: (IBGE, Pesquisa Nacional por Amostra de Domicílios 2009). 28](#_Toc336811496)

[Figura 15: Ferramentas Desenvolvimento Apple. Fonte: (Apple, https://developer.apple.com/) 31](file:///E:\TCC\androidino\Documento\Doc.%20TCC%203.0.1.docx#_Toc336811497)

[Figura 16: Logomarca Apple. Fonte: (Apple, http://www.gizmodo.com.br/apple-divulga-resultados-do-trimestre-numeros-enormes-nada-sobre-brasil/apple/) 32](#_Toc336811498)

[Figura 17: Logotipo Eclipse IDE de Desenvolvimento. Fonte: Eclipse (http://www.eclipse.org) 34](#_Toc336811499)

[Figura 18: Logotipo do Plugin ADT. Fonte: ADT (http://developer.android.com/sdk/index.html) 35](#_Toc336811500)

[Figura 19: Gráfico de Distribuição de versões Seg. semestre 2012. Fonte: Android developer (http://developer.android.com/about/dashboards/index.html) 36](#_Toc336811501)

[Figura 20: Conjunto de dados históricos coletados durante o período de 14 dias encerrado em 1 de setembro de 2012 Fonte: Android Developer (http://developer.android.com/resources/dashboard/platform-versions.html) 36](#_Toc336811502)

[Figura 21: Tabela Distribuição x Versões. Fonte: Android Developer (http://developer.android. com/resources/dashboard/platform-versions.html) 37](#_Toc336811503)

[Figura 22: Placa Arduino Mega 2560. Fonte:Arduino (http://arduino.cc/en/Main/Hardware) 39](#_Toc336811504)

[Figura 23: Placa Arduino Ethernet Shield. Fonte: Arduino (http://arduino.cc/en/Main/Hardware) 40](#_Toc336811505)

[Figura 24: Esquema Sensor Temperatura/Umidade x Arduino. 45](#_Toc336811506)

[Figura 25: Esquema Sensor Presença x Arduino. 46](#_Toc336811507)

[Figura 26: Esquema Sensor Chamas x Arduino. 47](#_Toc336811508)

[Figura 27: Captura de tela - IDE Arduino. 48](#_Toc336811509)

[Figura 28 Simulador de Celular rodando o sistema operacional Android 2.3. 51](#_Toc336811510)

**LISTA DE TABELAS**

[Tabela 1: Comparativo entre as Tecnologias GSM (Fonte: http://www.teleco.com.br/tecnocel.asp) 21](#_Toc336811411)

[Tabela 2: Especificação do Sensor de Umidade e Temperatura. 44](#_Toc336811412)

[Tabela 3: Especificação do Sensor de Presença. 45](#_Toc336811413)

[Tabela 4: Especificação do Sensor de Chamas. 46](#_Toc336811414)

**LISTA DE ABREVIATURAS E SIGLAS**

3GPP - 3rd Generation Partnership Project (3º Geração de Parceria de Projetos)

ADT - Android Development Tools (Ferramenta de Desenvolvimento Android)

ADV - Android Virtual Device (Dispositivo virtual Android)

AMPS - Advanced Mobile Phone System (Sistema de Telefonia Móvel Avançado)

API - Application Programming Interface (Interface entre Aplicativo e Programação)

ºC - Graus Celsius

CDMA - Code Division Multiple Access (Acesso Múltiplo por Divisão de Código)

EDGE - Enhanced Data rates for GSM Evolution (Taxa de dados avançado para

EEPROM- Electrically-Erasable Programmable Read-Only Memory- (Memória programável eletricamente apagável somente de leitura)

GSM Evoluido)

GBPS - *Gigabit* por segundo

GND - Ground (Fio terra)

GPRS - General Packet Radio Service (Serviços Gerais de Pacote por Rádio)

GSM - Global System for Mobile Communications (Sistema Global Comunicações Móveis)

HSPA - High Speed Packet Access (Acesso em Pacotes em Alta Velocidade)

HTTP *HyperText Transfer* Protocol (Protocolo de Transferência de Hyper Texto)

IBGE - Instituto Brasileiro de Geografia e Estatística

ICSP - In Circuit Serial Programming (Entrada Serial de Programação)

IDE - Integrated Development Environment (Ambiente de Desenvolvimento Integrado)

IP - Internet Protocol (Protocolo de Internet)

KBPS - Kilobit por segundo

LTE - Long Term Evolution (Longo Termo de Evolução)

MBPS - *Megabit* por segundo

MS - Milissegundos

NM - Nanômetros

PNAD - Pesquisa Nacional de Amostras por Domicílios

SD Card - Secure Digital Card (Cartão Segurança Digital)

SDK - Software Develop Kit (*Kit* de Desenvolvimento de *Software*)

SMS - Short Message Service (Serviço de Mensagens Curtas)

TDMA - Time Division Multiple Access (Acesso Múltiplo por Divisão de Tempo)

UDP - User Datagram Protocol (Protocolo detagrama ao usuário)

UMTS - Universal Mobile Telecommunications Service (Sistema Universal de Comunicação Móvel)

UR - Umidade Relativa

USB - Universal Serial Bus (Barramento Serial Universal)

USD - United Stades Dollar (dolar dos Estados Unidos)

V - *Volts*

WCDMA - Wide-Band Code-Division Multiple Access (Banda Larga Acesso Múltiplo por Divisão de Código)

WIMAX - Wordwide Interoperability for Microwave Access (Interoperabilidade Mundial para Acesso de Micro-ondas)

**SUMÁRIO**

[INTRODUÇÃO 12](#_Toc336811299)

[OBJETIVO GERAL 13](#_Toc336811300)

[OBJETIVOS ESPECÍFICOS 13](#_Toc336811301)

[JUSTIFICATIVA 13](#_Toc336811302)

[PROBLEMATIZAÇÃO 14](#_Toc336811303)

[METODOLOGIA 14](#_Toc336811304)

[ORGANIZAÇÃO DA ARGUMENTAÇÃO 14](#_Toc336811305)

[1 – Celular e seus Serviços 15](#_Toc336811306)

[1.1 – História do Aparelho 15](#_Toc336811307)

[1.2 - No Brasil 18](#_Toc336811308)

[1.3 - Primeira Geração - 1G 18](#_Toc336811309)

[1.4 - GSM – 2G 19](#_Toc336811310)

[1.5 - Terceira Geração - 3G 19](#_Toc336811311)

[1.6 - 4G 22](#_Toc336811312)

[2 – Segurança Domiciliar 23](#_Toc336811313)

[3 – ESTUDO DE CASO: iOs x Android 30](#_Toc336811314)

[3.1 – iOs 30](#_Toc336811315)

[3.2 – Android 33](#_Toc336811316)

[4 – Hardwares Open Source 38](#_Toc336811317)

[4.1 – Arduino 38](#_Toc336811318)

[4.2 – Arduino Ethernet Shield 40](#_Toc336811319)

[5 – Projeto Proposto: Androidino - Alarme Residencial usando Ferramentas Open Source 43](#_Toc336811320)

[5.1 – O alarme 43](#_Toc336811321)

[5.1.1 – Hardware 43](#_Toc336811322)

[5.1.1.1 – Sensores 44](#_Toc336811323)

[5.1.2 – Comunicação 47](#_Toc336811324)

[5.1.2.1 – Interação com a rede social Twitter 48](#_Toc336811325)

[5.2 – Aplicação 49](#_Toc336811326)

[5.2.1 – Android 49](#_Toc336811327)

[5.2.1.1 – O Aplicativo 49](#_Toc336811328)

[5.2.1.2 – Linguagem de Programação 50](#_Toc336811329)

[5.3 – Funcionamento 51](#_Toc336811330)

[CONCLUSÃO 53](#_Toc336811331)

[REFERÊNCIAS 54](#_Toc336811332)

# INTRODUÇÃO

Inicialmente abordaremos uma visão geral sobre a tecnologia móvel, sua importância e como está difundida no cotidiano das pessoas.

No decorrer do capítulo II iremos apresentar alguns dados estatísticos sobre os índices de roubo a residências, bem como apresentaremos algumas soluções adotadas comumente para combater esses índices e tecnologias que são usadas a favor da segurança.

Serão apresentadas as principais tecnologias para desenvolvimento de aplicativos móveis usadas atualmente, com o objetivo de desenvolver ferramentas que auxiliem no monitoramento residencial.

Mostraremos as vantagens e desvantagens em utilizar ferramentas de desenvolvimento *open source* com o objetivo de direcionar o desenvolvimento para determinada plataforma com base nas pesquisas e análises desenvolvidas no decorrer do projeto.

Iremos abordar sobre microcontroladores *open source* que se enquadre na construção do Sistema de Monitoramento Residencial. escolhendo o microcontrolador, explicaremos como utilizá-lo.

Após o estudo sobre as tecnologias iremos apresentar a nossa proposta, que é o desenvolvimento de um dispositivo, capaz de monitorar uma residência de forma remota utilizando *software* e *hardware open source*.

# OBJETIVO GERAL

Avaliação das principais tecnologias atuais e implementação de um aplicativo móvel  
integrado com microcontrolador visando uma solução de monitoramento de sistema de alarme.

# OBJETIVOS ESPECÍFICOS

* Estudo de duas ferramentas para desenvolvimento de aplicativos móveis utilizados atualmente.
* Análise das ferramentas de desenvolvimento e dos circuitos eletrônicos de microcontroladores.
* Levantamento dos possíveis protocolos de comunicação.

# JUSTIFICATIVA

Dados da Polícia Militar informam que, nos primeiros 15 dias do ano de 2012, a cidade de São Paulo teve uma residência assaltada a cada quatro horas. Os períodos mais críticos são manhã e noite, quando as pessoas estão saindo ou chegando em casa.

Atualmente vivemos em um mundo que nos consome com outras necessidades e nem sempre conseguimos cuidar dos bens que ao longo dos anos fomos adquirindo.

Os altos custos com seguradoras e alarmes não nos deixam outra escolha a não ser a vivermos enjaulados.

Pensando nessa necessidade, porque não utilizarmos da tecnologia que dispomos no dia-a-dia para monitorar nossas residências?

Devido ao fácil acesso de aquisição e viabilidade dos aparelhos celulares com capacidade de comunicação e processamento cada vez maior, o uso desses aparelhos no cotidiano das pessoas se tornou frequente. Segundo a Anatel, atualmente, o Brasil tem 116,5 linhas habilitadas para cada grupo de 100 pessoas. (2011, INFO ABRIL).

Como o fator que muitas vezes impulsiona o mercado é o custo, não poderíamos deixar de abordá-lo. Durante sua palestra Schmidt afirmou: "Acredito que o ponto de virada é quando os *smartphones* passarem a custar menos de US$ 70, mas já no ano que vem veremos *smartphones* com Android a menos de US$ 100" (2012, EXAME).

Este trabalho terá foco no desenvolvimento de uma aplicação de monitoramento de alarme residencial para dispositivos móveis capaz de efetivar comandos ao alarme remotamente.

# PROBLEMATIZAÇÃO

* O microcontrolador será capaz de se comunicar com um aplicativo desenvolvido em uma ferramenta de código aberto?
* Existe um protocolo que suporte a comunicação entre os dispositivos?
* Será possível gerenciar o microcontrolador de forma remota?

# METODOLOGIA

Os métodos utilizados para o desenvolvimento deste estudo serão:

* Leitura de livros, artigos e revistas com o objetivo de adquirir conhecimento sobre segurança domiciliar, tecnologias disponíveis para comunicação de alarme residencial e celular;
* Pesquisa em *sites* especializados sobre microcontroladores visando o conhecimento mais atualizado;
* Leitura de livros e pesquisa sobre plataformas de desenvolvimento de aplicativos para aparelhos móveis e *hardwares* *open source*.

# ORGANIZAÇÃO DA ARGUMENTAÇÃO

A organização do trabalho segue da seguinte forma:

No primeiro capítulo será apresentada uma fundamentação teórica sobre a história do celular e seus serviços e também sobre segurança domiciliar.

No segundo capítulo serão apresentados dados de amostra de uma pesquisa feita pelo IBGE sobre insegurança domiciliar.

No terceiro capítulo apresentaremos um estudo sobre duas principais plataformas para desenvolvimento de aplicativos para aparelhos móveis.

No quarto capítulo descreveremos sobre os *hardwares* *open source* que será usado no desenvolvimento do trabalho para interagir com a aplicação.

No quinto capítulo apresentaremos o projeto Alarme Residencial Androidino a ser desenvolvido.

# 1 – Celular e seus Serviços

Neste capítulo serão apresentados alguns aspectos teóricos sobre a história do aparelho celular, desde o seu surgimento e também a evolução doos seus serviços desde o primeiro lançamento.

## – História do Aparelho

O celular está cada vez mais presente no cotidiano das pessoas, tem se tornado cada vez mais um item indispensável, mas nem sempre foi assim, o aparelho celular como conhecemos hoje, sofreu várias mudanças consideráveis desde sua invenção.

O primeiro aparelho considerado celular foi o Ericsson MTA (MobilieTelephony A) que pesava por volta de 40 quilos, porém o portátil se restringia a ser transportado em carro, barcos e não era uma tecnologia tão acessível como conhecemos hoje. O MTA operava na frequência de rádio de 160 MHz.



Figura 1: Celular Ericsson MTA (Fonte: http://worldcom.net.br/blog-world/o-telefone-movel-mais-antigo-da-historia/)

Em 1983 foi a vez do Motorola Dyna Tac 8000X que pesava aproximadamente 1 quilo tinha 7 cm de largura e 25 cm de comprimento seu preço iniciava em $3999 dólares e podia chegar até pouco mais de $8600 dólares. Comparado ao MTA o DynaTac era uma grande evolução tanto no quesito peso quanto no *design*. Porém proporcionava 60 minutos de conversação em condições favoráveis.



Figura 2: Dyna TAC 8000x (Fonte: http://infomais.blog.br/celulares-e-sua-evolucao/)

Em 1984 um ano depois do lançamento do Dyna Tac, a empresa NOKIA, mundialmente conhecida atualmente, apresentou o Nokia Mobira Talkman que tinha o intuito de proporcionar horas de conversação contínuas, porém não aparentava ser muito confortável.



Figura 3: Nokia MobiraTalkman (Fonte: http://blog.dialaphone.co.uk/2008/10/01/vintage-era-of-mobile-phones)

No final da década de 1989 a Motorola lançou novamente um celular conceito, agora com tamanho reduzido, que poderia ser colocado no bolso, foi o lançamento do Motorola Micro TAC.



Figura 4: Motorola Micro TAC (Fonte: <http://www.retrobrick.com/motomicro.html>)

Em 1994 foi à vez do **Motorola Phone Bag 2900 que continha um transportador e bateria, podia ser levado nos ombros, porém o seu volume limitava o transporte a carros.**



Figura 5: Motorola Phone Bag 2900 (Fonte:<http://www.maximumpc.com/article/features/tin_cans_touchscreens_40_most_important_phones_history?page=0,4>)

**Em 1996, a Motorola lançou o modelo que se popularizou por ser mais leve e pequeno estava lançado o Star Tac. Era um celular de *flip* que permitia dobrar o celular quando não estivesse em uso.**



Figura 6: Motorola Star Tac (Fonte: <http://www.maximumpc.com/article/features/tin_cans_touchscreens_40_most_important_phones_history?page=0,4>)

## - No Brasil

A primeira rede de telefonia celular do [Brasil](http://pt.wikipedia.org/wiki/Brasil) foi lançada pela TELERJ, na cidade do Rio de Janeiro em 1990.

A denominação “celular” é utilizada no Brasil devido a sua topologia de rede, onde cada região ou área de transmissão é uma célula.

**Até a década de 90 a tecnologia que permitia a comunicação entre os aparelhos celulares é a chamada 1G.**

**Hoje com os avanços das tecnologias é possível realizar além de chamadas telefônicas, enviar e receber mensagens de texto, mensagens multimídias, acessar a internet, ler e enviar *e-mail*, entre muitos outros recursos que só se tornaram possíveis com a evolução tanto do *hardware* do aparelho quando dos protocolos de comunicações.**

## ****- Primeira Geração - 1G****

A primeira **geração do serviço móvel passou a funcionar através de cidades ou regiões de pequenas áreas geográficas denominadas células, origem do nome “celular”. Cada uma dessas células é transmitida pelo seu próprio conjunto de rádios transmissores e receptores de baixa potência. Quando a chamada de um celular alcança uma torre de transmissão e recepção, a mesma é transferida para o sistema de telefonia fixa. Cada célula possui diversos canais com o objetivo de prover serviços para muitos usuários simultaneamente. À medida que um usuário se movimenta na área de cobertura, o sinal do seu telefone celular passa automaticamente de uma célula para outra, sem sofrer interrupções. O primeiro sistema telefônico celular ficou conhecido pela sigla AMPS (*Advanced Mobile Phone System*). O AMPS é formado por sistemas analógicos permitindo assim apenas transmissão de voz.**

**No Brasil o AMPS funciona em conjunto com as tecnologias CDMA (***Code Division Multiple Access*, ou Acesso Múltiplo por Divisão de Código**) e TDMA (***Time Division Multiple Access,* ou Acesso Múltiplo por Divisão de Tempo**).**

* 1. **- GSM – 2G**

**O GSM (***Global System for Mobile Communications*) **ou Sistema Global para Comunicações Móveis é considerado a segunda geração por ser digital. É uma tecnologia e o padrão mais popular para os telefones móveis. Alguns serviços como troca de mensagem foram originalmente desenvolvidos para o GSM. A tecnologia GSM possui baixo custo de infraestrutura.**

* **Serviços:** Serviços de comunicação entre dois assinantes como telefonia, serviço de mensagens curtas (SMS) e FAX; Serviços de localização padronizados para o GSM permitem estimar com precisão a localização da estação móvel servindo de base para vários serviços oferecidos ao assinante.

Serviços de transporte de dados usados para conectar dois elementos de uma rede com taxas de dados de 2400 a 9600 bit/s.

O **GPRS** (*General Packet Radio Service*) é um serviço para comunicação de dados que permite a estação móvel uma conexão a Internet sem a necessidade de se estabelecer uma chamada telefônica. Este serviço implica em uma taxa que teoricamente poderia chegar a 115 kbit/s.

O **EDGE** (*Enhanced Data rates for GSM Evolution*) é um padrão desenvolvido para aumentar a taxa de dados para serviços oferecidos pela rede GSM o que possibilitaria o oferecimento de conexões IP (*Internet Protocol)* de até 384 kbit/s.

* 1. **- Terceira Geração - 3G**

A evolução do GSM para serviços de terceira geração com taxas de dados de até 2Mbit/s vem sendo padronizada pelo terceiro *Generation Partnership Project* (3GPP). Suportar um número maior de clientes de voz e dados, permite transmissão de 384 kbits/s em sistemas móveis e até 7 megabits/s em sistemas fixos.

**UMTS (Universal Mobile Telecommunications Service):** É a evolução do GSM mais que ainda se baseia nessa tecnologia (GSM), embora o seu acesso por rádio seja diferente. Essa tecnologia usa uma técnica CDMA chamada Direct Sequence Wideband (DS-WCDMA), por isso é comum o uso intercalado de UMTS e WCDMA, embora a sigla UMTS se refira ao sistema inteiro.

A UMTS é uma tecnologia baseada em IP que suporta voz e dados em pacotes oferecendo taxas máximas de transmissão de dados de até 2 Mbps e velocidades médias de 220 a 320 kbps quando o usuário está andando ou dirigindo. Essa tecnologia foi desenvolvida para prover serviços com altos níveis de consumo de banda, como *streaming*, transferência de grandes arquivos e videoconferências para uma grande variedade de aparelhos como telefones celulares, PDAs e laptops.

A UMTS é compatível com a EDGE e a GPRS permitindo ao usuário sair de uma área de cobertura UMTS e ser automaticamente transferido para uma rede EDGE ou GPRS, dependendo de fatores como disponibilidade de rede e o consumo de banda do seu aplicativo. Assim, os usuários UMTS têm sempre assegurado um nível de serviço de pacotes de dados em casa e em viagem.

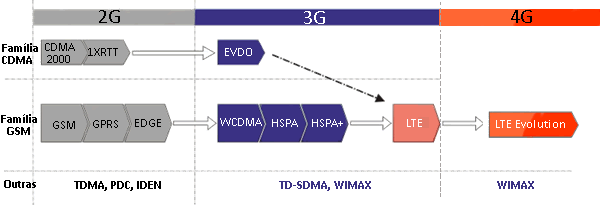


Figura 7: Divisão da tecnologia Celular (GSM) (Fonte: http://www.teleco.com.br/tecnocel.asp)

Segue tabela com comparativo entre as tecnologias GSM das gerações 2G, 3G e 4G, onde mostram os valores das taxas máximas de dados teóricos, média de taxas de dados e canalização para cada tecnologia (GSM, GPRS, EDGE, WCDMA, HSPA, HSPA+, LTE e LTE Advanced).

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Geração** | **2G** | | | **3G** | | | | **4G** |
| Tecnologia | GSM | GPRS | EDGE | WCDMA (UMTS) | HSPA (WCDMA) | HSPA+ | LTE | LTE-Advanced |
| Taxa de dados máx. teórica | 14,4 Kbps | 171,2 Kbps | 473.6 Kbps | 2,0 Mbps | 14,4 Mbps | 21/28/42 Mbps | 100  Mbps | 1,0  Gbps |
| Taxa de dados média (kbit/s) | - | 30-40 | 100-130 | 200-300 | 550-1100 | - | - | - |
| Canalização (MHz) | 0,2 | 0,2 | 0,2 | 5 | 5 | 5 | 5 |  |

**Tabela 1: Comparativo entre as Tecnologias GSM** (Fonte: http://www.teleco.com.br/tecnocel.asp)

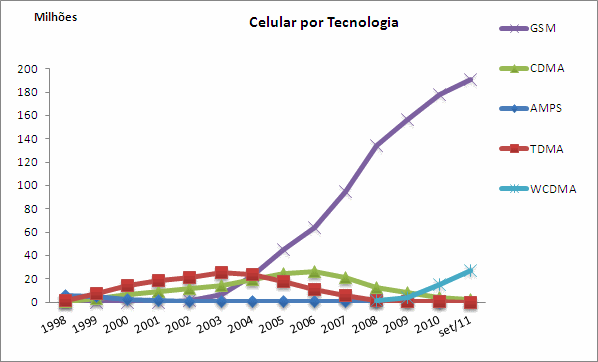


Figura 8: Celular por Tecnologia no Brasil. Fonte:(http://www.teleco.com.br/tecnocel.asp).

## - 4G

A quarta geração de telefonia móvel não pode ser considerada uma evolução da tecnologia, como ocorria com o 3G. O 4G vai além da tecnologia para celulares tornando-se um padrão de conexão de rede móvel, onde as conexões serão feitas por IP, o que abrange muito mais dispositivos do que apenas celulares.

Para que tenha acesso a esta nova tecnologia é preciso que os aparelhos tenham embargados em sua arquitetura o WIMAX (Interoperabilidade Mundial para Acesso por Micro ondas) ou LTE (Long Time evolution). Através destas arquiteturas, será possível utilizar o 4G onde o sinal for disponibilizado.

A tecnologia pretende atingir 0,5 Gbps em dispositivos em movimento e 1,5 Gbps em dispositivos parados.

WIMAX “trata-se de uma tecnologia de banda larga sem fio, capaz de atuar como alternativa a tecnologias como cabo e DSL na construção de redes comunitárias e provimento de acesso de última milha. Em teoria, espera-se que os equipamentos Wi-Max tenham alcance de até 50Km e capacidade de banda passante de até 70 Mbps." (RNP,2005).

O LTE segundo a TELECO é o novo padrão de tecnologia de banda larga móvel, Trabalha com canais de 1,4, 3,5, 10, 15 e 20 MHz, as velocidades mais altas só são obtidas com canais de 20 MHz, quando é possível obter 100 Mb/s de velocidade de pico de *downlink* e 50 Mb/s de *uplink*.

“A melhor faixa de frequência para a implantação de 4G é a de 700 MHz, liberada com o fim da transição da TV Aberta analógica para a digital. No Brasil isto deve ocorrer em 2016. A Anatel destinou para o 4G [(Res. 544 de 11/08/2010)](http://www.anatel.gov.br/Portal/verificaDocumentos/documento.asp?numeroPublicacao=255900&assuntoPublicacao=null&caminhoRel=In%EDcio-Radiofreq%FC%EAncia-Apresenta%E7%E3o&filtro=1&documentoPath=255900.pdf) a faixa de frequências de 2.500 MHz a 2.690 MHz anteriormente destinada ao MMDS” (TELECO,2012).

# – Segurança Domiciliar

Segundo dados da última Pesquisa Nacional de Amostras por Domicílios (Pnad) realizados pelo IBGE (Instituto Brasileiro de Geografia e Estatística) 78,6% das pessoas de 10 anos ou mais de idade se sentiam seguras no domicílio em que residiam, 67,1% no bairro e 52,8% na cidade.

A pesquisa revela que a população que com maior rendimento médio mensal domiciliar per capita (82,8 para cinco ou mais salários mínimos contra 77,8 para menos de ¼ de salário mínimo) sente mais segura, Esta relação se inverteu quando analisado por bairro ou na cidade.

O que aponta que o poder aquisitivo é um fator que na maioria das vezes pode proporcionar mais segurança aos moradores.

O fato de poder realizar investimentos na segurança domiciliar ou ainda habitar bairros ou regiões metropolitanas que não estejam tão sucintas a criminalidade, também influencia na pesquisa, pois em regiões onde os índices de criminalidades são menores, a sensação de segurança tende a ser maior.

Figura 9: Sensação de insegurança dos brasileiros. Fonte: (IBGE)

A pesquisa do Pnad revela que 60% dos 58,6 milhões de domicílios particulares em 2009, cerca de 34,8 milhões usavam pelo menos um dispositivo de segurança em suas residências sendo eles:

* Grade na janela/porta;
* Olho mágico,
* Abertura na porta,
* Corrente no trinco da porta ou interfone;
* Cerca eletrificada,
* Muro ou grade com mais de 2 metros de altura ou com cacos de vidro ou arame farpado, e/ou alarme eletrônico;
* Fechaduras extras e/ou barras na porta/janela contra arrombamento;
* Cachorro;
* Câmera de vídeo;
* Segurança privada e/ou cancela, entre outros.

Figura 10: Uso de dispositivo de segurança. Fonte: (IBGE)

Observando o gráfico a região que mais utiliza dispositivos de segurança em suas residências é a Centro Oeste, responsável por 64,9% dos 34,8 domicílios particulares, a região sudeste fica logo atrás com 63,9% seguida pela região sul e norte com 59,6% e 54,3% respectivamente.

A região Norte é apontada como a maior porcentagem de pessoas com sensação de insegurança e é a que possui menos dispositivos para proteção residencial.

**Percentual de domicílios com cada tipo de dispositivo de segurança, no total de domicílios particulares permanentes (%)**

Figura 11: Percentual de domicílios com cada tipo de dispositivo de segurança, no total de domicílios particulares permanentes (%). Fonte: (IBGE, Pesquisa Nacional por Amostra de Domicílios 2009).

O gráfico ilustra por região do país, qual o percentual de dispositivos de segurança mais utilizados para a proteção dos domicílios brasileiros.

Em todo o país os itens mais usados para a proteção são:

* Grade na Janela e/ou Porta 35,7%;
* Olho mágico, abertura na porta 20,4%;
* Cercas elétricas, muros altos 18,8%;
* Fechaduras extras, barras 18,4%;
* Cachorro 9,4%;
* Segurança privada 6,7%;
* Câmera de vídeo 4,2%;
* Outros mecanismos 1,3%

Como revela a pesquisa, muito dos dispositivos não envolvem tecnologia, apenas 4,2% dos imóveis possuem um circuito de câmeras para vigilância, mesmo sendo um circuito fechado de monitoramento, dependem de agentes externos para manipulá-los, como vigias, caseiros ou qualquer outro agente que dedique à vigilância constante o que aumenta o custo com as contratações dos mesmos, isto acaba sendo um fator por muitos optarem pela segurança privada (6,7%), onde uma empresa especializada fornece o serviço de segurança ao imóvel.

Como indica as estatísticas, os brasileiros ainda utilizam vários recursos considerados de menor custo, porém que garantem o mínimo de segurança ao domicílio.

Estes dispositivos são facilmente encontrados e o custo de instalação é relativamente baixo comparado aos dispositivos considerados mais seguros e avançados por possuir algum tipo de tecnologia.

**Dispositivos de Seguranças por Estado (Região Sudeste)**

Figura 12: Dispositivos de Segurança por Estado (Região Sudeste). Fonte: (IBGE, Pesquisa Nacional por Amostra de Domicílios 2009).

Figura 13: Dispositivos de Segurança. Fonte: (IBGE, Pesquisa Nacional por Amostra de Domicílios 2009).

O estado do São Paulo possui a maior parte dos imóveis com algum dispositivo de segurança, Comparado ao gráfico de Sensação de segurança (Região sudeste), São Paulo e Minas gerais são os estados onde as pessoas se sentem mais seguras.

Figura 14: Sensação de Segurança. Fonte: (IBGE, Pesquisa Nacional por Amostra de Domicílios 2009).

Entre homens e mulheres apenas no estado do Rio de janeiro as mulheres se sentem mais seguras que os homens em relação aos seus domicílios, nos demais estados a situação é inversa.

O intuito de todos esses dados e histórias sobre o surgimento e avanço tecnológico que envolve o celular tem como objetivo principal, unir a tecnologia que hoje dispomos com mais facilidade, e mostrar que é possível implantar sistemas de informações que ajudem a população a proteger seu patrimônio de forma simples e com um custo acessível.

Partindo desta premissa desenvolveremos uma aplicação que possa proporcionar aos seus usuários uma maneira simples e eficaz de monitoramento, agregando valor aos dispositivos que habitualmente são usados para fins de lazer e/ou comunicação, transformando-os em verdadeiras soluções a favor da proteção e segurança residencial.

# – ESTUDO DE CASO: iOs x Android

Apresentaremos nesse capítulo as principais características sobre as plataformas de desenvolvimento de aplicativos para aparelhos móveis: Android e o iOs. Descrevendo os ambientes de desenvolvimento, particularidades e os métodos de disponibilização das aplicações aos usuários. Tendo como finalidade escolher ao final do capítulo a tecnologia que melhor atenda as necessidades do projeto.

## – iOs

O iOs, sistema operacional da Apple está no mercado desde 2007, inicialmente foi desenvolvido para o Iphone, mas logo se tornou o sistema operacional de todos os dispositivos móveis que a Apple lança sendo o mais recente deles Apple TV.

O desenvolvimento do iOs foi baseado no hardware, construído exclusivamente para os dispositivos Apple, ou seja, somente os aparelhos da própria autora executam com sucesso o Sistema operacional.

O iOs possui características particulares aos sistemas operacionais. A primeira delas é que apenas uma aplicação pode estar ativa de cada vez, não sendo possível interagir com várias aplicações ao mesmo tempo.

**Limite de Processamento**

O desenvolvedor deve conhecer o dispositivo e os recursos oferecidos para que use o bom-senso no momento de criação do seu aplicativo. É possível bloquear a aplicação para que não seja executada em aparelhos mais antigos, onde o processamento não seja suficiente, ainda é possível restringir a versão do iOs que irá executar o aplicativo.

**Tempo de resposta**

Se a aplicação criada demorar mais de 5 segundos para responder o iOs considera como estado de ausência de resposta e aborta a execução. Este procedimento foi implementado para a Apple com o intuito que o dispositivo nunca trave.

**Tamanho de memória**

Os dispositivos mais atuais da Apple possuem entre 256 a 512 megabytes de memória RAM, Sendo que a maior parte desta memória é destina ao SO, podemos considerar que menos da metade fica livre e destinada as aplicações.

**Resolução Gráfica**

Os dispositivos possuem diversos tamanhos de tela, variando de 320x480 a 2560x1440 nos Macintosh. O Iphone 4 possui 640X960 já o Ipad (terceira geração) 2048 x 1536, essas diferenças devem ser levadas em consideração no desenvolvimento de um aplicativo.

**Ambiente de Desenvolvimento:**

O que é preciso para criar um aplicativo para o IOs?

* Possuir um computador MAC:

Não há como fugir, quando se trata de ferramentas de desenvolvimento Apple, é necessário possuir uma máquina da fabricante.

* Cadastro de desenvolvedor Apple

É preciso que seja feito o cadastro no *Iphone Developer Program* para que seja possível ter acesso às ferramentas de desenvolvimento, documentação, fóruns de discussão da Apple e exemplos de códigos.

* Software de desenvolvimento Apple

Após o cadastro no site como desenvolvedor Apple, é preciso baixar o pacote de *softwares* necessário, o IOs SDK 4.1. O Download é gratuito e estão contidos nele:

- Xcode: Ambiente de desenvolvimento equivalente às ferramentas Eclipse para programação em Java e Visual Studio para programação em linguagens da Microsoft.

- *Interface Builder*: ferramenta para criação de interfaces do Iphone e Ipod, estas ferramentas são totalmente integrada com Xcode;

- *Intruments*: Ferramenta de análise de desempenho e memória;

- Iphone Simulator: Software para simular os aplicativos em desenvolvimento para os aparelhos Iphone e Ipod.



Figura 15: Ferramentas Desenvolvimento Apple. Fonte: (Apple, <https://developer.apple.com/>)

**Plataforma de Hardware:**

Com ferramentas Apple, para desenvolver qualquer produto para os dispositivos da empresa é necessário uma maquina MAC.



**Figura 16: Logomarca Apple.** Fonte: (Apple, <http://www.gizmodo.com.br/apple-divulga-resultados-do-trimestre-numeros-enormes-nada-sobre-brasil/apple/>)

**Licenças de desenvolvimento**

Basicamente existem três formas de interagir com o desenvolvimento e licenças de aplicativos para a Apple, a primeira dela, e a mais simples, e a não utilização de licença comercial. Neste modo você terá acesso as ferramentas de desenvolvimento, no entanto não é possível testar o aplicativo em um aparelho real nem publicá-la na AppStore, ou seja, aplicativos que dependam do hardware como acelerômetro, câmera, giroscópio não poderão ser testadas adequadamente sem adquirir uma licença comercial paga.

* **Licença Standard**

A primeira licença comercial disponibilizada pela Apple é a Standard, com custo de U$$ 99,00 por ano, com esta licença é possível publicar os aplicativos desenvolvidos, mediante aprovação pela Apple, desta forma sendo possível baixar seu aplicativo no aparelho real e assim testá-lo. É uma licença individual liberada para um único acesso.

* **Licença Enterprise**

Esta licença possui todos os venefícios da Standard ao custo anual de U$$ 299,00 com a diferencial de permitir o acesso à publicação de um mesmo aplicativo por mais de uma pessoa.

**Publicação de Aplicativos para sua empresa.**

Para a publicação de aplicativos na Apple Store é preciso cadastrar sua empresa como desenvolvedora, cadastro similar ao feito para desenvolvedores. Efetuar o pagamento de USD (United Stades Dollar) 99,00 ter CNPJ (Cadastro Nacional de Pessoa Jurídica) válido e também é necessário enviar a Apple um contrato escriturado em inglês. Este processo pode demorar dias ou até meses dependendo do porte da empresa.

Prós: A empresa terá um registro próprio e poderá sempre atualizar seus aplicativos e acompanhar as vendas.

Contras: O cadastro não deve ser feito pelo programador, pois a documentação que deve ser transmitida à Apple são os documentos oficiais da empresa. Caso seja feito pelo desenvolvedor do aplicativo, reduz a burocracia, porém qualquer atualização no *app* (Aplicativo) deve ser feito pelo desenvolvedor e consequentemente o *app* leva a assinatura do desenvolvedor e não da empresa.

Após esta etapa do cadastro da empresa estiver concluída poderá ser submetida à aplicação na Apple Store e em cerca de 5 a 10 dias seu *app* será aprovado ou recusado. Se for recusado deverá ser revisado e enviado novamente para análise.

**- Público Alvo:**

Em geral a Apple busca atingir um publico mais jovem e busca inspirar novos designs, porém o custo dos seus produtos reflete a uma classe econômica com maior poder aquisitivo. Os produtos Apple seguem uma linha voltada para o ambiente gráfico, design ousado que atrai a todos que gostam de inovação.

## – Android

O sistema operacional Android tem como característica positiva a arquitetura flexível e focada em integração de aplicação. Possibilita integrar aplicações nativas com as aplicações desenvolvidas por terceiros e também substituir qualquer aplicação nativa existente por uma desenvolvida pelo próprio usuário.

O Android conta com o diferencial de ser a primeira plataforma para aplicações móvel completamente livre e de código aberto. (2010, Lecheta) A licença Apache Software Foundation (ASF) permite que os programadores façam alterações no código-fonte para criar seus aplicativos customizados sem precisar compartilhar as alterações com ninguém. Característica que desperta o interesse dos desenvolvedores, pois envolve grandes vantagens como, por exemplo, o baixo custo de investimento para criações de aplicações e maior público alvo que pode ser atingido, uma vez que o sistema operacional Android está presente em diversos fabricantes de celulares.

Outro ponto forte desse sistema operacional é ser baseado no *kernel* 2.6 do Linux, que é encarregado de gerenciar a memória, os processos, *threads* e a segurança dos arquivos, pastas, redes e drives.

**Ambiente de Desenvolvimento:**

A preparação do ambiente de desenvolvimento para aplicativos Android é simples e fácil de ser montado.

O que é preciso para desenvolver uma aplicação para Android?

- Android SDK: Pacote ou Kit de software para desenvolvimento Android, nela estão inclusas todas as API (Application Programming Interface) necessárias para o desenvolvimento de uma aplicação para o sistema. Baixando a SDK do Site Developed Android virá o ADV (Android Virtual Device) que possibilita a emulação de qualquer versão do sistema operacional, para a simulação do aplicativo sem a necessidade de um aparelho real. Porém é indispensável o teste em dispositivos móveis com o sistema Android para garantir o bom funcionamento da aplicação.

- IDE (Integrated Development Environment) de desenvolvimento: A própria Google recomenda o Eclipse como IDE de desenvolvimento, porém é possível desenvolver com outras IDE’s como, por exemplo, o Visual Studio.

- Plugin ADT (Android Development Tools) para Eclipse: Este *plugin* garante um ambiente integrado e de rápida criação de projetos Android, também usado para desenvolver a interface gráfica.



Figura 17: Logotipo Eclipse IDE de Desenvolvimento. Fonte: Eclipse (http://www.eclipse.org)



Figura 18: Logotipo do Plugin ADT. Fonte: ADT (<http://developer.android.com/sdk/index.html>)

**Plataforma de Hardware:**

Para desenvolver um aplicativo para a plataforma Android não é preciso um hardware específico, apenas uma máquina que possibilite a execução de uma máquina virtual do java e uma IDE com  ambiente integrado para desenvolvimento de software, como o Eclipse ou Visual Studio da Microsoft.

É possível desenvolver em uma máquina MAC OS x (Intel), Linux (i386) e Windows.

**Publicação de Aplicativos pela empresa.**

* Criar uma conta no Android Market, pagar USD 25,00 através de cartão de crédito, Liberação é on-line.

Prós: cadastro simples, controle dos aplicativos através deste cadastro.

Contras: Pagamento somente através de cartão de crédito

A publicação é imediata sem necessidade de aprovações. O nome do desenvolvedor ficará abaixo do nome do aplicativo.

No sistema Android é possível instalar e atualizar aplicativos sem fazer parte do Google Play, o aplicativo só precisa no cartão de memória do aparelho celular, via USB (Universal Serial Bus) ou *e-mail*.

**- Público Alvo:**

Não há um público em específico, os aplicativos podem ser desenvolvidos tanto para a diversão quanto para aplicações empresariais, prova disso está na loja virtual Google Play que contém os mais diversos tipos de aplicação.

**Distribuição Atual**



Figura 19: Gráfico de Distribuição de versões Seg. semestre 2012. Fonte: Android developer (<http://developer.android.com/about/dashboards/index.html>)

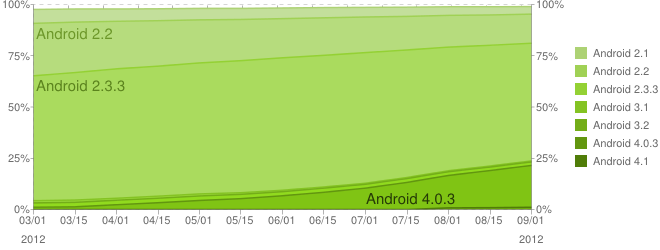


Figura 20: Conjunto de dados históricos coletados durante o período de 14 dias encerrado em 1 de setembro de 2012 Fonte: Android Developer (<http://developer.android>.com/resources/dashboard/platform-versions.html)

**Tabela de distribuição atual**

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Codename | API | Distribution |
| [1.5](http://developer.android.com/about/versions/android-1.5.html) | Cupcake | 3 | 0.2% |
| [1.6](http://developer.android.com/about/versions/android-1.6.html) | Donut | 4 | 0.4% |
| [2.1](http://developer.android.com/about/versions/android-2.1.html) | Eclair | 7 | 3.7% |
| [2.2](http://developer.android.com/about/versions/android-2.2.html) | Froyo | 8 | 14% |
| [2.3 - 2.3.2](http://developer.android.com/about/versions/android-2.3.html) | Gingerbread | 9 | 0.3% |
| [2.3.3 - 2.3.7](http://developer.android.com/about/versions/android-2.3.3.html) | 10 | 57.2% |
| [3.1](http://developer.android.com/about/versions/android-3.1.html) | Honeycomb | 12 | 0.5% |
| [3.2](http://developer.android.com/about/versions/android-3.2.html) | 13 | 1.6% |
| [4.0 - 4.0.2](http://developer.android.com/about/versions/android-4.0.html) | Ice Cream Sandwich | 14 | 0.1% |
| [4.0.3 - 4.0.4](http://developer.android.com/about/versions/android-4.0.3.html) | 15 | 20.8% |
| [4.1](http://developer.android.com/about/versions/android-4.1.html) | Jelly Bean | 16 | 1.2% |

Figura 21: Tabela Distribuição x Versões. Fonte: Android Developer (<http://developer.android>. com/resources/dashboard/platform-versions.html)

# – Hardwares Open Source

Existem poucos microcontroladores *open source* (código aberto) que são mantidos por uma comunidade ativa e que seja destinado ao usuário inexperiente. O Arduino é a exceção entre os microcontroladores do mercado, ele tem como principal característica o conceito de *software* livre, ele é desenvolvido de forma aberta, sem patente e tem uma grande comunidade ativa de usuários. Existem outros microcontroladores que possuem características e funcionalidades semelhantes ao Arduino como, por exemplo: Parallax Basic Stamp, Netmedia do BX-24, Phidgets, Handyboard do MIT (Massachusetts Institute of Technology), porém todos possuem uma linguagem complicada dos microcontroladores e não envolvem um pacote de fácil uso.

Por esse motivo optamos pelo estudo direcionado ao desenvolvimento para a plataforma do Arduino, que surgiu em 2005 como um projeto para estudantes e tem hoje cerca de 120 mil usuários (levando em consideração apenas as placas oficiais).

## – Arduino

O Arduino é um ambiente para desenvolvimento de computação física de fonte aberta, com base em uma placa simples de entrada e saída. Ele pode sentir o ambiente através de recepção de entrada a partir de uma variedade de sensores e pode afetar os seus arredores por luzes controladores, motores e outros atuadores. Pode também ser utilizado para criação de objetos interativos independentes ou conectado a *software* do computador, as placas podem ser montadas manualmente ou compradas pré-montadas e sua plataforma de desenvolvimento “IDE” de código aberto está disponível para *download* gratuito no *site* oficial [www.arduino.cc](http://www.arduino.cc).

 O Arduino é baseado nos microcontroladores Atmel ATmega8 e ATMEGA168 . Os planos para os módulos são publicados sob uma licença *Creative Commons[[1]](#footnote-1),*ou seja, pode ser adaptado e comercializado sem ferir direitos autorais e sem a obrigação de compartilhar a modificação ou a melhoria feita.

A placa Arduino consiste em uma pequena placa microcontroladora, ou seja, um pequeno circuito (placa) que contém um computador inteiro dentro de um pequeno chip (o microcontrolador). O microcontrolador na placa é programado usando a linguagem de programação do Arduino (baseado em *Wiring*[[2]](#footnote-2)) e o ambiente de desenvolvimento Arduino (baseado em *Processing*[[3]](#footnote-3)). Existem atualmente no mercado 14 placas oficiais do Arduino e 6 *shields* oficiais para serem integrados ao Arduino em diferentes projetos.



Figura 22: Placa Arduino Mega 2560. Fonte:Arduino (<http://arduino.cc/en/Main/Hardware>)

A imagem acima exibe o Arduino Mega 2560, ele tem 54 pinos digitais de entrada e saída, 16 entradas analógicas, 4 portas seriais de *hardware*, uma conexão USB, um conector de alimentação, um cabeçalho de entrada serial de programação (ICSP) e um botão de *reset*. Ele possui 256 KB de memória flash para armazenamento de código, 8 KB de SRAM e 4 KB de que pode ser lido e escrito com a biblioteca EEPROM.

A placa pode ser alimentada a partir da porta USB do computador e também de um adaptador de corrente alternada, é recomendado que seja utilizado adaptadores de 9 volts com pino redondo de 2,1 mm e centro positivo.

Para desenvolver para o Arduino é necessário primeiro fazer o *download* da IDE no *site* oficial compatível com a versão do sistema operacional do computador e em seguida instalar os *drivers* que permitem o computador se comunicar com a placa utilizando a porta USB. Após esses passos é possível iniciar o ambiente de desenvolvimento integrado, apenas executando o arquivo com extensão .exe (no windows), não havendo necessidade de instalá-lo.

A IDE é um programa executado no computador que permite criar esboços (código fonte) para a placa em uma linguagem simples e ao pressionar o botão que faz o *upload* do esboço, o código escrito é traduzido para linguagem C e transmitido para o compilador, importante *software* de código aberto que realiza a tradução final dos comandos para uma linguagem que pode ser compreendida pelo microcontrolador.

De acordo com www.arduino.cc, o Arduino tem vários fatores que o difere das outras plataformas presentes no mercado, como por exemplo:

* Ambiente multi plataforma; ele pode ser executado no Windows, Macintosh e Linux;
* É um *hardware* e *software* de código aberto;
* Tem por base um ambiente de desenvolvimento fácil de ser utilizado;
* Pode ser programado utilizando-se um cabo USB, sem necessidade de uma porta paralela;
* O *hardware* é barato, custa em média US$ 35;
* Há uma comunidade oficial ativa com em média 120 mil usuários;

## – Arduino Ethernet Shield

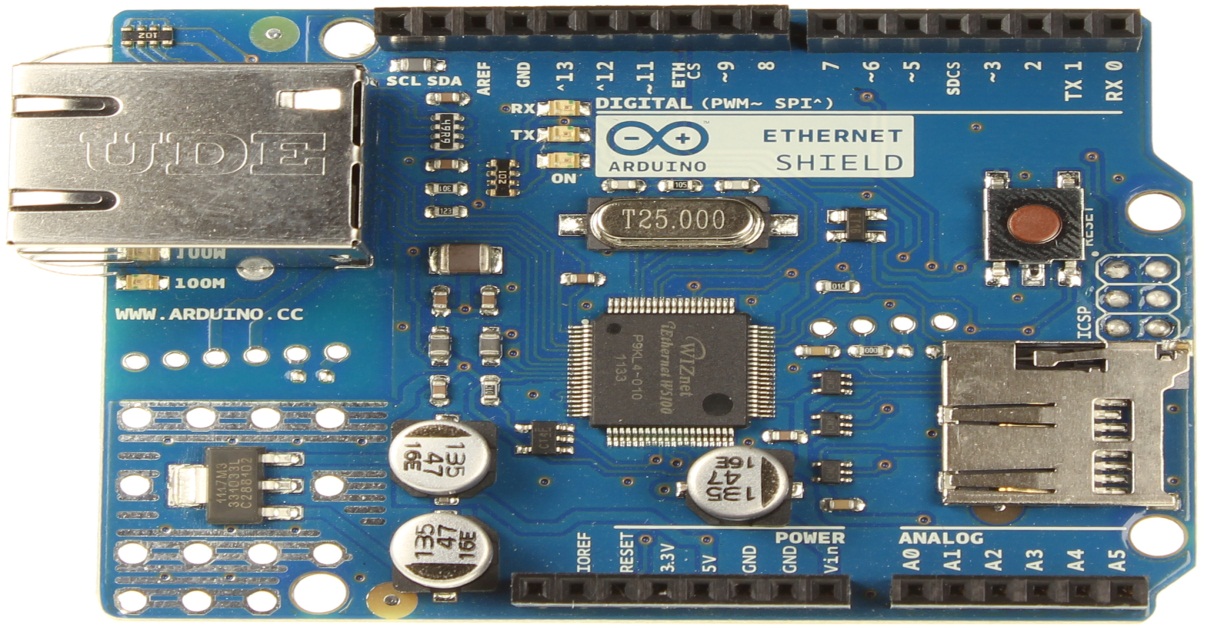


Figura 23: Placa Arduino Ethernet Shield. Fonte: Arduino (<http://arduino.cc/en/Main/Hardware>)

O Arduino Ethernet Shield permite que a placa Arduino conecte-se a internet. Ele é baseado no chip Wiznet W5100 que fornece uma biblioteca de *network* (IP) que suporta conexões TCP e UDP, e são permitidas até quatro conexões simultâneas. O *shield* *ethernet* conecta-se a placa Arduino utilizando pinos empilháveis, mantendo o leiaute e permitindo que o outro *shield* se encaixe por cima.

Ele também tem um slot de cartão micro SD que pode ser utilizado para armazenar arquivos que estejam disponíveis na rede. Esta revisão do shield também inclui um controlador de reset, para assegurar que o módulo Ethernet W5100 seja resetado apropriadamente quando a placa é ligada. O Arduino se comunica com o W5100 e com o cartão SD através do barramento SPI (Protocolo Serial de Dados Síncrono) que está nos pinos digitais 50, 51 e 52 do modelo Mega do Arduino.

A IDE do Arduino possui várias bibliotecas, que são utilizadas para abstrair comandos de baixo nível, de forma a deixar o fonte do esboço mais simples de entender e facilitar a manutenção do código. Isto é uma vantagem para quem está iniciando e acaba sendo suficientemente flexível para programadores experientes.

Utilizando as diversas bibliotecas oficias do Arduino que podem ser encontradas em

<http://arduino.cc/en/Reference/Libraries>.

Podemos construir diversos protótipos funcionais que interajam com o ambiente e com outros dispositivos. O foco deste estudo de caso está na montagem de um *web server*, que pode ser criado através do *ethernet shield.*

Para a construção de um *WEB SERVER*, com o *Ethernet Shield*, há duas bibliotecas que precisam ser importadas no esboço, são elas SPI.h e ETHERNET.h,

* Ethernet.h é uma biblioteca que possui métodos que possibilitam a comunicação do Arduino com uma rede interna (*ethernet*) ou externa (*internet*) transformando o microcontrolador em um pequeno servidor que suporta até quatro conexões simultâneas sendo elas de entrada ou saída ou uma combinação de ambas.
* SPI.h é a biblioteca que possibilita a comunicação através do barramento SPI. Este barramento é utilizado para comunicação de um ou mais dispositivos periféricos de forma rápida e em distâncias curtas, ou seja, faz a comunicação do Arduino com os *shields*.

Com as duas bibliotecas importadas no esboço é possível criar, um *web server* simples que atenda a requisições *http* e envie comandos para as portas seriais ou analógicas do Arduino, ativando/desativando outros periféricos que irão interagir com o ambiente ou outros dispositivos. Esta interação pode ser feita através de atuadores ou sensores conectados as portas do Arduino que dependendo de sua função irão retornar um estado para o mircrocontrolador, este será tratado através de programação e a partir deste ponto poderá disparar outro comando ou qualquer outra rotina que tenha sido implementada.

* 1. **– Sintaxe**

A linguagem desenvolvida para o Arduino é baseada em C/C++ AVR (microcontroladores Atmel) e a IDE em *Processing*, com o intuito de auxiliar iniciantes a desenvolver com rapidez e simplicidade.

Existem duas funções principais para a execução do esboço no Arduino, são elas: **void setup()** e a **void loop()**. Ambas são obrigatórias e void (vazio), pois é uma função sem retorno.

* A função setup() é executada apenas uma vez, no momento em que o esboço inicia-se através da ação de ligar a placa. Geralmente é usada para inicializar valores em variáveis ou preparar as portas do microcontrolador.
* A função loop() é executada após a função setup()e como o próprio nome sugere, é uma função que fica em um laço constante. Nesta função são feitas as interações com o Arduino.

# – Projeto Proposto: Androidino - Alarme Residencial usando Ferramentas Open Source

Analisando o mundo globalizado em que vivemos, é um anseio de todos que utilizam da tecnologia no seu dia-a-dia, querer que seu investimento em comunicação ou lazer também agregue funcionalidades que os ajudem a controlar, monitorar e proteger seus bens. Desta forma optamos por desenvolver um produto que atenda essa necessidade.

Discorreremos sobre o processo de montagem da parte física e também sobre o desenvolvimento da aplicação para o celular.

* 1. **– O alarme**
     1. **– Hardware**

Para construção da parte física do sistema de monitoramentofoi utilizado o microcontrolador Arduino, que atualmente é o hardware de código aberto mais acessível e que dispõe de diversos módulos que podem ser acoplados a ele para agregar novas funcionalidades ao projeto.

Criar um alarme parecia uma tarefa complexa e que demandaria muito esforço para programar em baixo nível e que o *hardware* respondesse aos comandos enviados por um celular através da *internet*. Porém com o Arduino a programação se tornou simples e eficiente, a ferramenta disponibilizada para trabalhar com o Arduino, que pode ser baixada no site oficial, é descomplicada e intuitiva. A sintaxe remete a linguagem C/C++, ou seja, não é preciso ter um conhecimento em uma linguagem de baixo nível para programar um *hardware,* o que facilita bastante para iniciantes.

A comunicação por IP foi possível por meio do Ethernet Shield wizenet w5100, este módulo é um dos *shields* oficiais do Arduino. Ele possui entrada para cabo de rede através do conector RJ45 que é modelo padrão para conexões de rede.

A montagem do Ethernet Shield Wizinet w5100é feita através dos pinos abaixo do módulo que se encaixam perfeitamente nas portas do Arduino. Após conectar o *shield* no Arduino e plugar o cabo de rede no conector RJ45 já é possível conectar a uma rede.

No site oficial arduino.cc é possível baixar a ferramenta de desenvolvimento para diversas plataformas como MAC, Linux e Windows ou ainda obter o fonte da IDE, neste projeto foi utilizado o ambiente Windows com a versão v1.0.1 da IDE, sendo necessário a instalação de um *driver* para que seja feita a comunicação através de uma porta serial (USB).

Para instalar o *driver* no Windows é preciso abrir o Gerenciador de Dispositivos e identificar o Arduino na seção de portas (COM e LPT), para isso o microcontrolador deve estar plugado no USB, deve ser feita a instalação manual do drive, clicando com o botão direito do mouse em Atualizar *driver.* O driver fica na pasta onde esta a IDE do Arduino (arduino-1.0.1\drivers). Feito este procedimento pode ser que o Windows emita um aviso dizendo que o drive não é de fonte confiável, porém deve prosseguir com a instalação. Após a instalação com êxito do *drive* o Arduino estará pronto para receber os esboços.

* + - 1. **– Sensores**

Para interagir com o ambiente o Arduino utiliza as portas seriais e analógicas, estas portar precisam estar conectadas a sensores que irão enviar algum retorno. Este retorno será tratado por uma lógica de programação que poderá acarretar em alguma ação a ser executada.

Este projeto utiliza os seguintes sensores:

* **Temperatura e umidade**

Utilizamos o sensor DHT11, este sensor requer a implementação de uma biblioteca, a mesma pode ser encontrada em: <http://arduino.cc/playground/Main/DHT11Lib>.

Esta biblioteca ajuda na utilização do sensor e poupa trabalho do desenvolvedor. Segue a baixo alguns dados técnicos do sensor utilizado:

Tabela 2: Especificação do Sensor de Umidade e Temperatura.

|  |  |
| --- | --- |
| **Sensor de umidade e temperatura** | |
| Tensão de alimentação | 3 volts a 5 volts |
| Faixa de medição de umidade | 20 a 90% UR (Umidade Relativa); |
| Faixa de medição de temperatura | 0º a 50 ºC |
| Precisão de umidade de medição | ± 5,0% UR |
| Precisão de medição de temperatura | ± 2.0 ºC |

**Esquema de ligação do sensor:**

Fio preto – alimentação 5v;

Fio vermelho – negativo (GND);

Fio branco – comunicação entrada porta serial (*Input*);

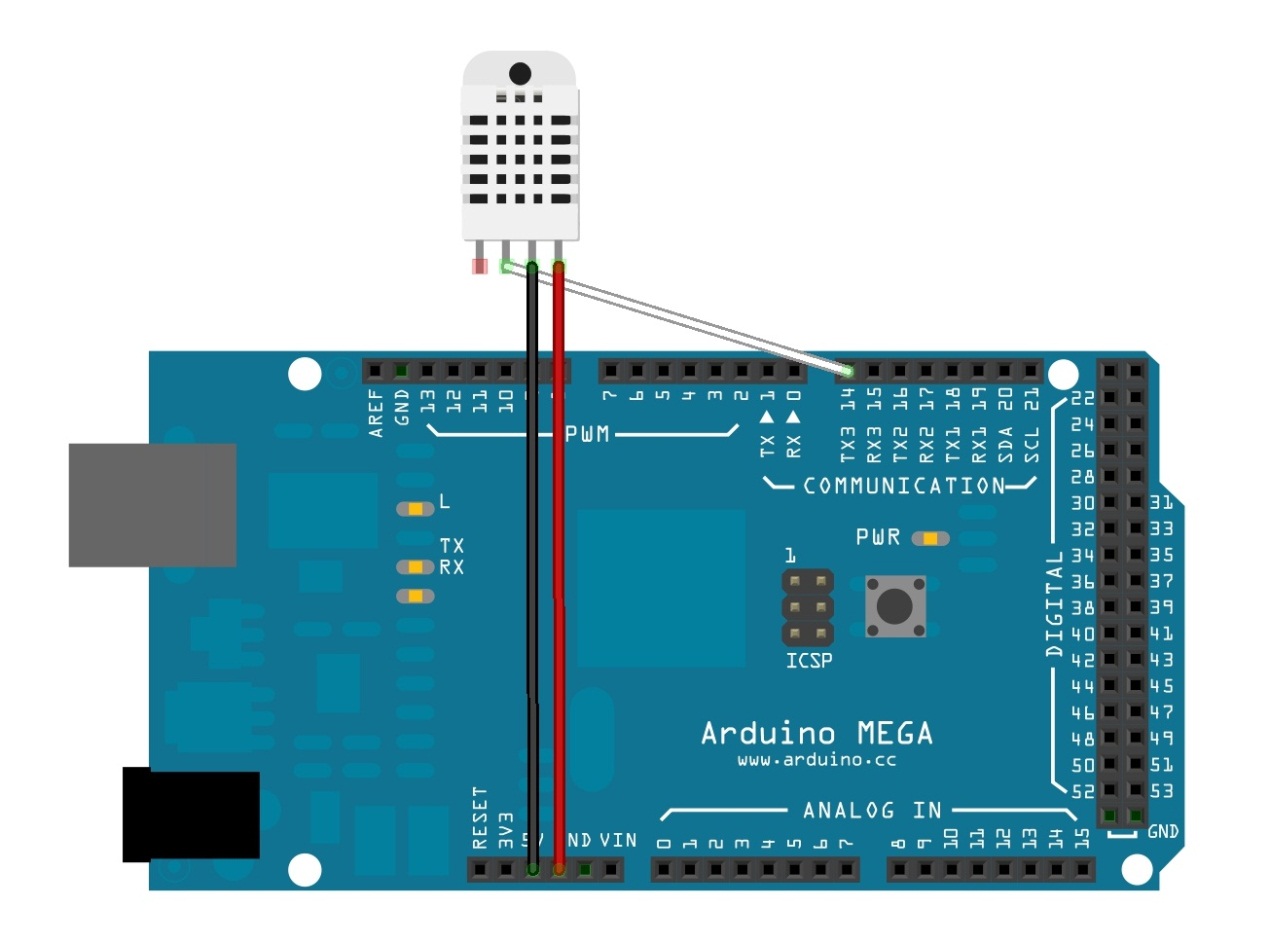


Figura 24: Esquema Sensor Temperatura/Umidade x Arduino.

* **Presença**

Utilizamos o sensor de presença de infravermelho, este sensor detecta o movimento em um ângulo de 100º com alcance de até 7 metros e pode ter o tempo de disparo ajustável.

Não é preciso nenhuma biblioteca a parte para que seja possível utilizar o sensor, podendo apenas ligá-lo na alimentação e em uma porta serial configurada como entrada (*Input*);

|  |  |
| --- | --- |
| **Sensor de Presença** | |
| Tensão de Alimentação | 3 volts a 5 volts |
| Tempo de disparo | 0,3ms a 18ms |
| Sensibilidade | Ajustável |
| Alcance | Ângulo de 100º até 7 metros |

Tabela 3: Especificação do Sensor de Presença.

**Esquema de ligação do sensor:**

Fio preto – alimentação 5v;

Fio vermelho – negativo (GND);

Fio amarelo – comunicação entrada porta serial (*Input*);

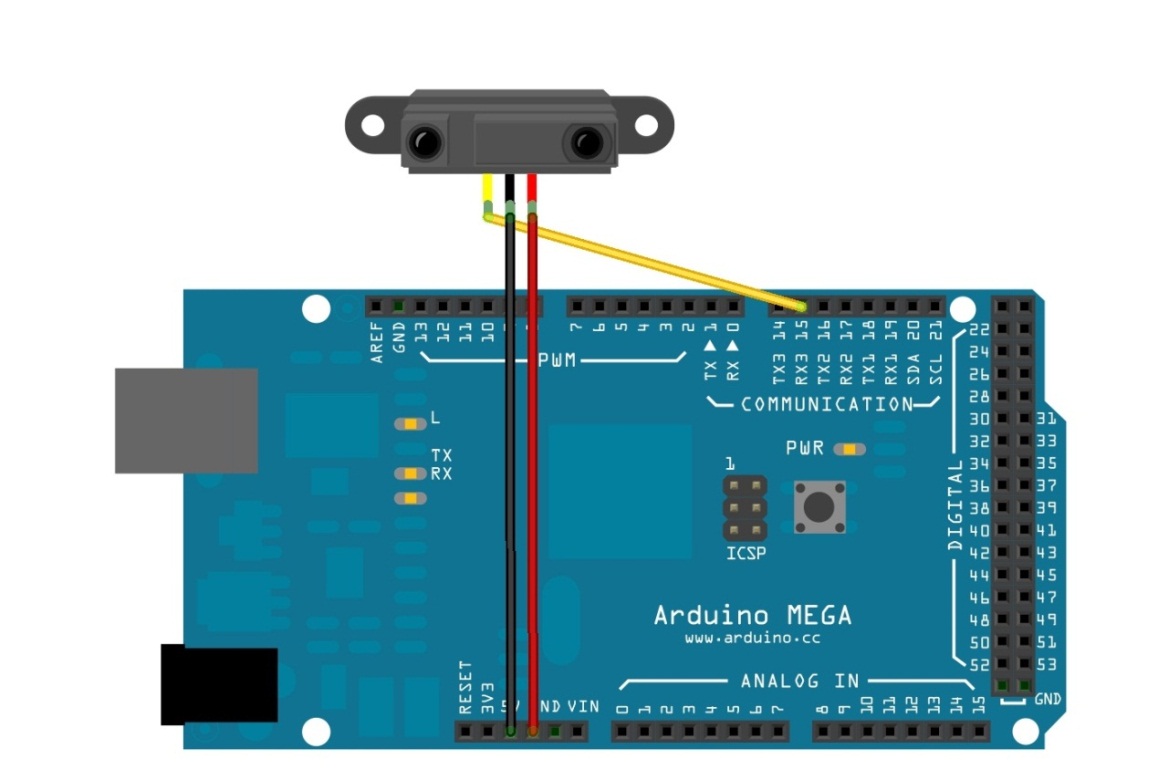


Figura 25: Esquema Sensor Presença x Arduino.

* **Chamas**

Este sensor é capaz de captar ondas de infravermelho entre 760nm a 1100 nm e pode ter a detecção de chamas ajustável. Sua instalação é simples, é preciso apenas ligá-lo a alimentação e a uma porta serial configurada como entrada;

Tabela 4: Especificação do Sensor de Chamas.

|  |  |
| --- | --- |
| **Sensor de chamas** | |
| Tensão de Alimentação | 3 volts a 5 volts |
| Temperatura de funcionamento | 0 a 50 ºC |
| Sensibilidade | Ajustável |
| Alcance | Ângulo de 60º entre 10 a cm (ajustável) |

**Esquema de ligação do sensor:**

Fio preto – alimentação 5v;

Fio vermelho – negativo (GND);

Fio amarelo – comunicação entrada porta serial (*Input*);

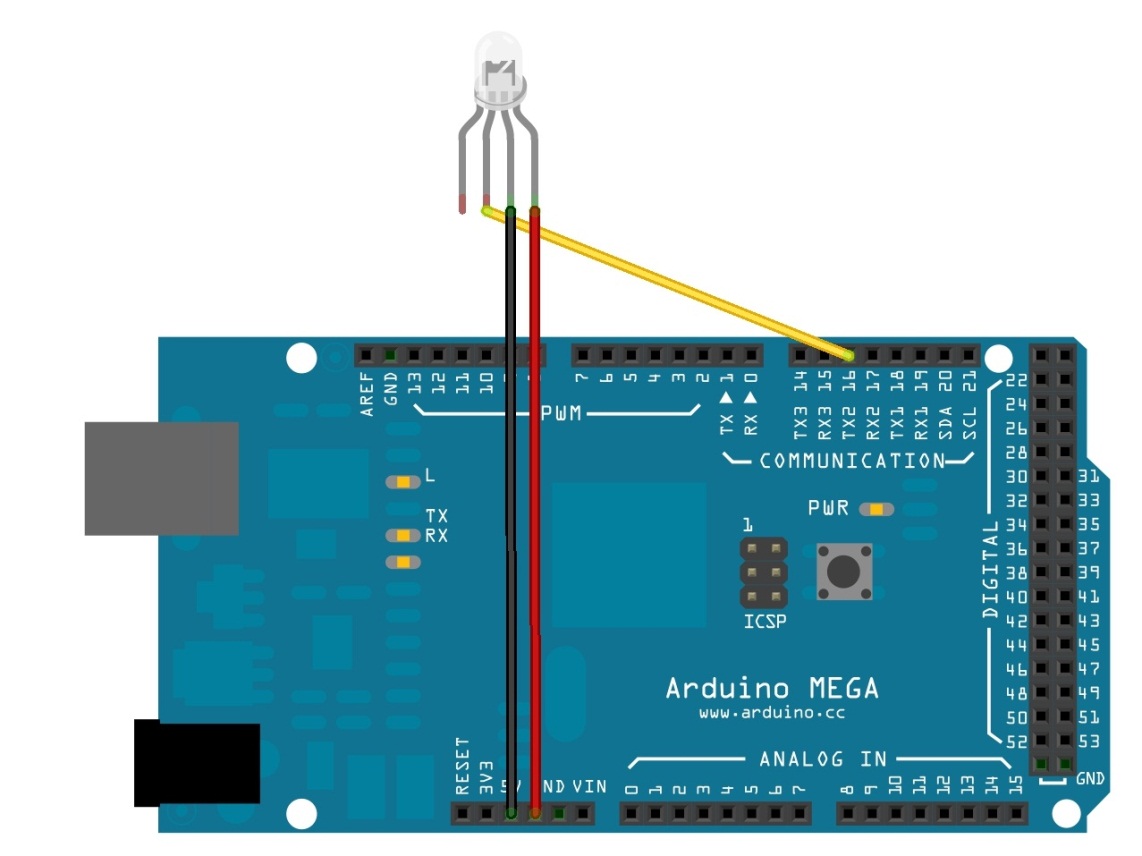
****

Figura 26: Esquema Sensor Chamas x Arduino.

* + 1. **– Comunicação**

Para a comunicação entre o hardware e a aplicação de gerenciamento optamos por usar um *Web Server*, onde ficará disponível os serviços de validação de login, redefinição de senha, ativar e desativar os sensores, cadastro do *token* de acesso ao Twitter, postagem no Twitter, ativação de todos os sensores e desativação dos mesmos. Esse *web server* pode ser construído a partir dos exemplos que são disponibilizados na IDE de desenvolvimento, onde também pode ser encontrado exemplos de *chat server*, *web cliente*.

Para abrir o exemplo pronto basta acessar a opção *file* no menu e em seguida *Examples* > *Ethernet* > *WebServer.* Após esse processo o esboço com o exemplo oficial do *Web Server* será aberto, ele também pode ser encontrado no site oficial do Arduino. O esboço pode ser alterado conforme a necessidade de cada projeto e sua sintaxe pode ser verificada a partir do botão *verify*, para se certificar que está livre de erros. Caso haja algum erro, a IDE irá exibi-lo no rodapé de informação para ser analisado e corrigido, ao contrário o esboço poderá então ser enviado para o Arduino, clicando no botão *update*.

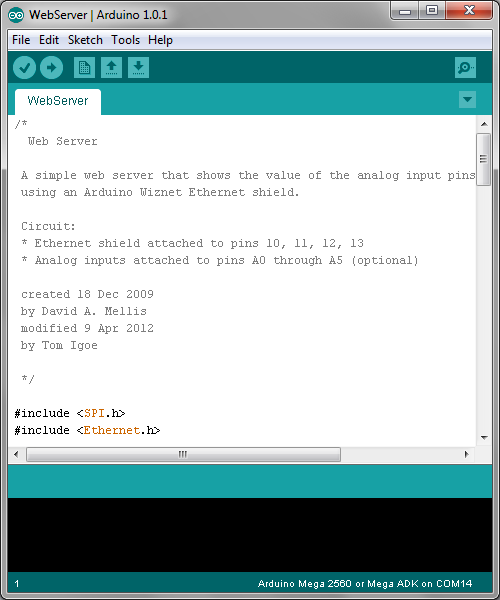


Figura 27: Captura de tela - IDE Arduino.

O exemplo retorna um HTML para o navegador de *internet* com os estados das portas analógicas do Arduino. A partir deste exemplo é possível criar com Arduino diversos serviços que geralmente são criados em um servidor a parte. Neste projeto criamos um *Web Server* que recebe requisições HTTP e retorna um serviço para a aplicação que a requisitou.

A aplicação também se comunicará com os dados do cartão micro SD disponível no *shield* Ethernet Shield Wizenet w5100 por meio de uma requisição aos serviços disponibilizados no *web server.* Neste projeto a funcionalidade do cartão micro SD esta sendo utilizada para armazenar dados de validação do usuário para acesso à aplicação, execução de operações e também para armazenar dados de configuração necessários para o bom funcionamento do alarme.

* + - 1. **– Interação com a rede social Twitter**

O Arduino possibilita que seja enviado informações para o Twitter, para isso é preciso implementar no esboço, uma biblioteca que pode ser encontrada no seguinte endereço: <http://www.arduino.cc/playground/Code/TwitterLibrary#Download>.

A comunicação com o Twitter não é de forma direta, a postagem vai para um servidor da O Auth Stuff que precisa autorizar, para que então a postagem possa ser feita em uma determinada conta. A autorização consiste em gerar um *token*[[4]](#footnote-4) da conta do Twitter para onde será enviada a mensagem. No momento da postagem da mensagem na rede social é necessário que o *token* seja enviado junto para validação de autenticidade do processo.

Limitações:

* Não é possível enviar uma mesma mensagem ao Twitter;
* Não é recomendado o envio de mais de uma mensagem ao Twitter por minuto;
  1. **– Aplicação**
     1. **– Android**

Um dos diferenciais do projeto é a mobilidade de gerenciamento do produto. Desta forma, para o desenvolvimento da interface com o usuário do projeto, escolhemos a plataforma de desenvolvimento para dispositivos móveis Android. Ele nos possibilita a criação de uma aplicação interativa, levando em consideração que a maior parte dos produtos disponíveis no mercado, têm como interface um painel fixo ao alarme para se comunicar com a central de segurança.

Nossa proposta é que o usuário gerencie o seu alarme de forma remota, de modo que possa controlar totalmente seu sistema de segurança residencial, ativando e desativando os sensores instalados. Não apenas de dentro de casa, como também, fora dela, através de uma rede IP.

* + - 1. **– O Aplicativo**

O aplicativo tem como intuito possuir uma interface com o usuário simples, porém funcional, permitindo que o usuário tenha uma experiência agradável durante o uso e que consiga realizar as tarefas facilmente e com agilidade e confiança. Recebendo sempre um *feedback* de retorno após cada ação tomada.

Ele pode ser executado em todos os aparelhos que tenham como sistema operacional padrão o Android, a partir da versão 2.3.

* + - 1. **– Linguagem de Programação**

Programar para Android requer conhecimento em Java, e assim como é recomendado no site de desenvolvimento para a plataforma do Google, o ambiente de programação deve ser feito na IDE de desenvolvimento Eclipse, devido às facilidades que os *plugins* ADT oferecem na montagem da aplicação.

Toda a documentação necessária para iniciar o desenvolvimento e pesquisa sobre funcionalidades e particularidades da plataforma do Android pode ser encontrada no site: [http://developer.android.com](http://developer.android.com/).

O SDK do Android possui uma funcionalidade interessante, o simulador do sistema operacional. Que simula perfeitamente um celular com sistema operacional Android na versão que for escolhido, isso facilita o desenvolvimento do aplicativo, pois ao compilar o projeto é enviado para o simulador e já é possível realizar os teste e até mesmo *debugar* o código fonte enquanto testa o aplicativo. Além do simulador é possível testar o aplicativo diretamente em um celular físico, pois toda vez que é compilado o projeto é gerado o arquivo com extensão .apk que é o aplicativo final. Este arquivo então, pode ser copiado na pasta “bin”, localizada no diretório do projeto, em seguida pode ser instalado no celular e testado.

Outra possibilidade viável, é poder utilizar o simulador próprio do aparelho celular com o sistema Android para testar a aplicação e também *debugar*, é necessário que a versão seja compatível com o projeto. Para isso basta então, conectar o celular ao computador via cabo USB e o sistema disponibilizará no celular disponibilizará uma funcionalidade que fica no menu Aplicação, desenvolvimento, USB *debugging*.

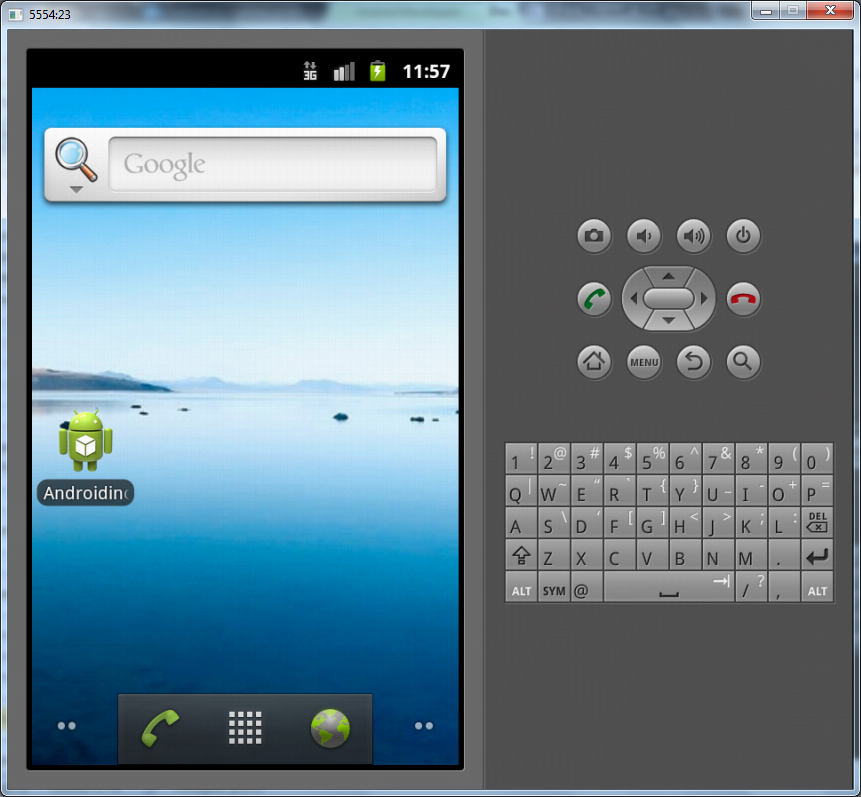


Figura 28 Simulador de Celular rodando o sistema operacional Android 2.3.

## – Funcionamento

O alarme foi construído baseado na plataforma do Arduino o qual possui todos os serviços gravados em seu chip EEPROM Electrically Erasable Programmable Read-Only Memory (Memória programável eletricamente apagável somente de leitura), estes serviços são disponibilizados em uma rede IP e que podem ser acessada através da aplicação desenvolvida para o sistema operacional Android.

Todas as requisições de serviços são validadas com o *login* (válido) de acesso do aplicativo que fica gravado na memória do celular enquanto está em uso, para ser enviado nas solicitações de todos os serviços, isso garante que a chamada do serviço seja validada. Temos como exemplo de serviço que pode se requerido pela aplicação é o acionamento de um sensor.

O acesso ao Alarme pode ser feito de qualquer lugar do mundo que possua Internet, desde que o endereço de rede e a porta de comunicação com o Arduino estejam devidamente configurados, estas configurações estão gravadas no SD acoplado ao *Ethernet* *Shield*.

**CONCLUSÃO**

O desenvolvimento deste projeto mostra que é possível ciar um sistema de monitoramento residencial, usando apenas ferramentas de código aberto.

A abordagem do segundo capitulo detalha como está a sensação de segurança das pessoas em relação ao seu imóvel, e deixou claro que os níveis de insegurança são maiores em regiões mais pobres onde o investimento em segurança é mínimo ou em muitos casos inexistente.

O projeto Androidino, como foi batizamos pela união das plataformas de desenvolvimento Android e Arduino. Tem o foco na simplicidade unindo o cotidiano das pessoas a um serviço de utilidade a um custo beneficio significativamente barato. Algo em torno de R$ 250 reais dependendo dos sensores e outros dispositivos que podem ser facilmente integrados ao sistema de monitoramento, este valor sem o investimento em um celular com o Sistema Operacional Android, porém para rodar o aplicativo não é preciso de um celular robusto, qualquer celular que seja capaz de rodar o SO da gigante das buscas na versão 2.3 em diante, será capaz de rodar nosso aplicativo.

A disseminação do sistema operacional Android, e popularização dos celulares que suportam este SO, segundo as noticias do Olhar Digital está entre 900 mil ativações diárias, oque reforça a ideia de que o mercado esta em contínua expansão.

O *hardware* é o grande segredo do projeto, a plataforma totalmente *open source* do Arduino contribuiu para que o Sistema de Monitoramento Residencial fosse feito com o nível de dificuldade reduzido, uma vez que esta foi á primeira experiência com programação física que tivemos.

A proposta foi alcançada, conseguimos construir um produto, com ferramentas de código aberto e que agregue um serviço a mais no cotidiano das pessoas, levando mais mobilidade, controle e segurança quando o assunto é a proteção de seus lares.

# REFERÊNCIAS

3GPP (Org.). **LTE.**Disponível em: <http://www.3gpp.org>. Acesso em: 03 mar. 2012.

ANDROID, Blog do (Org.). **Android Passa das 850,000 Ativações Por Dia, Colocando iOS na Mira.**Disponível em: <http://www.blogdoandroid.com/2012/02/android-passa-das-850000-ativacoes-por-dia-colocando-ios-na-mira/>. Acesso em: 03 mar. 2012.

ANDROID DEVELOPERS. **Versões da plataforma.**Disponível em: <http://developer.android.com/resources/dashboard/platform-versions.html>. Acesso em: 01 jun. 2012.

APPLE DEVELOPER. **IOs Dev Center.**Disponível em: <https://developer.apple.com/>. Acesso em: 01 jun. 2012.

ARDUINO (Org.). **Arduino**. Disponível em: <<http://arduino.cc/>>. Acesso em: 01 maio 2012.

IBGE (Org.). **Características da Vitimização e do Acesso à Justiça no Brasil.**Disponível em: <http://biblioteca.ibge.gov.br/visualizacao/monografias/GEBIS%20-%20RJ/pnadvitimizacao.pdf>. Acesso em: 03 abr. 2012.

IBGE (Org.). **47,2% das pessoas não se sentem seguras na cidade em que moram.**Disponível em: <http://www.ibge.gov.br/home/presidencia/noticias/noticia\_visualiza.php?id\_noticia=1786&id\_pagina=1>. Acesso em: 03 mar. 2012.

JORDÃO, Fabio(São Paulo) (Ed.). **História: a evolução do celular.** Disponível em: <http://www.tecmundo.com.br/celular/2140-historia-a-evolucao-do-celular.htm>. Acesso em: 22 maio 2009.

LECHETA, Ricardo R.. **Google Android: aprenda a criar aplicações para dispositivos móveis com o Android SDK.**2. ed. São Paulo: Novatec Editora, 2010.

MENEZES, Felipe. **Dicas de Empresas de como publicar aplicativos na Apple Store e Android Market.**Disponível em: <http://f7pro.blogspot.com.br/2012/01/android-apple-iphone-apple-store-felipe.html>. Acesso em: 03 jun. 2012.

MILANI, André Milani **Programando para Iphone e Ipad** 1 ed. São Paulo : Novatec Editora, 2012.

OLHAR DIGITAL (Org.). **Android: 900 mil ativações diárias.**Disponível em: <http://olhardigital.uol.com.br/negocios/digital\_news/noticias/android-900-mil-ativacoes-diarias>. Acesso em: 11 jun. 2012.

REDE NACIONAL DE ENSINO E PESQUISA (Rio de Janeiro) (Org.). **O que é Wi-Max?**Disponível em: <http://www.rnp.br/noticias/2005/not-050927-coord.html>. Acesso em: 05 maio 2012.

REVISTA COBERTURA MERCADO DE SEGUROS (Org.). **Alto índice de roubo alerta para necessidade de seguro residencial.**Disponível em: <http://www.revistacobertura.com.br/lermais\_materias.php?cd\_materias=86942&friurl=:-Alto-indice-de-roubo-alerta-para-necessidade-de-seguro-residencial->. Acesso em: 10 mar. 2012.

TELECO (São Paulo) (Org.). **Seção: 4G.** Disponível em: <http://www.teleco.com.br/lte.asp>. Acesso em: 05 maio 2012.

1. Creative Commons é uma organização não governamental sem fins lucrativos de São Francisco E.U.A que visa o compartilhamento e cópia com menos restrições e direitos autorais. [↑](#footnote-ref-1)
2. Wiring é uma plataforma de prototipagem eletrônica que permite escrever programas para controlar aparelhos conectados a ele e assim criar todo tipo de objetos interativos. [↑](#footnote-ref-2)
3. Processing é uma linguagem/ferramenta de código aberto para escrever programas que interajam com outros computadores. [↑](#footnote-ref-3)
4. Token é uma chave eletrônica que é usada para validações de acesso a aplicações de forma a fortalecer a segurança. [↑](#footnote-ref-4)