

INSTITUTO FEDERAL DA BAHIA - IFBA

WILDSON OLIVEIRA DOS SANTOS

CARPETE INTELIGENTE

Sistema para detecção de quedas de idosos utilizando plataforma Arduino

WILDSON OLIVEIRA DOS SANTOS

CARPETE INTELIGENTE

Sistema para detecção de quedas de idosos utilizando plataforma Arduino

Trabalho de Conclusão de Curso,
apresentado ao Instituto Federal da
Bahia- IFBA, como parte das exigências
para a obtenção do título de Técnico em
Eletromecânica.

Orientador: Prof. Paulo Vicente Moreira
dos Santos

WILDSON OLIVEIRA DOS SANTOS

CARPETE INTELIGENTE

Sistema para detecção de quedas de idosos utilizando plataforma Arduino

Trabalho de Conclusão de Curso,
apresentado ao Instituto Federal da
Bahia- IFBA, como parte das exigências
para a obtenção do título de Técnico em
Eletromecânica.

Aprovado em: ____ de _____
de ____.

BANCA EXAMINADORA

Prof. Paulo Vicente Moreira dos Santos
Afiliações

Prof. Miguel Neto
Afiliações

Profª. Mônica Silveira
Afiliações

RESUMO

A queda é um dos acidentes mais comuns entre a população idosa, possuindo consequências bastante significativas para esta faixa etária. As consequências de uma queda podem ser agravadas, caso a vítima seja incapaz de levantar-se sozinha após a mesma, e/ou não tenha ajuda próxima.

Com base na plataforma de prototipagem eletrônica Arduino, pensou-se num “carpete inteligente”, sistema domótico instalado na residência do idoso, capaz de detectar a ocorrência de uma queda e notificar imediatamente, por SMS, o responsável pelo mesmo e/ou um serviço de emergência.

Após estudos, um protótipo com dimensões reduzidas foi montado, e submetido a testes de desempenho. O dispositivo foi capaz de detectar grande parte das quedas simuladas e de enviar SMS para os números programados, embora algumas melhorias ainda sejam necessárias para um funcionamento completamente eficaz do dispositivo.

Palavras-chave: Domótica, Arduino, Idosos, Queda.

SUMÁRIO

1. INTRODUÇÃO	5
2. OBJETIVO GERAL	6
3. FUNDAMENTAÇÃO TEÓRICA.....	6
3.1. Análise do público alvo.....	6
3.2. Robótica	8
3.3. Domótica	9
3.4. A plataforma Arduino.....	10
3.4.1. A placa Arduino	11
3.4.2. O IDE Arduino.....	13
4. METODOLOGIA	13
4.1. Estudo da plataforma Arduino.....	14
4.2. Análise da relação Proteção X Autonomia	14
4.3. Busca por dispositivos semelhantes no mercado	15
4.4. Projeto.....	16
4.5. Descrição dos sistemas	17
4.5.1. Sistemas de entrada.....	17
4.5.2. Sistema de controle	22
4.5.3. Sistema de saída	23
4.6. Orçamento	24
4.7. Montagem	24
4.7.1. Montagem do protótipo	24
4.7.2. Montagem na protoboard	28
5. RESULTADOS ALCANÇADOS	31
6. DIFICULDADES ENCONTRADAS	32
6.1. Dificuldade de aquisição de componentes	32
6.2. Falhas de funcionamento	32
7. POSSÍVEIS MELHORIAS	32
8. CONCLUSÃO	33
9. REFERÊNCIAS.....	34
ANEXOS	35
ANEXO 1: Código utilizado em teste com piezos	35
ANEXO 2: Orçamento dos itens utilizados no projeto	36
ANEXO 3: Código utilizado no Arduino Mini	37
ANEXO 4: Código utilizado no Arduino Uno com o Shield GSM	41

1. INTRODUÇÃO

A população brasileira está passando por um processo de envelhecimento, devido principalmente à redução nas taxas de natalidade e fecundidade e aumento na expectativa de vida. Em 1999, 3,9% da população brasileira (6,4 milhões de pessoas) possuíam idade superior a 70 anos, número que em 2009 atingiu 5,1%, correspondendo a 9,7 milhões de idosos. (IBGE, 2010, p.28).

O envelhecimento é caracterizado pela tendência ao declínio das atividades físicas, psíquicas e sociais (FREITAS et al., 2002, p.79), o que torna os idosos mais suscetíveis a acidentes. Entre eles está a queda, um acidente que, embora aconteça em todas as faixas etárias, possui consequências mais significativas para os idosos, podendo leva-lo à incapacidade e até mesmo ao óbito. A queda representa ainda um custo social, sobretudo quando ocorre diminuição da autonomia e da independência do idoso, ou quando ocorre a necessidade de internação do mesmo por longos períodos.

A queda é um evento causado por diversos fatores de risco, desde aqueles relacionados ao idoso (diminuição da força muscular, osteoporose alterações neurológicas, perda de equilíbrio, etc.) aos fatores relacionados ao ambiente (iluminação, arquitetura, disposição de móveis, presença de objetos escorregadios, entre outros). O risco de queda aumenta de acordo com a quantidade de fatores de risco presentes, e para prevenir quedas de forma eficaz, é necessária a adoção de estratégias de intervenção para múltiplos fatores (BVS, 2009).

Devido a um certo grau de fragilidade, alguns idosos vítimas de queda não conseguem levantar sozinhos após a ocorrência da mesma. Conforme será mostrado a seguir, a demora por ajuda pode agravar os danos ocasionados pela queda. Assim, pensou-se um dispositivo capaz de chamar auxílio, seja de um responsável, de um serviço de emergência, ou de ambos, imediatamente após uma queda. Embora o dispositivo não seja capaz de evitar a queda, o socorro imediato é um fator que pode ser considerado um diferencial no socorro à uma vítima de queda.

O protótipo desenvolvido é um “carpete inteligente”, dispositivo capaz de detectar a ocorrência de quedas e notificar o cuidador remotamente (por sms, web, etc.) O dispositivo é destinado principalmente a pessoas idosas que moram sozinhas, ou

cujos acompanhantes estão ausentes durante um período do dia; foi construído com materiais de baixo custo e utiliza como sistema de controle a plataforma Arduino, plataforma de prototipagem eletrônica simples.

2. OBJETIVO GERAL

Este projeto tem por objetivo o desenvolvimento de um carpete inteligente, que consiste em um sistema para detecção de quedas, pensado principalmente para pessoas idosas e construído com custos reduzidos. O protótipo, desenvolvido a partir da plataforma Arduino, deverá realizar a comunicação com o cuidador localizado remotamente, e/ou com o serviço de emergência através de SMS, para alertar uma possível queda.

3. FUNDAMENTAÇÃO TEÓRICA

Serão apresentados neste tópico conceitos fundamentais para este trabalho. Inicialmente, será realizada uma análise da situação dos idosos, que são público-alvo do projeto, seguido pela apresentação dos conceitos de robótica e domótica, além da plataforma Arduino.

3.1. Análise do público alvo

O aumento da longevidade da população de forma geral leva a um paradoxo, uma vez que ocorre simultaneamente o desejo de uma maior expectativa de vida e o temor às limitações inerentes à própria velhice. Um desafio proposto à sociedade atual está na obtenção de uma expectativa de vida cada vez maior, com melhor qualidade de vida (FREITAS et al., 2002, p.79).

A busca por autonomia tornou-se uma tendência entre os idosos na sociedade atual. Um estudo da Organização das Nações Unidas (ONU), que consistiu na análise de arranjos domiciliares dos idosos em 130 países, indicou que existe atualmente entre os idosos uma tendência à adoção de uma modalidade de vida mais independente (sozinho ou com o cônjuge), especialmente em países desenvolvidos. O mesmo

estudo mostra que aproximadamente uma em cada sete pessoas idosas vive sozinha. (UNITED NATIONS, 2005 apud. CAMARGOS et al., 2007, p.37-38).

Para Freitas et al. (2002, p. 762), a alteração na configuração demográfica atual levou a uma alteração epidemiológica, onde doenças degenerativas foram incluídas no perfil epidemiológico da população. Segundo Camargos et. al. (2007, p.37), o processo de envelhecimento da população em países como o Brasil ocorreu de forma demasiadamente rápida, sem que as condições socioeconômicas pudessem permitir que medidas destinadas a cobrir as necessidades desse público fossem efetivamente criadas.

O envelhecimento traz consigo diversas alterações fisiológicas no organismo, que se torna mais frágil, lento e com prejuízos quanto à coordenação motora. Tal quadro torna o indivíduo mais suscetível a acidentes, entre eles, as quedas. Estas são consideradas eventos bastante comuns na vida do idoso, ao mesmo tempo, porém, um dos mais devastadores. A estimativa é que para cada três indivíduos com mais de 65 anos ocorra uma queda, sendo que a mesma resulta em fratura ou necessidade de internação em um a cada 20 casos. (BVS, 2009).

Uma pesquisa realizada por Fabrício et al. (2004, p.95-96) em hospitais públicos da cidade de Ribeirão Preto (SP) aponta que 66% das quedas ocorreram no próprio lar do idoso, e que 54% das quedas apresentaram o ambiente inadequado como causa. A queda pode ocasionar fraturas e outras enfermidades, resultando em uma maior dependência do idoso em relação a seus familiares, internação, ou até mesmo sua morte. Segundo o mesmo autor, a queda pode trazer ainda, prejuízos psicológicos ao idoso, que após o episódio é submetido a um maior controle (seja por parte de parentes/cuidadores, ou de um autocontrole do próprio paciente) por medo de cair novamente e de sofrer as consequências que uma nova queda lhe traria.

Segundo Freitas et al. (2002, p.629), parte dos idosos que são vítimas de queda não conseguem, após a ocorrência da mesma, levantar-se sem auxílio. A tendência é que esta parcela seja composta por indivíduos mais velhos e frágeis, em comparação aos que conseguem levantar sem ajuda. O decúbito de longa duração representa aos incapazes de levantarem-se por conta própria, riscos de desidratação, pneumonias, úlceras de pressão e rabdomiólise (síndrome provocada por lesão muscular).

3.2. Robótica

A robótica é um campo de pesquisa multidisciplinar, que envolve áreas como a computação, engenharia elétrica e mecânica, física, matemática, entre outras. (PEREZ, 2005, p.3). No idioma Tcheco, a palavra *robota* significa servidão ou trabalhador escravo, sendo equivalente ao termo inglês *robot*. A palavra robótica, por sua vez, teve sua primeira aparição num livro de ficção científica escrito por Isaac Asimov em 1942, denominado *Runaround*.

Um robô pode ser definido como “uma máquina que percebe, planeja e atua sobre um ambiente.” (BEKEY, 2005 *apud*. PEREZ, 2005, p.4), enquanto “a Robótica é definida como a ligação inteligente entre a percepção e a ação.” (DE SOUZA PIO et al., 2006, p.1). Ambos os trechos contemplam três aspectos determinantes na Robótica: O sentir, o interpretar e o agir. Cada uma destas funções é desempenhada por um sistema específico. Resumidamente, um sistema robótico deve conter os elementos abaixo:

- Sistema de Entrada (Sensores);
- Sistema de Controle (Planejamento);
- Sistema de Saída (Atuadores).
- Sistema de Energia (Fonte de energia).

Os sistemas de entrada são responsáveis por fazer a leitura de informações específicas do ambiente onde está inserido, de forma analógica¹ ou digital ². Fazem parte desta etapa os diversos tipos de sensores (luz, temperatura, umidade, etc.).

Uma vez realizada a leitura das variáveis externas, cabe ao Sistema de Controle interpretá-las. A etapa de Controle ou Planejamento é constituída, na maioria dos casos, por Controladores Lógicos Programáveis (CLP's) ou microcontroladores, que são programados através de *softwares* específicos. Neste sistema ocorre, a partir das variáveis interpretadas, a tomada de decisões.

¹ Leitura analógica: Leitura na qual o valor lido é comparado com uma escala específica, nela assumindo determinado valor.

² Leitura digital: Modo de leitura que admite apenas dois valores lógicos.

A decisão tomada pela etapa de Planejamento é executada, por sua vez, pelos atuadores. Tais dispositivos correspondem à etapa de saída do sistema robótico, com a atuação deste sobre um ambiente específico. Ou seja, através deles ocorre a interação do sistema com o mundo exterior.

Um sistema de iluminação inteligente pode aqui ser utilizado como exemplo da integração entre sistemas. Um fotorresistor³ atua como sensor (entrada), enviando sinais analógicos para um microcontrolador. Este verificaria se o sinal recebido encontra-se numa faixa crítica (controle), e, em caso negativo, emitiria um sinal digital para um relé (atuador) acionar uma lâmpada.

A “inteligência” do sistema é pensada então a partir da relação entre sentir, interpretar e agir. Nesse sentido, a inteligência pode ser vista como a capacidade do sistema em sentir características específicas do ambiente, lidando com diferentes informações recebidas e escolher a ação mais aplicável para cada uma delas. A inteligência é essencial para um sistema robótico.

Trabalhar em Robótica significa estudar, projetar e implementar sistemas ou dispositivos que, com a utilização de percepção e de certo grau de “inteligência”, sejam úteis na realização de uma determinada tarefa, pré-definida ou não, que envolva interação física entre o sistema (ou dispositivo) e o meio onde a tarefa está sendo realizada. (DE SOUZA PIO et al., 2006, p.497).

3.3. Domótica

A domótica refere-se ao controle automatizado residencial, sendo conhecida popularmente como “casa inteligente”. O termo deriva das palavras domus (casa) e robótica (controle automatizado de algo) (ALVES; MOTA, 2003). Um sistema domótico permite a realização de atividades como controlar janelas e portões, acender ou apagar luzes, ligar equipamentos, e muitas outras, de forma automatizada, e/ou por um comando remoto do usuário.

³ Fotorresistor: componente eletrônico que varia sua resistência elétrica de acordo com a quantidade de luz que é incidida sobre o mesmo.

Um sistema de domótica eficaz é aquele que conta com a integração de diversos sistemas, assim como a possibilidade de expansão. Ele deve ainda ser capaz de interagir de forma harmônica com o usuário (ALVES; MOTA, 2003).

As primeiras aplicações da automação no Brasil ocorreram na década de 70, predominantemente na área industrial. De lá para os dias atuais, a automação tornou-se cada vez mais acessível e barata, tecnologias específicas para residências foram desenvolvidas, e havendo ainda a possibilidade do controle ou supervisão ser feito de qualquer lugar, utilizando dispositivos como celulares, tablets e smartphones (DA COSTA et al., sd, p.1).

As casas inteligentes podem ser úteis tanto para pessoas que buscam maior sofisticação e conforto quanto para aquelas que encontram dificuldades na execução de simples tarefas cotidianas, como idosos e pessoas com necessidades especiais. Da Costa et al. (sd, p.2) apresenta que cerca de 300 mil residências brasileiras possuem algum sistema automatizado, segundo dados da Associação Brasileira de Automação Residencial (Aureside). Este número tem forte tendência de crescimento, apesar de todos os entraves que o setor ainda possui, entre eles a adaptação dos sistemas ao perfil do cliente e nos custos elevados da automação.

3.4. A plataforma Arduino

O Projeto Arduino foi criado em 2005 na Itália, visando principalmente *hobbistas*, designers e artistas. “Arduino é uma plataforma eletrônica de código aberto baseado em hardware e software de fácil utilização. É destinado para qualquer pessoa desenvolver projetos interativos.” (Arduino, 2015). A documentação do Arduino encontra-se disponível sob a licença Creative Commons Attribution-ShareAlike 2.5, que permite a cópia, redistribuição e alteração do material, inclusive para fins comerciais, desde que mantida a licença original.⁴

A facilidade na utilização da plataforma permite a criação de projetos interativos por pessoas sem especialização em eletrônica e/ou programação.

⁴<https://creativecommons.org/licenses/by-sa/2.5/br/>

A plataforma Arduino é basicamente composta por um *hardware* (placa Arduino) e um *software* (IDE).

3.4.1. A placa Arduino

A parte física do Arduino é baseada em microcontroladores da família Atmega (fabricados pela empresa ATMEL), sendo o Atmega328P o mais utilizado. O Atmega é um microprocessador que conta com entradas para sensores e saídas; deve ser programado usando uma linguagem de programação específica para o Arduino (baseada na linguagem C/C++) e o ambiente de desenvolvimento Arduino. (BORTHOLOTTO; BORGES, 2009, p.2). É possível comprar a placa já pronta, ou adquirir o Atmega328 e outros componentes e montar sua própria placa. Isso possibilita o surgimento de diversos modelos de placas diferentes das versões oficiais.

As placas oficiais também possuem grande diversidade, conforme pode ser visto na Imagem 1. Atualmente é possível encontrar modelos desde os mais poderosos e robustos (como a placa Arduino Esplora ou Mega) a modelos que, segundo o fabricante, podem ser até mesmo costurados na roupa (Arduino LilyPad).

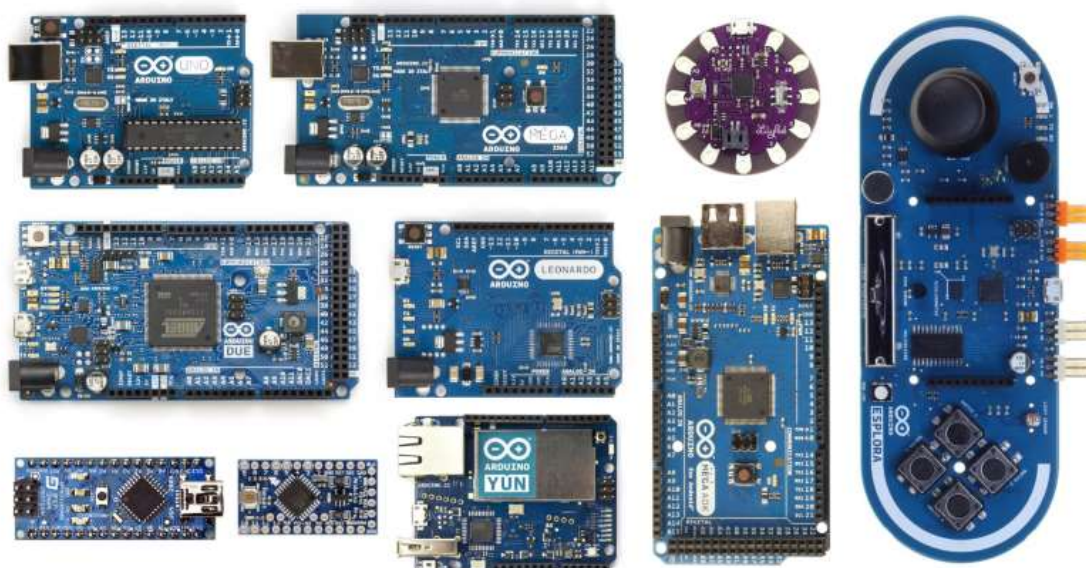


Imagem 1: Diferentes modelos de Arduino.

Fonte: Blog FilipeFlop⁵.

O modelo Uno é o mais utilizado em todo mundo, podendo ser adquirido no mercado brasileiro por menos de R\$ 100,00. É baseado no microcontrolador Atmega328. Possui 6 entradas analógicas (pinos A0 - A5) e 14 pinos digitais de entrada/saída (0-13). Entre os pinos digitais, dois deles são utilizados para recepção e transmissão de dados em comunicação serial (pinos 0 e 1, respectivamente), seis podem ser utilizados como saída PWM (pinos 3, 5, 6, 9, 10 e 11). A placa funciona com tensão operacional de 5V DC (corrente contínua), podendo ser alimentada diretamente via USB ou por fonte DC externa com tensão entre 7 e 12 V através do conector de alimentação ou dos pinos Vin e GND. A placa possui ainda um botão para reinício, e uma interface de comunicação USB (Imagem 2).

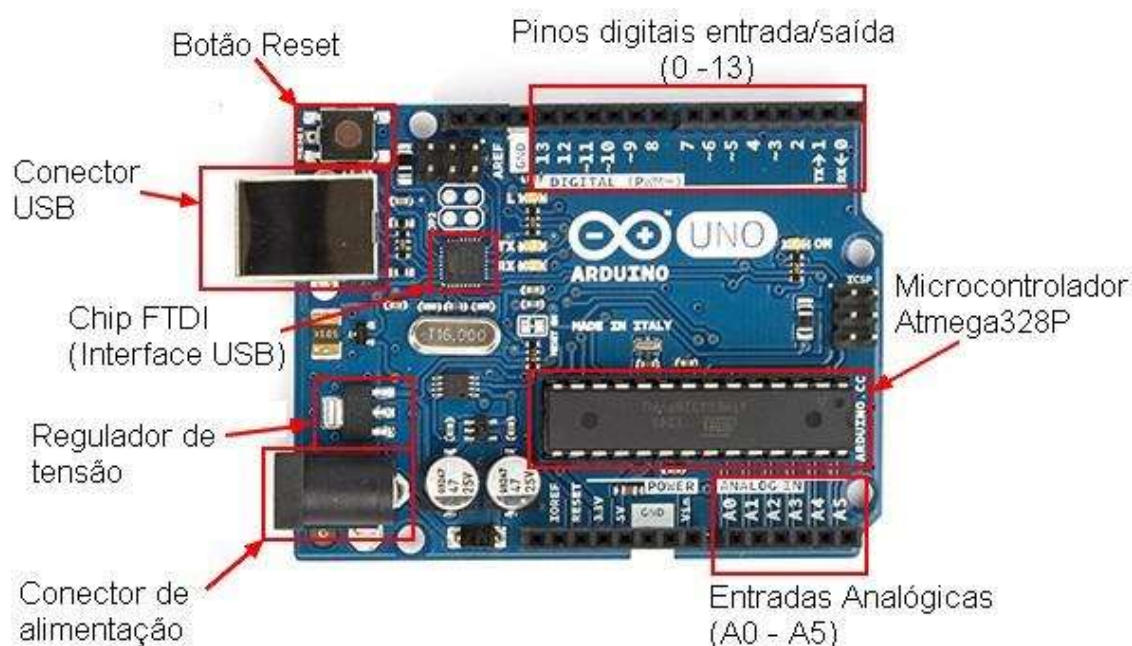


Imagem 2: Detalhes da placa Arduino Uno.

Fonte: Página oficial do Arduino (adaptado)⁶.

⁵ Disponível em <<http://blog.filipeflop.com/arduino/o-que-e-arduino.html>>. Acesso em 22. abr. 2015.

⁶ Disponível em <http://arduino.cc/en/uploads/Main/ArduinoUno_R3_Front.jpg>. Acesso em 18. Fev. 2015.

3.4.2. O IDE Arduino

No Ambiente de Desenvolvimento Integrado ou IDE (do inglês *Integrated Development Environment*) do Arduino são escritos os programas (denominados de *sketches*) responsáveis pela funcionalidade da placa. O IDE do Arduino (Imagem 3) foi desenvolvido em linguagem JAVA, com base em softwares livres como o projeto Processing e a biblioteca AVR-gcc , apropriada para microcontroladores da família AVR. A linguagem de programação utilizada é relativamente simples e de fácil compreensão, sendo baseada no projeto Wiring (possui estrutura semelhante à linguagem C/C++); encontrando-se disponível para plataformas Windows e Linux (ALVES, 2013, p.167).

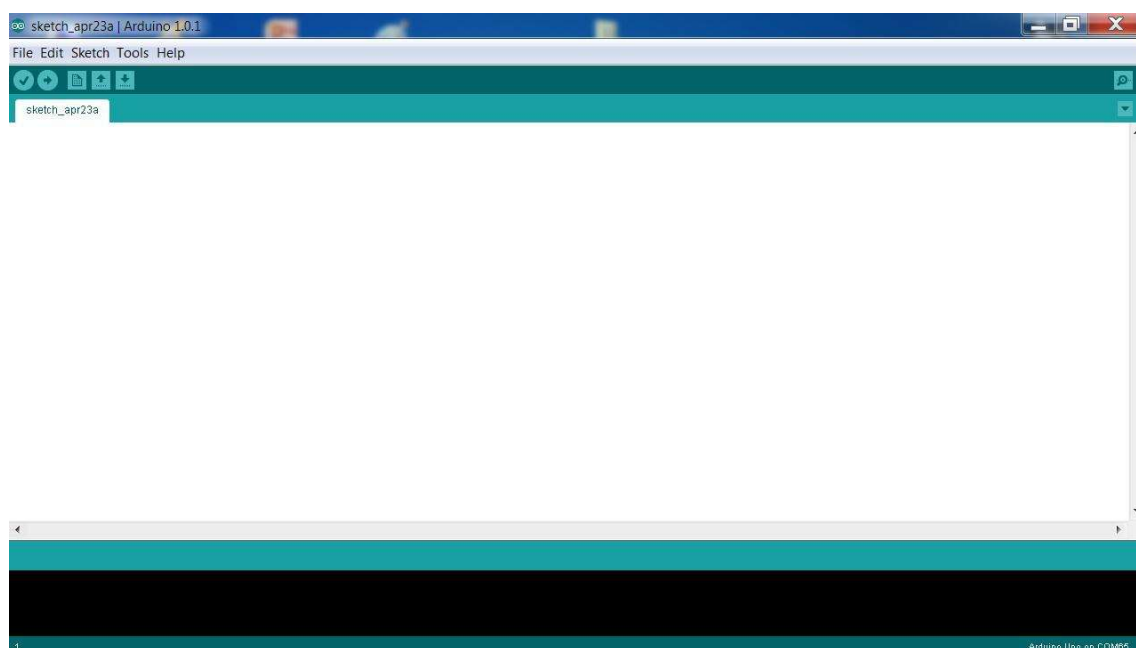


Imagem 3: *Print screen* do IDE Arduino, versão 1.05, no sistema operacional Windows 7.

4. METODOLOGIA

O presente trabalho foi desenvolvido para apresentação na I Semana de Arte, Cultura, Ciência, Tecnologia e Inovação no Instituto Federal da Bahia, Campus

Simões Filho, dentro do tema “Robótica aplicada ao auxílio de pessoas com necessidades especiais”, orientado pelo prof. Paulo Vicente.

O desenvolvimento deste trabalho deu-se de acordo com as seguintes etapas, que foram propostas de acordo com o cronograma do evento:

- Estudo da plataforma Arduino;
- Análise da relação Proteção X Autonomia;
- Busca por dispositivos semelhantes no mercado;
- Projeto;
- Orçamento;
- Montagem e testes.

4.1. Estudo da plataforma Arduino

Para a concepção do projeto, houve o estudo prévio da plataforma Arduino, resultando no conhecimento de seu funcionamento e possibilidades de uso. O orientador utilizou o livro *Arduino Básico*, de Michael McRoberts para o aprendizado e familiarização com a plataforma Arduino. O livro conta com diversos projetos, dos mais simples aos mais complexos, a serem desenvolvidos pelo estudante. Os exercícios realizados incluíam desde efeitos com LEDs a sonorizadores e utilização de displays de cristal líquido.

4.2. Análise da relação Proteção X Autonomia

Após o estudo apresentado anteriormente a respeito das condições do público-alvo (pessoas idosas), foi observada a preocupação existente em promover a proteção do idoso sem resultar em perda em sua autonomia, conforme Freitas et al. (2002, p.634) diz a seguir.

(...) A independência e a qualidade de vida do paciente idoso deve ser o contrapeso na balança proteção X independência para, como Downtown bem coloca, a vida possa valer a pena ser vivida, mesmo que com algum risco inevitável. (Downtown, 1998 apud Freitas et. al., 2002).

Partindo dessa ideia, a independência do idoso constituiu-se um fator a ser levado em conta na idealização deste projeto, uma vez que um dos prejuízos ocorridos após uma queda é o maior controle dos familiares e cuidadores sobre o idoso, e o próprio medo da vítima de cair novamente.

Conforme será visto no tópico seguinte, dispositivos com funções semelhantes às do protótipo apresentado neste trabalho já existem no mercado. A existência de um dispositivo que possa auxiliar no atendimento rápido após uma queda reduz o medo de queda entre idosos, fornecendo reforço psicológico ao paciente e familiares (Freitas et al., 2002, p. 633).

4.3. Busca por dispositivos semelhantes no mercado

Segundo Freitas et. al. (2002, p. 633), em alguns países como a França existem sistemas de alarmes (denominados *Téléalarme*), que são ligados a centrais de emergência e são acionados a partir de um dispositivo normalmente posicionado no pescoço do idoso, em forma de colar.

Durante a busca por dispositivos com propósitos semelhantes, observou-se que os equipamentos encontrados eram semelhantes ao descrito por Freitas. Tais dispositivos consistem num botão colocado no corpo do idoso, sob a forma de colar, pulseira, chaveiro, etc., e acionado pelo próprio em situações de emergência. Na Imagem 4 temos um modelo, que pode ser usado tanto como bracelete quanto como pulseira.



Imagem 4: Modelo de dispositivo de emergência ("botão do pânico") para idosos, que pode ser usado tanto como bracelete quanto como pulseira.

Fonte: Alibaba.com⁷.

Caso a vítima esteja desacordada após a queda, entretanto, não conseguirá acionar o botão. O mesmo ocorre com o idoso que caia de forma que o impeça o acionamento do botão, ou ainda esqueça de acioná-lo (idoso com Alzheimer por exemplo). Além disso, o fato de ter um equipamento preso ao corpo pode representar para alguns indivíduos uma quebra em sua liberdade e autonomia.

Devido a estes inconvenientes, foi pensado o desenvolvimento de um equipamento capaz de detectar ocorrência de queda, que não seja preso ao corpo do indivíduo monitorado e cujo acionamento não dependa do idoso.

4.4. Projeto

Sabendo que a maioria das quedas acontece no próprio lar do idoso, e que a demora na chegada do socorro pode agravar possíveis traumas provocados na vítima, julgou-se necessário o desenvolvimento de um dispositivo capaz de detectar

⁷ Disponível em <<http://portuguese.alibaba.com/product-gs/gsm-panic-button-elderly-60037724622.html>> . Acessado em 23. Abr. 2015.

a ocorrência da queda e solicitar ajuda imediatamente, interferindo minimamente na autonomia do idoso. Considerou-se também que muitos idosos podem apresentar dificuldades para levantar e pedir socorro após a queda. Assim, surgiu a ideia da criação de um carpete inteligente, que fosse capaz de detectar a queda e permanência do idoso no chão e comunicar o evento a uma pessoa distante ou até mesmo ao um serviço de emergência.

O carpete, teoricamente, ocuparia todo o ambiente da residência, e seria fixo ao chão. Vale ressaltar que o dispositivo em questão trata-se de protótipo, construído em tamanho reduzido para maior simplicidade na montagem e manuseio.

4.5. Descrição dos sistemas

O protótipo em questão utiliza os princípios da domótica e robótica, e, por ser um sistema robótico, pode ser dividido em três sistemas básicos: O de entrada, de controle e de saída.

4.5.1. Sistemas de entrada

Conforme apresentado anteriormente, o sistema de entrada corresponde ao que realiza a detecção das variáveis do ambiente. A fim de escolher o melhor método de detecção a ser empregado no projeto, foram realizados testes com dois tipos de componentes eletrônicos.

O primeiro a ser testado foi a cápsula piezoelétrica, conhecida como disco piezo, ou simplesmente, piezo (Imagem 5). O componente é constituído por um cristal que gera tensão elétrica através da variação de pressão em sua superfície. No mercado, são encontradas em dimensões variadas, com custos relativamente baixos. Sua razoável sensibilidade e dimensões reduzidas (especialmente sua espessura) mostraram-se como grandes vantagens para sua utilização.



Imagem 5: Cápsula piezoelétrica.

Fonte: Baú da eletrônica⁸.

Foram adquiridas 25 cápsulas piezoelétricas⁹ de 15mm para testes, com custo unitário de R\$ 0,69. Foram realizados testes com o Arduino, utilizando portas analógicas para efetuar a leitura analógica do sinal gerado por três piezos (Imagem 6). No caso dos três piezos apresentarem sinal acima de um limite preestabelecido, assemelhando-se a uma queda sobre várias unidades piezoelétricas, ocorria a emissão de um alerta sonoro (o código utilizados no teste encontra-se no Anexo 1).

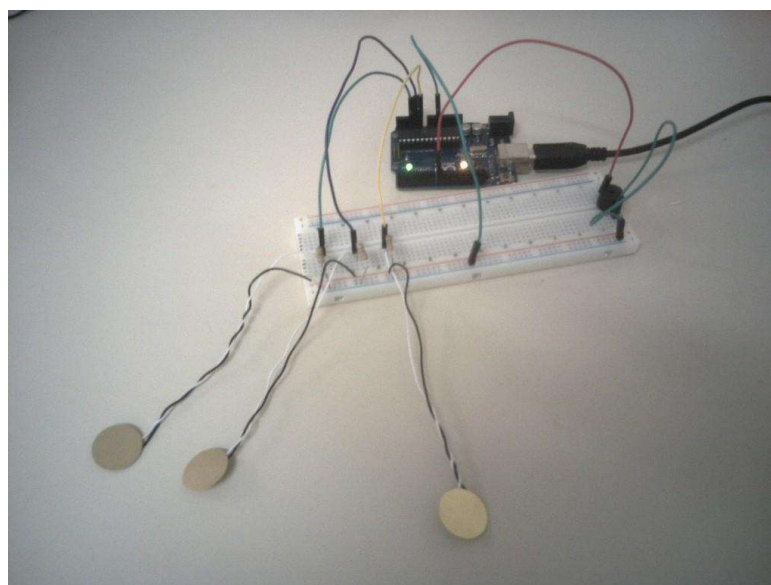


Imagem 6: Teste com piezos.

⁸ Disponível em < <https://www.baudaeletronica.com.br/capsula-piezoelétrica-15mm.html>>. Acessado em 24.abr.2015.

⁹ Datasheet disponível em <<https://www.sparkfun.com/datasheets/Sensors/Flex/p37e.pdf>>. Acessado em 24. abr. 2015.

Fonte: Elaborado pelo autor.

Imaginou-se que, caso uma força constante fosse aplicada sobre ele (como a exercida por uma pessoa imóvel caída sobre o solo) um sinal de tensão continuaria presente na saída do piezo. Contudo, descobriu-se que o piezo era capaz de detectar apenas variações de pressão e não uma pressão contínua, sendo assim inadequado ao protótipo. Seria possível utilizar esse componente para detectar o impacto de uma queda, mas não se após tal impacto, a pessoa continua ou não sobre o chão; e nem caso a queda se desse de forma não brusca (a pessoa cair aos poucos). Além disso, o comportamento do piezo mostrou-se imprevisível, com grandes possibilidades de gerar alarmes falsos, ou quedas não detectadas.

Após o teste com os piezos, outro componente, denominado sensor de força resistivo (FSR)¹⁰ foi cogitado como sistema de entrada (Imagem 7). Este componente totalmente compatível com Arduino, varia sua resistência elétrica de acordo com a força aplicada sobre ele. Acredita-se que o FSR seria, entre os componentes pensados, o mais eficiente para ser utilizado no projeto, uma vez que seria possível verificar não apenas se uma força foi ou não aplicada, mas também a sua intensidade. Algumas desvantagens, no entanto, inviabilizaram a realização de testes com o mesmo, e, conseqüentemente, sua utilização no projeto; entre elas, o custo elevado do componente (o custo unitário varia entre aproximadamente R\$ 46,00 e R\$ 196,00), a pequena área sensível à força (4 cm x 4 cm) e limitações quanto aos valores limites de medição, de no máximo, 10 kg.

¹⁰ Datasheet disponível em <<https://www.sparkfun.com/datasheets/Sensors/Pressure/fsrguide.pdf>>. Acessado em 05. jun. 2015.

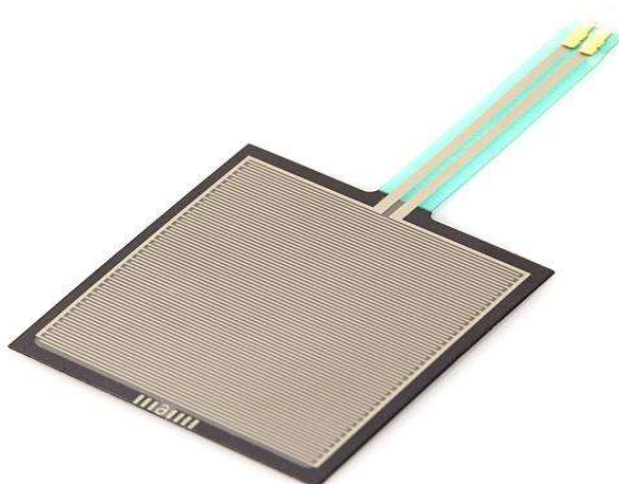


Imagem 7: Sensor de Força Resistivo (FSR) quadrado.

Fonte: Sparkfun¹¹.

Outro componente testado foi a chave tátil, também chamada de *push-button*, sendo, após os testes, empregada no protótipo. O dispositivo é uma chave que, enquanto pressionada, efetua uma ligação entre dois terminais, fechando um circuito.

Os *push-buttons* são componentes simples, baratos e facilmente encontrados, além de serem relativamente confiáveis. Internamente, a chave tátil é composta por duas placas de metal. Numa chave normalmente aberta, através de uma força aplicada em um pino, uma das placas é empurrada ao encontro da outra, fechando o circuito e ocorrendo também a deformação de uma mola. Quando a força cessa, a mola retorna ao seu estado original, afastando o contato móvel, abrindo o circuito (Imagem 8).

¹¹ Disponível em <<https://www.sparkfun.com/products/9376>>. Acessado em 22. abr. 2015.

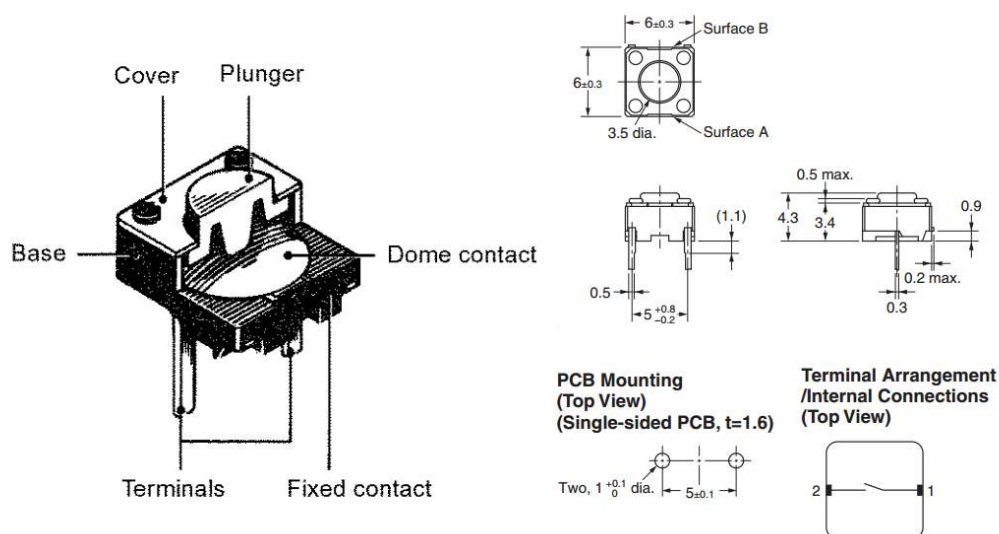


Imagem 8: Corte parcial, vistas e esquema de funcionamento de chave táctil (*push-button*).

Fonte: Omron¹²

O sistema de entrada deste projeto consiste em vários botões distribuídos em uma superfície, de forma que a decisão do controle será tomada a partir da quantidade de botões acionadas. Foi observado que a leitura individual dos *pushbuttons* seria inviável, devido à quantidade de botões ser maior que o número de portas digitais disponíveis no Arduino, e também por tornar o código bastante complexo. Optou-se assim pela distribuição dos botões em linhas. Cada linha, composta por botões ligados em paralelo é lida pelo Arduino como um único botão.

Os botões foram agrupados em quatro linhas, com três botões cada uma. Cada linha possui uma saída de sinal representada pelas letras A, B, C e D, cada uma delas ligada a uma porta digital do Arduino. O conjunto é alimentado por V_{cc} , uma tensão positiva de 5 Volts. O agrupamento em paralelo dos botões em cada linha, conforme mostra a Imagem 9, permite que quando um ou mais botões de determinada linha for(em) acionado(s), o contato entre a alimentação V_{cc} e a saída correspondente seja fechado. A saída, então, passa para o estado lógico alto, ou HIGH.

¹² Disponível em <<http://www.omron.com/ecb/products/pdf/en-b3f.pdf>>. Acesso em 11.jul.2015.

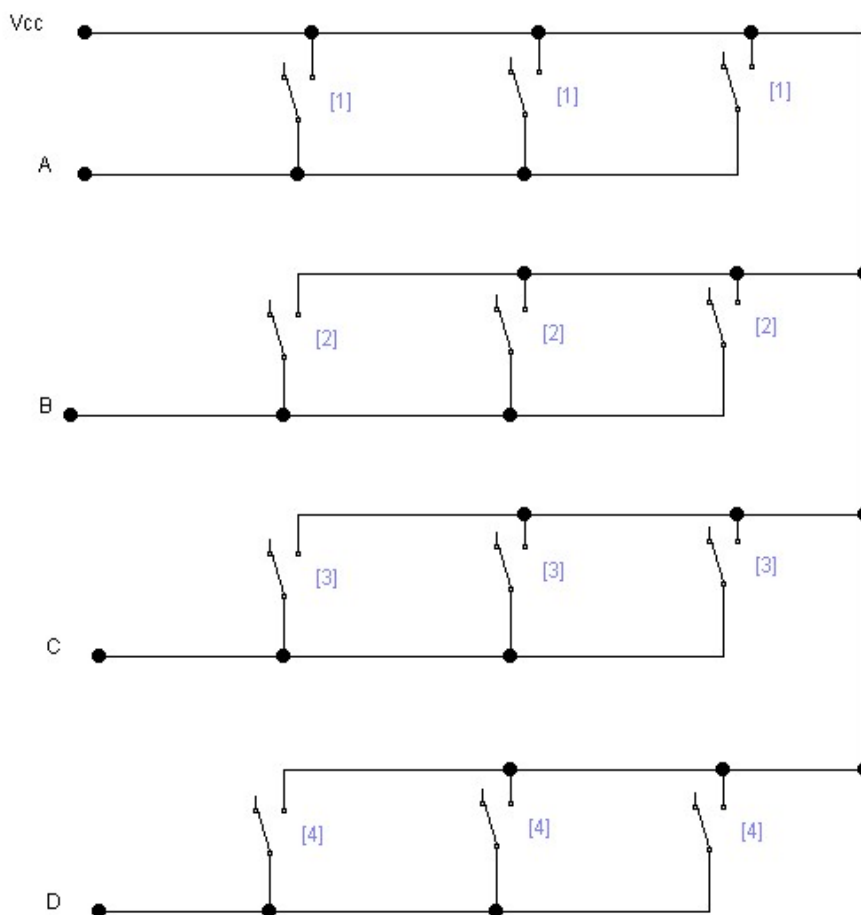
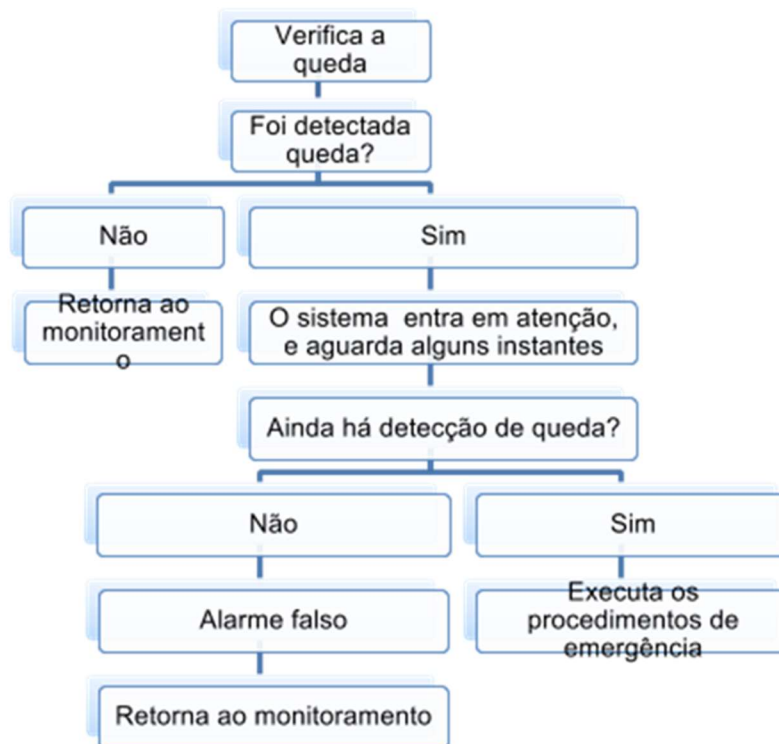


Imagem 9: Esquema elétrico de ligação dos pushbuttons.

Fonte: Elaborado pelo autor.

4.5.2. Sistema de controle

A etapa de controle é feita através da placa Arduino, que, com base na leitura do status das saídas do dispositivo, irá interpretar se houve queda ou não. Antes do desenvolvimento do *sketch* no IDE Arduino, foi criado um diagrama contendo a lógica utilizada no programa, representado logo a seguir no Esquema 1.



Esquema 1: Representação esquemática da programação utilizada.

Fonte: Elaborado pelo autor.

O sistema verificará constantemente se ocorreu uma queda, e em caso positivo, entra em estado de atenção. Após alguns instantes (foi utilizado um tempo de dois segundos, em caráter didático), uma nova verificação é feita, e, se a queda ainda estiver sendo detectada, são executados os procedimentos de emergência. Caso contrário, trata-se de um alarme falso, e o sistema retorna ao seu estado normal.

4.5.3. Sistema de saída

O sistema de saída é compreendido por todos os dispositivos que sofrerão atuação, de acordo com a interpretação do controle da leitura do ambiente. Tais componentes podem funcionar como indicadores locais de status (leds, buzzer, display LCD) ou ainda para estabelecer a comunicação com uma pessoa remota, como o próprio shield GSM.

4.6. Orçamento

Após a etapa de projeto, foi feita uma lista contendo os materiais a serem utilizados no mesmo, bem como os respectivos preços de mercado. Para que os custos pudessem ser reduzidos, pensou-se na utilização de materiais de baixo custo, de forma a tornar viável a montagem do protótipo e, futuramente, uma possível montagem industrial. O orçamento do projeto encontra-se disponível no Anexo 2.

4.7. Montagem

4.7.1. Montagem do protótipo

A base do protótipo foi desenhada no software AutoCAD 2013, da Autodesk, para em seguida ser reproduzida e montada em uma placa de papelão, de dimensões 45 x 90 cm. Sobre a base, foram traçadas retas, de forma a orientar a posição dos botões.

Pequenos furos foram feitos no papelão, de forma a fixar os botões. Os botões foram agrupados em 4 linhas, cada uma contendo 3 botões cada. A distância entre os botões é de 11,25cm, e entre cada linha, 22,5 cm. Tais distâncias foram estabelecidas em caráter experimental. Sobre cada botão foi colocado um pequeno pedaço retangular de papelão, de forma a aumentar a superfície de contato de cada botão.

Na parte inferior da base, terminais dos botões foram interligados, através de fios finos e flexíveis, obtendo-se a disposição elétrica mostrada anteriormente. As conexões foram protegidas mecanicamente utilizando-se cola quente e material emborrachado (EVA). O acabamento da parte superior foi feito em material próprio para carpete.

O processo de montagem foi registrado através de fotografias, podendo ser acompanhado pela sequência de imagens a seguir, da Imagem 10 até a Imagem 15.

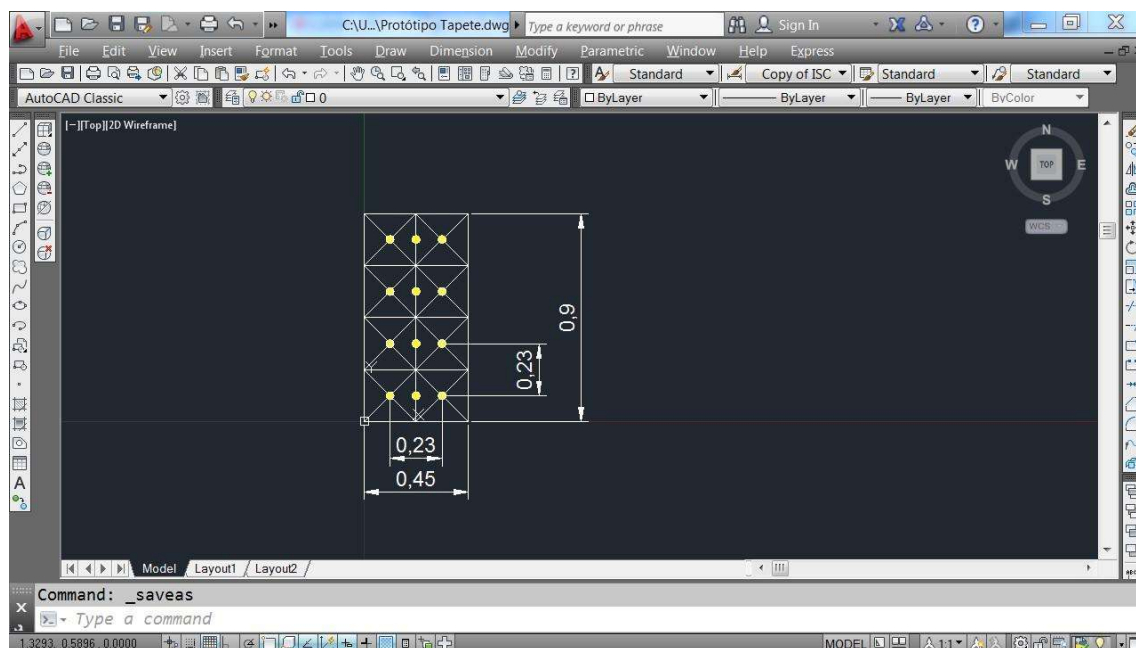


Imagem 10: *Print screen* do software AutoCAD 2013, contendo o desenho da base do protótipo.



Imagem 11: Base do protótipo com as marcações em papelão.

Fonte: Do Autor.

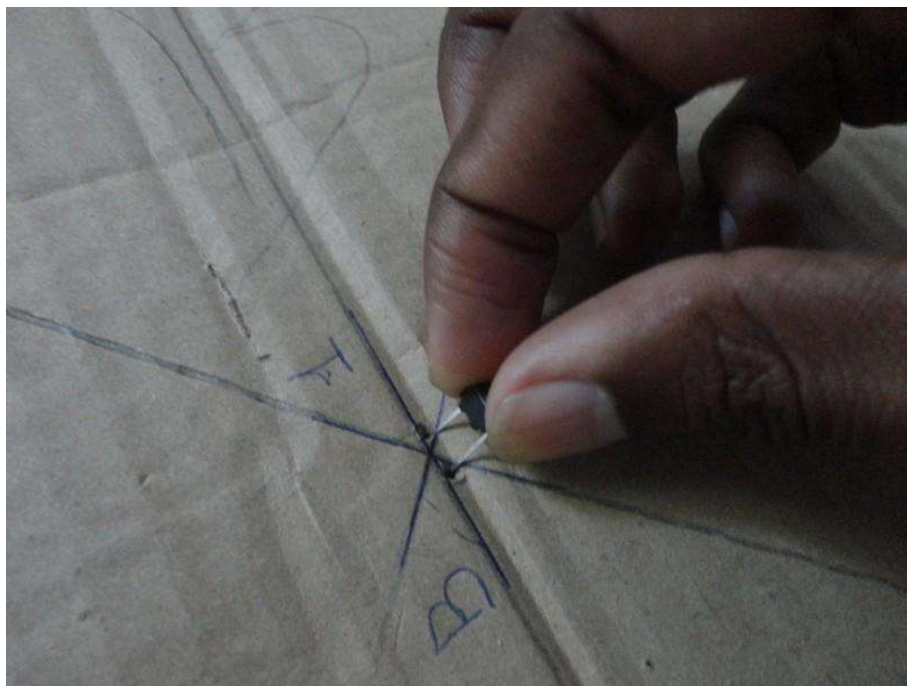


Imagem 12: Fixação de um botão na base.

Fonte: Elaborada pelo autor.

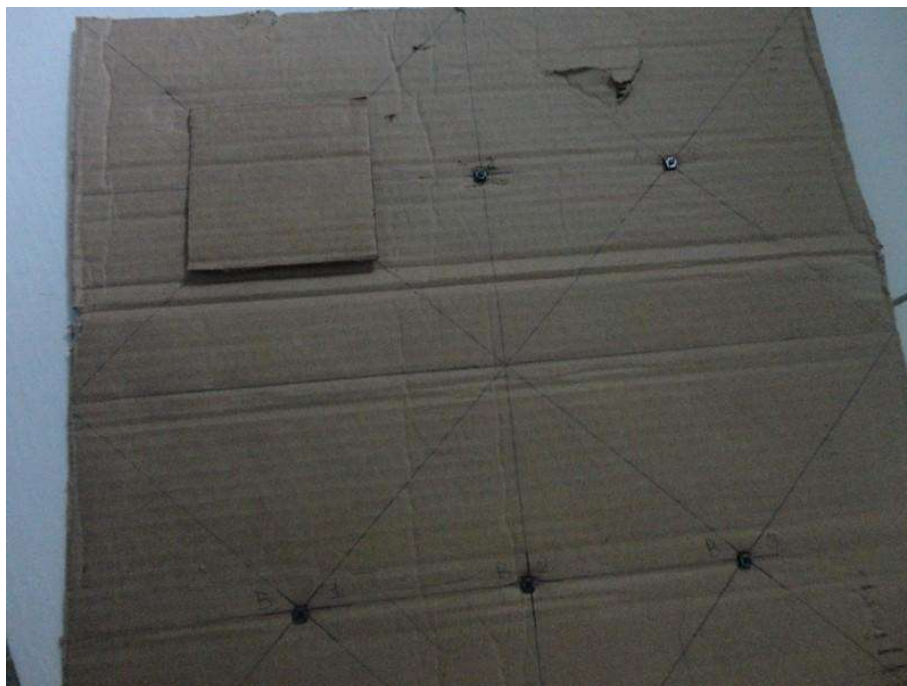


Imagem 13: Fixação de pedaço retangular de papelão.

Fonte: Elaborada pelo autor.



Imagem 14: Aspecto da parte superior do dispositivo inacabado.

Fonte: Elaborada pelo autor.



Imagem 15: Aspecto final do protótipo, já com acabamento em carpete na parte superior.

Fonte: Elaborada pelo autor.

4.7.2. Montagem na protoboard

No projeto foram utilizados dois Arduinos, sendo que o Arduino Uno é responsável somente para controlar o GSM Shield, e o Mini, utilizado para a leitura de dados e indicadores locais (LEDs, displays, alertas sonoros, etc.). Após o encaixe do *shield* GSM no Arduino Uno, a utilização das portas, mesmo as não utilizadas pelo shield, foi impossibilitada, pois todas as portas foram cobertas pelo *shield*. Além disso, outro fator determinante na utilização de dois Arduinos é a quantidade de memória a ser utilizada, que pode ser comprometida com a utilização de um único código de grande extensão.

O Arduino Mini e todos os componentes como LEDs, display, buzzer e resistores foram encaixados em um protoboard. Para conectar o carpete ao Arduino, foram utilizados os fios de um cabo de internet, de aproximadamente 2,0m de comprimento. O esquema de ligação dos componentes pode ser visto na Imagem 16. Os pinos do Arduino Mini foram ligados, conforme indicado na Tabela 1.

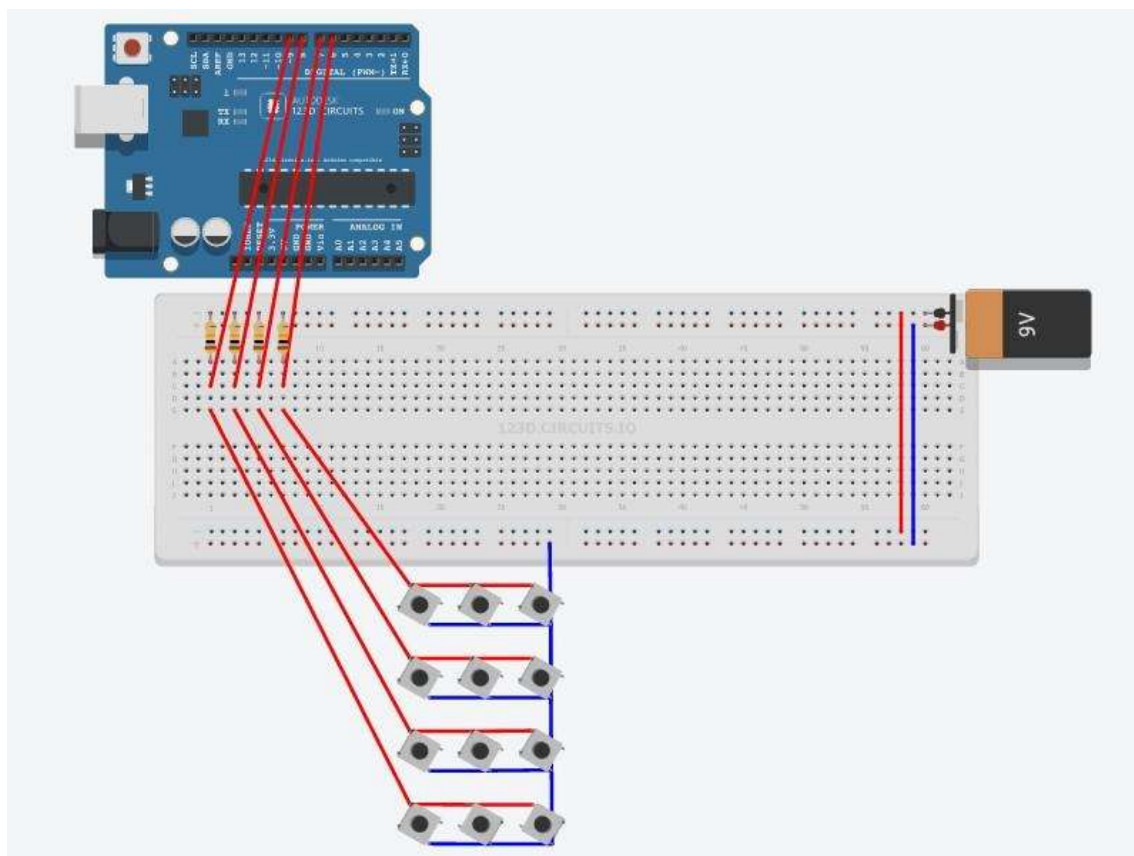


Imagem 16: Esquema da ligação de componentes na protoboard.

Fonte: Elaborada pelo autor.

Tabela 1: Lista de portas utilizadas no Arduino Mini.

Pinos	Componente(s) conectado(s)
3	Buzzer;
6, 7, 8 e 9	Ligar cada um a uma saída do carpete;
10	LED verde;
11	LED amarelo;
12	LED vermelho;
A0	mySerial RX;
A1	mySerial TX;

Fonte: Elaborada pelo autor.

O Arduino Uno comunica-se com o Arduino Mini via serial, através dos pinos 0 e 1 do Uno (TX e RX) e os pinos A0 e A1 do Mini (mySerial RX e mySerial TX). Os pinos GND de ambos os Arduinos foram interligados, para que todos tivessem o mesmo referencial elétrico. Os códigos utilizados em ambas as placas estão disponíveis, respectivamente, nos Anexos 3 e 4.

Para o envio de mensagens SMS, foi utilizado o Shield GSM GPRS ECom Sim900¹³ (Imagem 17). O shield completamente compatível com Arduino, baseia-se no módulo wireless SIM900 Quad-Band GSM/GPRS, capaz de realizar serviços como SMS, Ligações de voz, Dados via Internet e Fax em Quad-Band (850/900/1800/1900MHz).



Imagem 17: GSM GPRS Shield ECom Sim900.

Fonte: Filipeflop¹⁴.

O shield é diretamente conectado sobre a placa Arduino. Funciona com uma tensão interna de 5 V, porém, apresenta picos de potência durante seu funcionamento normal, que resulta em quedas de tensão. Como uma tensão abaixo da nominal

¹³ Datasheet disponível em <http://www.electfreaks.com/store/download/product/EFcom/EFcom_Datasheet.pdf>. Acessado em 21.abr.2015

¹⁴ Disponível em <<http://www.filipeflop.com/pd-77833-gsm-gprs-shield-para-arduino-efcom-sim900-antena.html>>. Acessado em 21.Abr. 2015.

pode prejudicar o desempenho do shield e do Arduino, o *datasheet* recomenda que o Arduino conectado ao *shield* seja alimentado por uma fonte que forneça tensão entre 6 e 12 Volts, e corrente mínima de 1A.

5. RESULTADOS ALCANÇADOS

Após a montagem, o dispositivo foi submetido a alguns testes de funcionamento. Nestes testes, foram simuladas quedas em posições diversas (de costas, de lado e de bruços), e foram avaliados critérios como o índice de detecção de quedas em cada uma destas posições, bem como se em todas as detecções o envio de SMS foi bem sucedido. Os testes obtiveram resultados bastante satisfatórios, apresentando funcionamento adequado na maioria dos testes realizados, ocorrendo, no entanto, algumas falhas que serão detalhadas no próximo tópico.

Caso uma pessoa caia deitada sobre o protótipo, todas as linhas serão acionadas simultaneamente, e todas as saídas ficarão no estado lógico HIGH. O sistema de controle fará a leitura da informação, e executará os procedimentos de emergência.

Localmente é possível acompanhar as variações de status do dispositivo através dos indicadores. A Tabela 2 informa o que é mostrado em cada um deles, seguindo cada uma das variações de status.

Tabela 2: Estados dos indicadores do Carpete Inteligente a partir da variação de status.

Status do carpete	Status dos componentes			
	LED	Buzzer	Display	SMS
Normal	Verde	-	Monitoramento Status: Normal	-
Atenção	Amarelo	-	Atenção! Possível queda!	-
Alerta	Vermelho	Sinal sonoro (440 Hz)	Alerta! Queda detectada!	Alerta! Queda detectada! [teste com o Carpete Inteligente]

Fonte: Elaborada pelo autor.

6. DIFICULDADES ENCONTRADAS

Durante os testes, o protótipo do Carpete Inteligente apresentou um desempenho satisfatório. No entanto, alguns problemas foram observados durante e após o desenvolvimento do trabalho.

6.1. Dificuldade de aquisição de componentes

A maior barreira logística encontrada foi a aquisição do shield GSM, utilizado no envio de SMS. O dispositivo constituiu o item de valor mais elevado no orçamento do projeto. O componente foi então adquirido pelo orientador, que negociará o reembolso por parte da instituição, condicionado à disponibilização do componente à mesma para futuras pesquisas, após a conclusão deste trabalho.

6.2. Falhas de funcionamento

O principal problema encontrado foi a falha que pode ocorrer na detecção da queda em posições específicas de queda (especialmente quando o indivíduo cai de costas ou de lado). O problema é causado pelo não acionamento de alguma linha de *pushbuttons*, devido à curvatura natural do próprio corpo humano. O problema pode ser resolvido aumentando o número de botões por área, o tamanho do dispositivo, ou ainda tornando-o mais anatômico.

Outros problemas detectados foram o possível alarme falso de queda, caso duas ou mais pessoas estejam sobre o dispositivo, e ainda a falta de detecção de queda nas duas direções do carpete. Tais falhas podem ser resolvidas alterando o arranjo dos botões e a forma de leitura dos mesmos.

7. POSSÍVEIS MELHORIAS

Apesar do bom desempenho do protótipo com o *pushbutton*, é importante lembrar que este componente não foi projetado para receber uma grande força, como a

aplicada durante uma pisada, possui pequena vida útil, produz um considerável ruído (que pode ser incômodo, principalmente à noite) e aumenta a espessura do dispositivo, o que pode representar um problema para fixação no piso e risco de acidente para os idosos.

A solução mais adequada seria a utilização de sensores de pressão/força, mas algumas alternativas podem ser bem adaptadas a este propósito. Uma das soluções encontradas foi empregar na fabricação de um carpete em uma estrutura semelhante a um teclado de membrana para substituir os *pushbuttons*. O princípio de funcionamento dos teclados de membrana poderia ser adaptado especialmente para esse fim, possibilitando a correção das adversidades citadas acima, mantendo o princípio de funcionamento dos botões.

Visto que o uso de carpete apresenta limitações (entre as quais, a sua aplicação em cômodos molhados, como cozinha e banheiro), o dispositivo apresentado poderia ser adaptado a outras formas de superfície (por exemplo, placas impermeáveis coladas ao piso).

8. CONCLUSÃO

O trabalho desenvolvido teve como propósitos gerais a aquisição de conhecimentos sobre robótica e a plataforma Arduino e sua aplicação no auxílio de pessoas idosas. Estes propósitos foram atendidos, embora algumas alterações ainda fazem-se necessárias neste projeto para garantir um melhor desempenho, segurança e confiabilidade,

Um desafio futuro seria a utilização de estruturas mais adequadas, podendo tornar o dispositivo mais barato e eficaz, além de explorar outras possibilidades de uso.

9. REFERÊNCIAS

ALVES, José Augusto; MOTA, José. Casas inteligentes. Centro Atlântico, 2003.

ALVES, Rafael Machado et al. Uso do Hardware Livre Arduino em Ambientes de Ensino-aprendizagem. Jornada de Atualização em Informática na Educação, v. 1, n. 1, p. 162-187, 2013.

ARDUINO. Disponível em <<http://www.arduino.cc>>. Acessado em 23. Abr. 2015.

BIBLIOTECA VIRTUAL EM SAÚDE (BVS) - MINISTÉRIO DA SAÚDE. Queda de Idosos. Disponível em <http://bvsmis.saude.gov.br/bvs/dicas/184queda_idosos.html>. Acessado em 11. Jul. 2015.

BORTHOLOTTO, Julio Cesar; BORGES, M. A. Integração de dispositivos robóticos a sistemas de apoio ao aprendizado utilizando a plataforma Arduino. LIAG–Laboratório de Informática, Aprendizagem e Gestão. UNICAMP. Campinas, 2009.

CAMARGOS Mirela Castro Santos, MACHADO Carla Jorge, RODRIGUES Roberto do Nascimento. A relação entre renda e morar sozinho para idosos paulistanos: 2000. Rev Bras Estud Popul, v. 24, n. 1, p. 37-51, 2007.

DA COSTA, Alison França Queiroz et al. Estudo da Automação Residencial.

DE SOUZA PIO, José Luiz et al.. A robótica móvel como instrumento de apoio à aprendizagem de computação. In: Anais do Simpósio Brasileiro de Informática na Educação. 2006. p. 497-506.

FABRÍCIO, Suzele et al.. Causas e consequências de quedas de idosos atendidos em hospital público. Rev Saúde Pública, v. 38, n. 1, p. 93-9, 2004.

FREITAS, Elizabete Viana et al. Tratado de geriatria e gerontologia. Rio de Janeiro: Guanabara Koogan, 2002.

INSTITUTO BRASILEIRO DE GEOGRAFIA E ESTATÍSTICA (IBGE). Síntese dos Indicadores Sociais: Uma análise das condições de Vida da População Brasileira (2010). Disponível em <<http://biblioteca.ibge.gov.br/visualizacao/livros/liv45700.pdf>>. Acessado em 17. Jul. 2015.

MCROBERTS, Michael. Arduino básico. Tradução Rafael Zanolli. São Paulo: Novatec Editora, 2011.

NASCIMENTO, J.; YONEYAMA, T. Inteligência artificial. Editora Blucher, 2000.

PEREZ, Anderson Luiz Fernandes. Robótica Inteligente: Tecnologias e Aplicações. Florianópolis, 2005.

TRENTIN, Paulo; DE BIASI, Herculano. Domótica via Dispositivos Móveis com Arduino.

UNITED NATIONS. Living arrangements of older persons around the world. New York: United Nations, 2005.

ANEXOS

ANEXO 1: Código utilizado em teste com piezos

```

/*
Teste com piezos: programa que faz a leitura analógica de determinado
número de piezos, e emite um alerta sonora se, em pelo menos três deles, o
valor for superior a um valor crítico.
Escrito por Wildson Oliveira.
*/

int nPiezos = 4;           // Número de piezos utilizado.
int refValue = 30;         // Valor crítico de leitura.
byte values[4];           // Array que armazenará os valores lidos.
byte portas[4] =          //
    {A0, A1, A2, A3};     // Portas analógicas para leitura dos piezos.
byte control = 0;         // Variável auxiliar.

void setup(){
    Serial.begin(9600);    // Inicia a comunicação serial.
}

void loop(){
    control = 0;           // Define Control como
    zero.
    for (int i = 0; i<nPiezos; i++){ // Faz a leitura das portas
        Serial.println(values[i]); // Imprime na serial o valor
        lido.

        // Se o valor lido for maior que o valor crítico, soma uma
        unidade à variável control.

        if (values[i] >=refValue){
            control++;
        }
    }

    // Se o valor crítico for ultrapassado em três ou mais piezos, emite
    um alerta sonoro (440 Hz) e imprime uma mensagem na serial.

    if (control >= 3){
        tone(8,440);
        Serial.println("Valor critico ultrapassado em mais de tres
        piezos!");
    }

    Serial.println("*****");
    delay(1000);           // Aguarda 1 segundo.
}

```

ANEXO 2: Orçamento dos itens utilizados no projeto

	Quant.	Valor unitário	Valor total
Arduino Uno	1	R\$ 69,00	R\$ 69,00
Placa Arduino Mini	1	R\$ 50,00	R\$ 50,00
Shield GSM EFCOM	1	R\$ 249,00	R\$ 249,00
Display LCD 16x2	1	R\$ 12,00	R\$ 12,00
Módulo I2C	1	R\$ 12,00	R\$ 12,00
Protoboard 830 furos	1	R\$ 29,00	R\$ 29,00
Chave tátil (Pushbuttons)	12	R\$ 0,60	R\$ 7,20
LEDs	3	R\$ 0,30	R\$ 0,90
Buzzer	1	R\$ 1,00	R\$ 1,00
Resistores (4x10k Ω e 3x330 Ω)	7	R\$ 0,30	R\$ 2,10
Kit 15 jumpers macho-macho (10 unidades)	1	R\$ 15,00	R\$ 15,00
Kit 15 jumpers macho-fêmea (10 unidades)	1	R\$ 15,00	R\$ 15,00
Total			R\$ 462,20

ANEXO 3: Código utilizado no Arduino Mini

```

/*
Protótipo: Carpete Inteligente
Criado em 24/12/2014. Atualizado em 05/04/2015
Wildson Oliveira

Montar o circuito da seguinte forma:

Pino    3: Ligar ao buzzer;
Pinos   6, 7, 8 e 9: Ligar cada um a uma fileira de push-butons;
Pino    10: LED verde;
Pino    11: LED amarelo;
Pino    12: LED vermelho;
Pino    A0: mySerial RX;
Pino    A1: mySerial TX;
*/

#include <Wire.h>
#include <LCD.h>
#include <LiquidCrystal_I2C.h>
#include <SoftwareSerial.h>

// Cria comunicação serial com o shield através dos pinos A0 e A1 (RX, TX)
SoftwareSerial mySerial(A0, A1);

// Localização e quantidade das fileiras de botões. Para modificar a
quantidade de fileiras no carpete, alterar estas variáveis.
int pins[4] = {6,7,8,9};
int nPins = 4;

int limite = 3;                // Limite de fileiras acionados ao mesmo
tempo

// Leds de sinalização
int ledOK = 10;                // Led de status normal.
int ledAtencao = 11;           // Led de atenção.
int ledAlerta = 12;            // Led de alerta.

// Delays utilizados no programa
const int mainTimer = 1000;    // Tempo entre as verificações de
queda.
const int alertTimer = 2000;    // Tempo que o programa esperará até
emitir o alerta, em caso de emergencia.

//Inicia o display LCD no endereço 0x20:
LiquidCrystal_I2C lcd(0x20,4, 5, 6, 0, 1, 2, 3, 7, NEGATIVE);

void setup(){
    //Inicia a comunicação serial com o PC
    Serial.begin(9600);
    Serial.println("A conexao serial foi bem estabelecida.");

    //Inicia a comunicação serial com o shield
    mySerial.begin(9600);

    // Inicia o LCD, e acende sua luz de fundo
    lcd.begin (16,2);
    lcd.setBacklight(1);

```

```

// Declara como INPUT todos os pinos declarados em pins[]
for (int x = 0; x < nPins; x++){
    pinMode(pins[x], INPUT);
}

// Declara como saída os pinos conectados aos leds
pinMode (ledOK, OUTPUT);
pinMode (ledAtencao, OUTPUT);
pinMode (ledAlerta, OUTPUT);

// Imprime mensagem no display, acende o led amarelo e aguarda 30
segundos (tempo aproximado para o carregamento do shield).
lcd.clear();
lcd.setCursor(0,0);
lcd.print("Aguardando");
lcd.setCursor(0,1);
lcd.print("shield...");

ledManager(ledAtencao);
delay(30000);

ledManager(ledOK);          // Acende o led verde
tone(3, 3250, 500);         // Emite um pequeno sinal sonoro
}

void loop(){
    // Se houver detecção de queda (forem acionados simultaneamente mais
    botões que o limite)

    if ( buttonCheck() > limite){
        atencao();          // Coloca sistema em estado de atenção.

        // Se os sensores ainda detectarem a queda
        if ( buttonCheck() > limite){
            // Executa os procedimentos de emergência.
            alerta();
        }

        // Ou retorna ao monitoramento, caso o alarme seja falso.
        else{
            Serial.println("Alarme      falso.      Retomando      o
            monitoramento...");
        }
    }

    // Se não houver detecção, coloca o sistema no estado normal.
    else{
        ok();
    }

    delay(mainTimer);      // Intervalo entre os monitoramentos.
}

// Funções auxiliares

// Função ok(): Procedimentos executados no estado normal do sistema.

void ok(){
    // Acende o led verde e imprime que tudo está normal no LCD
    ledManager(ledOK);

```

```

        lcd.clear();
        lcd.setCursor(0,0);
        lcd.print("Monitoramento");
        lcd.setCursor(0,1);
        lcd.print("Status: Normal");
    }

    // Função atencao(): Procedimentos executados em caso de suspeita de queda.
    void atencao(){
        // Acende a luz amarela e imprime no LCD que uma possivel queda foi
        // detectada e comunica também com o PC via serial
        ledManager(ledAtencao);
        lcd.clear();
        lcd.setCursor(0,0);
        lcd.print("Atencao!");
        lcd.setCursor(0,1);
        lcd.print("Possivel queda!");
        Serial.println("Possivel queda detectada! Caso o sensor permaneça
        detectando a queda, um alarme soará em 2 segundos...");

        // Aguarda o tempo alertTimer para verificar novamente.
        delay(alertTimer);
    }

    // Função alerta(): Procedimentos de emergencia em caso de queda.
    void alerta(){
        // Acende a luz vermelha e imprime mensagem no LCD.
        ledManager(ledAlerta);
        lcd.clear();
        lcd.setCursor(0,0);
        lcd.print("Alerta!");
        lcd.setCursor(0,1);
        lcd.print("Queda detectada!");

        // Manda um alerta de emergencia via serial.
        Serial.println("Alerta de emergencia!");

        // Envia um comando para o outro Arduino enviar um SMS
        if (mySerial.available()){
            mySerial.println(2);
        }

        // Emite um alarme enquanto os sensores detectarem a queda.
        while (buttonCheck() > limite ){
            tone(3, 440, 1000);
        }
    }

    // Função ledManager: Acende o led recebido como parâmetro.
    void ledManager(int led){
        // Apaga todos os leds
        digitalWrite(ledAtencao, LOW);
        digitalWrite(ledAlerta, LOW);
        digitalWrite(ledOK, LOW);
    }

```



```
        // Aciona o led recebido como parâmetro.
        digitalWrite(led, HIGH);
    }

    // Função buttonCheck: Verifica quantos botoes estao acionados
    simultaneamente (retorna tal quantidade como um inteiro).

    int buttonCheck(){
        int y = 0;

        // Faz a leitura digital de todos os pinos declarados em pins[],
        // armazena seu valor lógico e retorna a soma dos valores (número de
        // botões acionados).
        for (int x = 0; x < nPins; x++){
            int value = digitalRead(pins[x]);
            y = (y + value);
        }
        return (y);
    }
```

ANEXO 4: Código utilizado no Arduino Uno com o Shield GSM

```

/*
Programa de envio de SMS com o GSM Shield, a partir de comando via serial
Escrito por Wildson Oliveira.
Baseado no código de exemplo do Sim900, disponível em FILIPEFLOP
*/

#include "SIM900.h"
#include <SoftwareSerial.h>
#include "sms.h"

SMSGSM sms;

int numdata;
boolean started=false;
char smsbuffer[160];
char n[20];

void setup(){
    //Inicializa a serial
    Serial.begin(9600);
    Serial.print("Testando GSM shield... ");

    //Inicia a configuracao do Shield
    if (gsm.begin(2400)){
        Serial.print("nstatus=READY ");
        started=true;
    }

    else {Serial.print("nstatus=IDLE ");}
    Serial.print(">>");
}

void loop(){
    int serialRead = 0;

    // Ler a porta serial, caso ela esteja disponível
    if (Serial.available() > 0) {
        serialRead = Serial.read()-48;

        // Envia o SMS caso o Arduino receba o valor 0x2
        if (serialRead == 0x2) {
            if(started) {
                if (
                    sms.SendSMS("XXXXXXXX", "Alerta! Queda
                    detectada! [Teste com o Carpete
                    Inteligente]")
                    && sms.SendSMS("YYYYYYYY", "Alerta! Queda
                    detectada! [Teste com o Carpete
                    Inteligente]")
                    //Adicionar aqui todos os destinatários
                )
                    Serial.print("nSMS sent OK ");
            }
        }
    }
}

```