

MARCEL HOSHINO E RENATA WAI

Detector e alarme de quedas vestível para a terceira idade

SALUS - Sensor And Alert Unified System For Falls

São Paulo

2014

MARCEL HOSHINO E RENATA WAI

Detector e alarme de quedas vestível para a terceira idade

SALUS - Sensor And Alert Unified System For Falls

Entrega do Trabalho de Formatura à
Escola Politécnica da Universidade de São
Paulo para obtenção da Graduação de
Engenharia Mecatrônica.

Orientador: Prof. André Riyuiti Hirakawa

Co-Orientador: Oswaldo Horikawa

São Paulo

2014

Catálogo-na-publicação

Hoshino, Marcel Jun

SALUS – Sensor and Alert Unified System for Falls: detector e alarme de quedas vestível para a terceira idade / M.J. Hoshino; R.Y.P. Wai. – São Paulo, 2014.

145 p.

Trabalho de Formatura - Escola Politécnica da Universidade de São Paulo. Departamento de Engenharia Mecatrônica e de Sistemas Mecânicos. Departamento de Engenharia de Computação e Sistemas Digitais.

1.Monitoramento 2.Idosos 3.Acidentes por quedas I.Wai, Renata Yukie Pacheco II.Universidade de São Paulo. Escola Politécnica. Departamento de Engenharia Mecatrônica e de Sistemas Mecânicos. Departamento de Engenharia de Computação e Sistemas Digitais. III.t.

Dedicamos esse trabalho aos nossos pais,
que muito nos apoiaram e sempre
acreditaram em nosso potencial.

AGRADECIMENTOS

Aos professores Andre Hirakawa e Oswaldo Horikawa, pela orientação ao longo de todo o projeto.

Aos amigos das diversas áreas da faculdade, que nos apoiaram e nos estimularam durante todo o projeto e, em diversas ocasiões, colaboraram de maneira direta ou indiretamente no desenvolvimento do projeto.

O envelhecimento da população é um triunfo para a humanidade, mas também um desafio para a sociedade.

(Organização Mundial da Saúde)

LISTA DE FIGURAS

Figura 1 - Movimento inicial da queda.....	3
Figura 2 - Diagrama de Casos de Uso	13
Figura 3 - Fluxo da informação.....	18
Figura 4 - Sistema esquemático do funcionamento do acelerômetro.....	20
Figura 5 – Representações de algoritmos de localização	24
Figura 6 - Exemplo de serviço USSD	35
Figura 7 - Casos de uso - aplicativo	41
Figura 8 - Arduino Uno v3	45
Figura 9 - Raspberry Pi Modelo B	46
Figura 10 - BeagleBone Black.....	47
Figura 11 - Caminho da informação - especificação	52
Figura 12 - Estrutura do dispositivo	53
Figura 13 - MPU-6050 – vista frontal.....	59
Figura 14 - MPU-6050 – vista posterior.....	60
Figura 15 - Módulo GSM Sim900	60
Figura 16 – Fluxo final	63
Figura 17 - Fluxograma do GSM	64
Figura 18 - Software Trigger GPRS	65
Figura 19 - Fluxograma da leitura da serial	67
Figura 20 - Fluxograma do envio de SMS	69
Figura 21 - Fluxograma do cadastro de celular	70
Figura 22 - Perfil de queda.....	73
Figura 23 - Fluxograma de detecção de queda	74
Figura 24 - Fluxograma do recebimento de SMS	80
Figura 25 - Notificação de queda	80
Figura 26 - Visualização do marcador no mapa	82
Figura 27 - Diagrama de classes do aplicativo	84
Figura 28 - Interface do aplicativo	87
Figura 29 - Montagem do divisor de tensão da bateria.	88
Figura 30 - Queda de costas.....	94
Figura 31 - Queda lateral.....	94

Figura 32 - Queda frontal.	95
Figura 33 - Queda sentado.....	95
Figura 34 - Atividade de correr.	96
Figura 35 - Atividade de sentar e deitar.....	97
Figura 36 - Atividade de caminhar.....	97
Figura 37 - Pinagem do dispositivo	99
Figura 38 - Comando de verificação de rede	121
Figura 39 - Habilitação do GSM para a visualização e obtenção dos dados das antenas.....	122
Figura 40 - Comando de envio de SMS	123
Figura 41 - Procedimento de cadastro de celular.....	124
Figura 42 - Sequência para inserção do chip	125
Figura 43 - Exemplo de mensagem de queda.....	127
Figura 44 – Procedimento na notificação de queda	127
Figura 45 - Visualização do local da queda.....	128

LISTA DE TABELAS

Tabela 1 - Comparativo de plataformas para smartphones.....	40
Tabela 2 - Comparativo de processadores.....	48
Tabela 3 - Geolocation - Dados utilizados.....	50
Tabela 4 - Teste de localização - API.....	51
Tabela 5 - Comparativo Arduino.....	57
Tabela 6 - Distribuição das Versões Android	61
Tabela 7 - Mapa do registrador 28.	75
Tabela 8 - Configuração da amplitude máxima do acelerômetro.	75
Tabela 9 - Mapa do registrador 29.	76
Tabela 10 - Mapa do registrador 30.	77
Tabela 11 - Mapa do registrador 56.	78
Tabela 12 - Mapa do registrador 105.	78
Tabela 13 - Configuração do decremento do contador da interrupção.	79
Tabela 14 - Tabela para a construção de divisor de tensão.....	89
Tabela 15 - Leitura de dados do MPU-6050 em queda.....	93
Tabela 16 - Protótipos	100
Tabela 17 - Teste de detecção de queda	106
Tabela 18 - Teste de detecção de queda	106
Tabela 19 - Teste do dispositivo em atividades do cotidiano	107
Tabela 20 - Teste de tempo de envio do SMS	108
Tabela 21 - Teste de tempo de recebimento do SMS	108
Tabela 22 - Testa da precisão da localização	109

RESUMO

O presente projeto tem como objetivo melhorar a qualidade de vida das pessoas idosas, ao garantir um atendimento rápido e eficiente em eventos adversos em que ele sofra acidentes por queda. Assim, a proposta do projeto é elaborar um dispositivo portátil capaz de detectar quando o usuário se envolve em um evento de queda e, tendo ocorrido o acidente, emitir alertas para cuidadores indicando o local do evento para que as devidas medidas sejam tomadas. Para isso, o projeto foca em três pontos principais: o monitoramento, a emissão de alerta e a localização. Para o monitoramento, é necessário que um sensor rastreie, continuamente, enquanto em uso, as variações de movimento do usuário e que seja capaz de diferenciar atividades comuns e quedas. Para a emissão de alerta, é necessário garantir que ele seja enviado com sucesso, mantendo sua integridade durante a recepção e que seja enviado em pouco tempo após o acidente. Finalmente, para a localização, é preciso que ela seja precisa o suficiente para que seja possível que o cuidador encontre o usuário facilmente.

ABSTRACT

The present project aims to improve the quality of life of the elderly by ensuring a fast and efficient care in adverse events in which the elderly falls. Therefore, the project's proposal is the creation of a portable device capable of detecting when the user is involved in a fall accident. And, once it detects the fall, it sends an alert for the caregivers, indicating the location of the accident, so the appropriate measures may be taken. In order to do this, the project focuses in three main points: monitoring, alert emission and location. For the monitoring, it is necessary that a sensor continuously monitor, while in use, the user's movements and it is capable of distinguishing between a normal activity and a fall. For the alert emission, it is necessary to guarantee that the message is successfully sent and that it is sent, and received, shortly after the fall. Finally, for the location, it is necessary to be precise enough so the rescue may be able to find the user quickly.

SUMÁRIO

1	Introdução	1
1.1	Contexto	1
1.2	Objetivo	4
1.3	Escopo	4
1.4	Estado da arte	4
2	Metodologia.....	7
2.1	Análise dos requisitos	7
2.2	Requisitos.....	7
2.3	Componentes	7
2.4	Integração dos componentes	8
2.5	Simulações.....	8
3	Desenvolvimento.....	11
3.1	Análise dos requisitos	11
3.1.1	Operação	11
3.1.2	Funções do software	11
3.1.3	Características dos usuários.....	12
3.2	Requisitos.....	12
3.2.1	Casos de uso.....	12
3.2.2	Requisitos funcionais.....	14
3.2.3	Requisitos não-funcionais.....	16
3.2.4	Fluxo da Informação	17
3.3	Métodos	18
3.3.1	Detecção de queda.....	18
3.3.2	Localização – ambientes externos.....	22
3.3.3	Localização – Ambiente domiciliar.....	28
3.3.4	Bateria	31

3.3.5	Percepção de uso.....	33
3.3.6	Alertas	34
3.3.7	Cadastro de telefones.....	37
3.3.8	Interface para alertas e localização	39
3.3.9	Processador.....	43
3.4	Testes e simulações.....	49
3.4.1	Geolocation API.....	49
3.5	Estrutura.....	52
4	Implementação.....	55
4.1	Escolha dos componentes	55
4.1.1	Smartphone como detector de queda.....	55
4.1.2	Arduino	57
4.1.3	Detector de queda	58
4.1.4	Módulo GSM.....	60
4.1.5	Bateria	61
4.1.6	Aplicativo	61
4.2	Softwares dos módulos	63
4.2.1	Módulo GSM.....	64
4.2.2	Unidade de processamento de movimento.....	72
4.2.3	Aplicativo	79
4.2.4	Medidor de bateria.....	88
4.3	Testes dos módulos	90
4.3.1	Módulo GSM.....	91
4.3.2	Unidade de processamento de movimento.....	92
4.3.3	Aplicativo	98
4.4	Integração	99
4.4.1	Protótipo 1	100

4.4.2	Protótipo 2	101
4.4.3	Protótipo 3	102
4.4.4	Protótipo 4	102
4.4.5	Protótipo 5	103
4.5	Testes	105
4.5.1	Teste 1 – Quedas	105
4.5.2	Teste 2 – Atividades do cotidiano	107
4.5.3	Teste 3 – Tempo de envio de sms.....	107
4.5.4	Teste 4 – Tempo de recebimento de sms.....	108
4.5.5	Teste 5 – Localização.....	109
4.5.6	Teste 6 – Dispositivo sem sinal de antena.....	109
4.6	Análise dos resultados	109
5	Conclusão	111
5.1	Projeto final	111
5.2	Trabalhos futuros	112
6	Bibliografia	113
Apêndice A	Programa para teste do módulo GSM	120
Apêndice B	Manual de instruções	125

1 INTRODUÇÃO

Nesta seção descrevemos o contexto, o objetivo e o escopo do projeto. Também relatamos o estado da arte neste tema.

1.1 CONTEXTO

O envelhecimento da população é um fenômeno que já não se restringe apenas a países desenvolvidos, mas sim, ao mundo todo. E tal acontecimento só tende a ser acelerado. Segundo a ONU, a porcentagem de idosos no mundo em 2005 era de 14,8%, já em 2050, espera-se que tal número suba para 28,8%. Entende-se por idoso as pessoas com idade igual ou acima de 60 anos. A consequência direta disso é que se torna cada vez mais necessária a atenção aos problemas comuns que atingem a essa parcela da sociedade e, a partir deles, propor soluções eficazes de modo a prevenir ou até sanar tais problemas. (WORLD HEALTH ORGANIZATION, 2007)

Entre os principais problemas que atingem os idosos, acidentes de quedas são um dos mais comuns e podem trazer graves consequências à saúde do idoso. Define-se queda como o ato de inadvertidamente atingir o chão ou outro nível inferior, excluindo-se mudanças intencionais para repouso em objetos diversos como, por exemplo, móveis. Segundo dados apresentados pela WHO (World Health Organization), aproximadamente 30% das pessoas acima de 65 anos de idade sofrem ao menos uma queda por ano. Tal índice aumenta para 44% quando consideradas pessoas acima de 70 anos. Não obstante, cerca de 40% dos idosos que já sofreram uma queda durante a velhice voltam a se acidentarem novamente. Além disso, o estudo mostrou também que, das mortes de idosos causadas por ferimentos, 40% são decorrentes de queda. Assim, é preciso dar uma atenção especial a esse problema. (WORLD HEALTH ORGANIZATION, 2007)

Estudos indicam que uma queda pode ter diversos efeitos sobre a vida do idoso, que não se limitam aos danos físicos. A perda de confiança após um acidente de queda é um dos maiores problemas encontrados, limitando as atividades e liberdade do idoso, e aumentando sua insegurança. Dos idosos que caem, mais de 75% relata

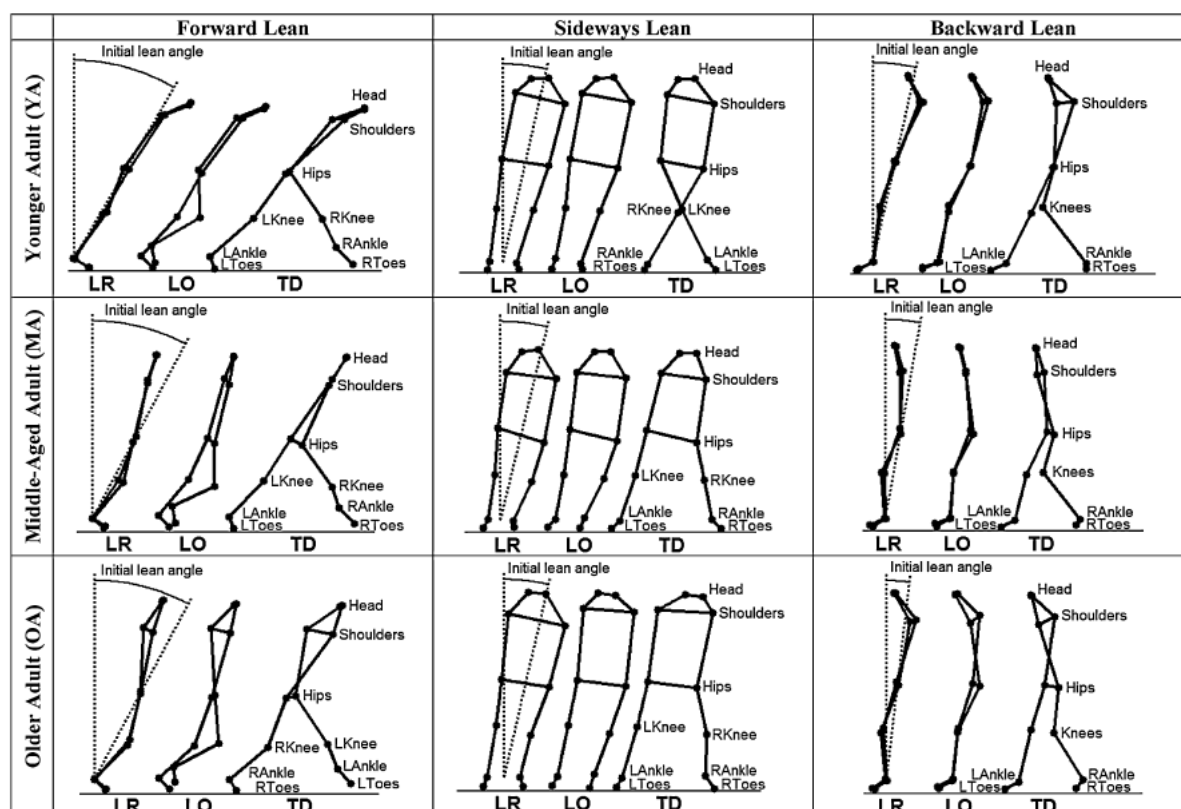
que mudou seus hábitos e atividades após o acidente, restringindo seu movimento e liberdade (OVERCASH, JR. e SCHAICK, 2010).

As quedas comumente estudadas são: frontais, laterais e de costas. As quedas mais difíceis de serem evitadas, após uma inclinação inicial, são as de costas, seguidas pelas laterais e as frontais (CARBONNEAU e SMEESTERS, 2014). No entanto as quedas frontais são as mais comuns em idosos (OVERCASH, JR. e SCHAICK, 2010).

Além das diferenças de direção também foi encontrada uma variação na duração da queda quando os participantes dos estudos são idosos. Para essa faixa etária pode-se perceber dois grupos, um que caía devagar, e outro que caía rápido. No primeiro grupo encontram-se os idosos que, tendo ciência de suas fraquezas, se movimentam de maneira mais cautelosa e vagarosa. Já no segundo grupo estão os idosos que por se apresentarem saudáveis e fortes se movimentavam com maior confiança, tal como pessoas mais jovens. No entanto, ambos grupos apresentaram, igualmente, maior propensão a quedas que pessoas mais jovens. A causa apontada para esse maior risco de quedas foi a força muscular da panturrilha e a velocidade com que esse músculo consegue ser utilizado, já que, no ato da queda, a pessoa dá um passo para tentar reequilibrar o corpo, sendo a panturrilha essencial para esse reequilíbrio. (TYTELL, 2004).

A dinâmica do passo utilizado para o reequilíbrio é mostrado na Figura 1. No estudo, o corpo do participante era inclinado de modo a induzir uma queda, essa inclinação corresponde ao LR. O LO corresponde à força exercida na perna de apoio e o TD corresponde ao passo dado pelo participante de modo a evitar a queda.

Figura 1 - Movimento inicial da queda



Fonte: CARBONNEAU e SMEESTERS (2014)

LR:Lean Release

LO: Liftoff

TD: Touchdown

Este estudo mostra que o idoso tende a cair com inclinações do corpo menores que as dos jovens. Os jovens tendem a se recuperar de inclinações 48% maiores que as dos idosos (CARBONNEAU e SMEESTERS, 2014).

Outros estudos procuram definir qual o limiar de tempo a partir do qual a pessoa não consegue evitar a queda. Este limiar é da ordem de 700 ms (SMEESTERS, HAYES e MCMAHON, 2001).

Os procedimentos iniciais para atender a um paciente que sofreu uma queda são o de imobilizar o acidentado, acalmá-lo e chamar por socorro médico. No entanto, nem sempre o acidentado está acompanhado por pessoas que possam lhe prestar socorro ou está em condições de chamar por ajuda, o que justifica a necessidade de uma solução que garanta um pronto-atendimento rápido e automático para esse tipo de acidente. Além do mais, atualmente, há uma grande dificuldade em se documentar as características da queda, tais como o local e hora do evento,

procedimento que é recomendado pelos profissionais do setor. (GOVERNMENT OF WESTERN AUSTRALIA DEPARTMENT OF HEALTH, 2013)

1.2 OBJETIVO

O presente projeto tem como objetivo propor uma ferramenta capaz de otimizar e melhorar o atendimento aos idosos no momento em que o mesmo se acidenta. Para isso, tendo em vista os altos índices de acidentes de queda e as possíveis consequências que esses casos podem trazer à saúde dos idosos, optou-se por focar nesse problema.

1.3 ESCOPO

Com o objetivo acima em mente, o escopo do projeto foi o de desenvolver um dispositivo vestível capaz de identificar quedas de pessoas da terceira idade e que, ocorrido a identificação, consiga fazer a localização do evento e o envio de um alerta aos cuidadores do acidentado.

1.4 ESTADO DA ARTE

O problema de quedas de idosos é algo que há muito já vem sendo enfrentado em países desenvolvidos. Assim, diversos estudos foram realizados sobre as causas, consequências e características das quedas, e também foram propostos diversos mecanismos para se detectar uma queda, tal como será descrito a seguir.

Atualmente, existem diversas pesquisas que propõe métodos que visam o monitoramento e/ou o pronto atendimento aos idosos com o intuito de identificar quedas. Dentre as propostas existentes, pode-se agrupá-las em três grandes grupos: método fixo, método vestível e método misto.

Os métodos fixos são todos aqueles em que o monitoramento do idoso é feito através de dispositivos que ficam acoplados no ambiente em que o mesmo se encontra. Já os métodos vestíveis englobam aqueles em que o monitoramento é

feito através do uso de dispositivos que ficam junto ao corpo do usuário. Finalmente, os métodos mistos são os que utilizam ambos os métodos anteriores.

Confrontando os diferentes métodos, estudos indicam que os métodos de monitoramento fixo possuem uma maior aprovação entre os potenciais usuários. Isso ocorre devido ao fato do método ser o menos invasivo possível, ao passo que os outros dois, por mais discretos que sejam os dispositivos, acabam incomodando minimamente o usuário, seja fisicamente, seja psicologicamente. Além do mais, métodos vestíveis precisam estar junto ao corpo do idoso para que possam funcionar, fazendo com que este método dependa da mudança de hábito do usuário de fazer uso do mesmo. (SIXSMITH e JOHNSON, 2004)

No entanto, os métodos fixos possuem uma taxa de falso-positivos mais elevadas que as dos métodos vestíveis. Isso ocorre pois o monitoramento fixo possui dificuldades de diferenciar o idoso no espaço, podendo confundi-lo com outras pessoas ou até com animais. Além disso, a presença de objetos no ambiente monitorado pode comprometer o monitoramento (SIXSMITH e JOHNSON, 2004), (POPESCU, LI, *et al.*, 2008). Por outro lado, os métodos vestíveis possuem uma taxa de falso-positivos inferior devido ao perfil de comportamento que o idoso possui em geral: baixa mobilidade e movimentos sutis. (SOAZ, LEDERER e DAUMER, 2012)

Com um escopo muito semelhante ao do projeto, (ZHENG, ZHANG e WU, 2009) e (CAMPO e GRANGEREAU, 2008) apresentam propostas alternativas para alcançar o objetivo em comum. (CAMPO e GRANGEREAU, 2008) faz o uso de um acelerômetro tri-axial para fazer a detecção de queda, um GPS para a localização e um módulo Zigbee (protocolo de comunicação sem fio de baixo alcance) para o envio de alertas. Já (ZHENG, ZHANG e WU, 2009) também faz o uso de um acelerômetro para a detecção de queda e um módulo de GPS para a localização e o envio de alerta via SMS.

Dentre as duas propostas, a segunda é a mais próxima ao SALUS. Tal convergência vem dos métodos de detecção de queda através do uso de acelerômetro e do modo em que o alerta é enviado, via SMS. A divergência, porém, vem do modo em que é feito a localização. (ZHENG, ZHANG e WU, 2009) propõe o uso do GPS para fazer a localização. No entanto, tal método limita-se à ambientes aonde há sinal de GPS, em geral, lugares abertos. Já o SALUS utiliza a triangulação de antenas de celular para fazer a localização, ampliando os ambientes de uso.

É válido citar também (THAMMASAT e CHAICHARN, 2012) que propõe o uso de celulares para a identificação de quedas.

2 METODOLOGIA

Nesta seção será descrita a metodologia que foi adotada nesse projeto para definir os requisitos, os componentes e a implementação do projeto. Os tópicos seguintes descrevem as metodologias escolhidas.

2.1 ANÁLISE DOS REQUISITOS

Para iniciar o projeto, realizou-se o levantamento dos requisitos. Dentre as principais características que foram levadas em conta, tem-se a definição dos principais cenários em que o sistema iria operar, quais as funcionalidades de software que seriam essenciais para o projeto, independentemente dos métodos escolhidos para a implementação do projeto e a identificação das características mais relevantes do usuário final do projeto, o idoso.

2.2 REQUISITOS

Na especificação do projeto foi necessário apontar os requisitos funcionais e não-funcionais do sistema. Para tal, optou-se por utilizar a linguagem de modelagem UML para que fosse possível enumerar todos os casos de uso do projeto e os caracterizar.

Finalmente, com base nos casos de uso, identificaram-se os requisitos funcionais e não funcionais do projeto e foi possível atribuir valores mensuráveis a cada um dos requisitos, utilizando valores condizentes com a realidade.

2.3 COMPONENTES

Com a definição dos requisitos do projeto, foi necessário definir os componentes que iriam compor o projeto. Para essa atividade, inicialmente, dividiu-se as principais funcionalidades que compunham o escopo do projeto e, para cada uma dessas funcionalidades, foi feito um estudo em artigos e livros sobre as mesmas e quais os métodos mais utilizados para a funcionalidade em questão.

Após o estudo, listaram-se os diferentes métodos e, levando em consideração os requisitos levantados, optou-se pelo método mais adequado ao projeto.

Finalmente, com base nos métodos escolhidos para cada funcionalidade, foi feito um levantamento de diversos componentes eletrônicos que pudessem ser utilizados no projeto. Para a escolha dos componentes, além da funcionalidade, foi levado em conta o preço, disponibilidade no mercado, facilidade de uso e facilidade de integração com outros componentes.

2.4 INTEGRAÇÃO DOS COMPONENTES

Após a escolha e compra dos componentes, primeiramente foi feita uma manipulação individual de cada um dos componentes para que pudesse ser feitos testes de funcionamento, estudo aprofundado do componente e sua comunicação. Feito isso, foram implementados as comunicações individuais de cada um dos componentes com o microprocessador do projeto e, após cada uma das integrações individuais, eram realizados testes de operação para garantir o funcionamento do componente. O passo final foi a integração completa de todos os componentes com o microprocessador e, assim, a montagem do protótipo do projeto.

2.5 SIMULAÇÕES

Ao longo do desenvolvimento do projeto, foram realizados alguns testes, tanto dos componentes quanto dos protótipos para que fosse possível entender, estudar e aperfeiçoar os itens testados. As primeiras simulações foram feitas com os componentes individuais para se testar o funcionamento dos mesmos e para auxiliar na elaboração dos algoritmos a serem implementados em cada um dos componentes. Já as outras simulações foram feitas com os diversos protótipos do projeto e o intuito delas era o de melhorar os mesmos. Tais melhorias podem ser resumidas em ligações elétricas entre os componentes, tempo de aquisição e resposta do sistema e parâmetros limiares a serem definidos no algoritmo de detecção de queda.

Um cuidado que se tomou foi garantir que todas as simulações fossem realizadas em condições parecidas entre si e similar aos cenários de operação definidos anteriormente.

3 DESENVOLVIMENTO

Nesta seção será descrita como foi o desenvolvimento do projeto ao longo do ano, tendo como referência a metodologia adotada.

3.1 ANÁLISE DOS REQUISITOS

Para se definir os requisitos do projeto, foi necessário fazer algumas análises a respeito do funcionamento do dispositivo, das características de seu usuário principal e dos cenários de operação ao qual ele seria submetido. Essa análise prévia se fez necessária pois, dependendo das condições de uso ao qual o dispositivo seria exposto, algumas funcionalidades poderiam ser essenciais ou não.

3.1.1 OPERAÇÃO

Durante a concepção do projeto considerou-se que o cenário de operação do dispositivo se resume apenas a atividades relacionadas ao cotidiano do idoso: caminhar, correr, sentar, deitar, levantar e estar em repouso. Assim, atividades como exercícios de academia e esportes, por exemplo, fogem do escopo do projeto.

3.1.2 FUNÇÕES DO SOFTWARE

As principais funções que foram implementados no projeto são as de:

- Detecção de queda: capaz de fazer o monitoramento constante das atividades do usuário e, caso este sofre uma queda, conseguir identificar tal evento de maneira rápida;
- Localização geográfica do dispositivo: capaz de obter e armazenar essa informação para ser enviada em uma mensagem;
- Envio de alertas de queda: capaz de enviar uma mensagem especificando localização e hora do evento para os cuidadores cadastrados no sistema e

enviar mensagens de níveis críticos de carga da bateria também para os cuidadores;

- Cadastramento do cuidador: recebe e armazena em sua memória o cadastro dos cuidadores para o envio dos alertas.

3.1.3 CARACTERÍSTICAS DOS USUÁRIOS

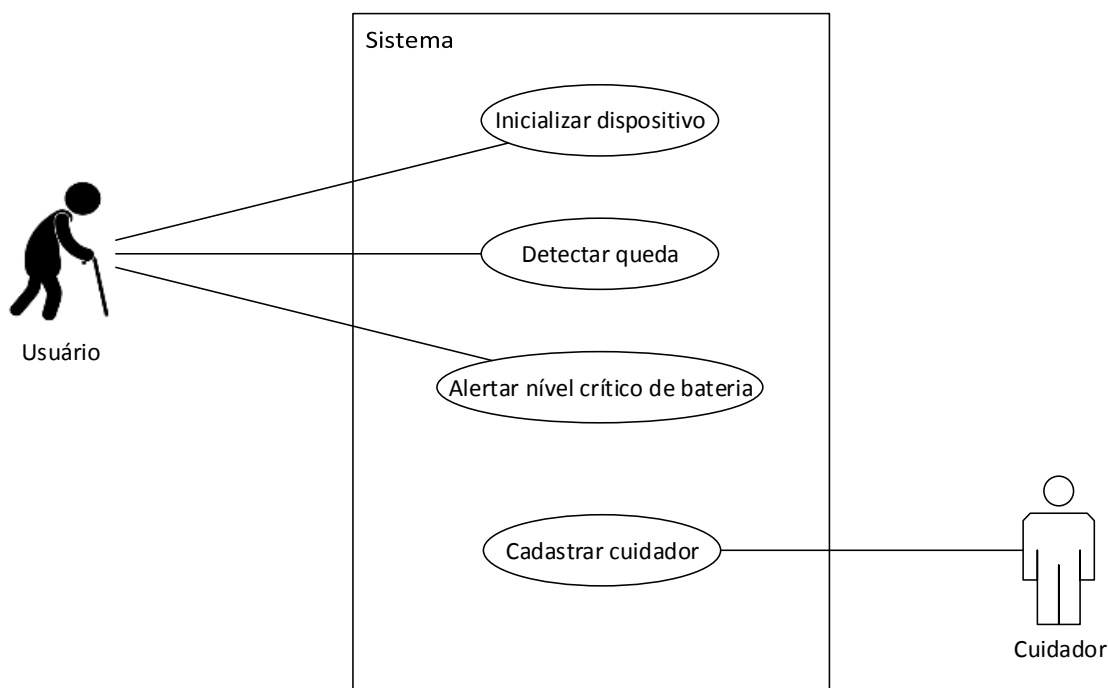
Para o projeto, existem dois perfis de usuários envolvidos no sistema: o usuário monitorado e o seu cuidador. O usuário monitorado é composto por um grupo de pessoas mais suscetíveis a sofrer quedas, em geral, este grupo é formado por idosos. A característica dos integrantes deste grupo é que eles necessitam de monitoramento constante, mas também procuram independência e privacidade. Já cuidadores são aqueles próximos aos usuários monitorados e que se preocupam com o bem estar destes.

3.2 REQUISITOS

Tendo em mente as condições de uso ao qual o dispositivo será submetido, foi possível definir e caracterizar quais seriam os requisitos funcionais e não funcionais do projeto.

3.2.1 CASOS DE USO

Tomando como base o modelo UML, foi feito um diagrama de casos de uso para o projeto. O diagrama se encontra a seguir.

Figura 2 - Diagrama de Casos de Uso

Fonte: HOSHINO e WAI (2014)

Caso de uso 1: Inicializar dispositivo

Descrição: Este caso de uso descreve o processo de ativação do monitoramento de queda.

Evento iniciador: O dispositivo é “vestido” pelo usuário

Atores: Dispositivo

Pré-condição: Bateria com carga

Cenário:

- O dispositivo percebe que está sendo utilizado
- O sistema ativa o monitoramento de queda
- O sistema ativa o monitoramento da carga da bateria

Pós-condição: Monitoramento de queda e carga ativos

Caso de uso 2: Detectar queda

Descrição: Este caso de uso descreve o processo de detecção de queda do dispositivo.

Evento iniciador: O dispositivo sofre uma queda

Atores: Dispositivo

Pré-condição: Dispositivo inicializado

Cenário:

- O sistema detecta a queda
- O sistema obtém a localização do dispositivo
- O sistema envia um alerta de queda com a localização

Pós-condição: Alerta de queda enviado

Caso de uso 3: Alertar nível crítico de bateria

Descrição: Este caso de uso descreve o processo de detecção do nível crítico de bateria

Evento iniciador: Baixa carga da bateria

Atores: Dispositivo

Pré-condição: Dispositivo inicializado

Cenário:

- O sistema detecta o baixo nível de bateria
- O sistema envia um alerta de baixo nível de bateria

Pós-condição: Alerta de nível crítico de bateria enviado

Caso de uso 4: Cadastrar cuidador

Descrição: Este caso de uso descreve o processo de cadastro do cuidador

Evento iniciador: Cuidador envia um pedido de cadastro ao sistema

Atores: Cuidador

Pré-condição: Dispositivo ligado

Cenário:

- O sistema recebe o pedido de cadastro
- O sistema cadastra o cuidador para a emissão de alertas

Pós-condição: Cuidador cadastrado

3.2.2 REQUISITOS FUNCIONAIS

No.: RF01 – Monitoramento de queda

Descrição: O sistema é capaz de monitorar constantemente possíveis quedas.

No.: RF02 – Detecção de queda

Descrição: O sistema, a partir dos dados do monitoramento, é capaz de identificar quedas.

No.: RF03 – Obtenção da localização

Descrição: O sistema é capaz de obter a localização do dispositivo. Para esse requisito, é interessante que o sistema tenha uma precisão diferenciada entre diferentes ambientes de operação. Para ambientes internos, devido à presença de diversos obstáculos visuais, é interessante uma precisão maior na localização do idoso na ordem de 1 a 10 metros. Já para ambientes externos, devido a maior facilidade de se encontrar uma pessoa, a precisão pode ser menor, na ordem de 100 metros.

No.: RF04 – Registrar o momento da queda

Descrição: O sistema é capaz de registrar a data, hora e minuto em que ocorre a queda.

No.: RF05 – Envio de alertas de queda

Descrição: O sistema envia ao cuidador cadastrado um alerta informando a ocorrência de quedas, junto com a localização e o instante.

No.: RF06 – Envio de alertas de nível crítico de bateria

Descrição: O sistema envia ao cuidador cadastrado um alerta informando que a bateria atingiu um nível crítico.

No.: RF07 – Percepção de uso

Descrição: O sistema supõe que está em uso quando estiver ligado.

No.: RF08 – Monitoramento da carga de bateria

Descrição: O sistema é capaz de monitorar constantemente o nível de carga da bateria.

No.: RF09 – Detecção de nível crítico de bateria

Descrição: O sistema detecta quando a bateria atinge um nível crítico de carga, equivalente a 8h restantes de operação.

No.: RF10 – Receber cadastro de cuidador

Descrição: O sistema é capaz de receber pedidos de cadastro de cuidador.

No.: RF11 – Cadastrar cuidador

Descrição: O sistema é capaz de cadastrar os dados do cuidador, recebido no pedido, para ser utilizado no envio de alertas. O dado cadastrado deve ser o número de celular do cuidador.

No.: RF12 – Peso

Descrição: O dispositivo pesará menos que 100g, peso médio dos aparelhos celulares, dispositivo comumente utilizado pelas pessoas.

No.: RF13 – Dimensões

Descrição: O volume máximo do dispositivo será 50 cm³.

No.: RF14 – Durabilidade de bateria

Descrição: A bateria deve durar pelo menos uma semana.

No.: RF15 – Resistência à quedas

Descrição: O dispositivo deve ser resistente a impactos ocasionado por quedas. No mínimo, deve suportar um impacto de 700N (peso médio de um homem adulto, 70kg, multiplicado pela força da gravidade).

3.2.3 REQUISITOS NÃO-FUNCIONAIS

No.: RNF01 – Durabilidade de bateria

Descrição: A bateria deve durar pelo menos uma semana.

No.: RNF02 – Baixo custo

Descrição: O método mais comum para fazer o monitoramento de idosos é através de câmeras de monitoramento ou de profissionais. Para o primeiro caso, é comum colocar câmeras em pelo menos três cômodos da casa: quarto, sala, cozinha (ou banheiro), ou seja, média de 3 câmeras. Como cada câmera custa, no mínimo, R\$60,00, gasta-se cerca de R\$180,00 para esse método. Já para a contratação de cuidadores profissionais, se gasta, no mínimo, um salário mínimo mensal (R\$724,00 no presente ano, 2014). Assim, um valor abaixo de R\$180,00 já pode ser considerado um baixo custo para o presente projeto.

No.: RNF03 – Usabilidade

Descrição: O dispositivo dispensa a interação com o usuário monitorado, facilitando o seu uso e aumentando a chance de adesão ao projeto por parte do idoso.

No.: RNF04 – Confiabilidade

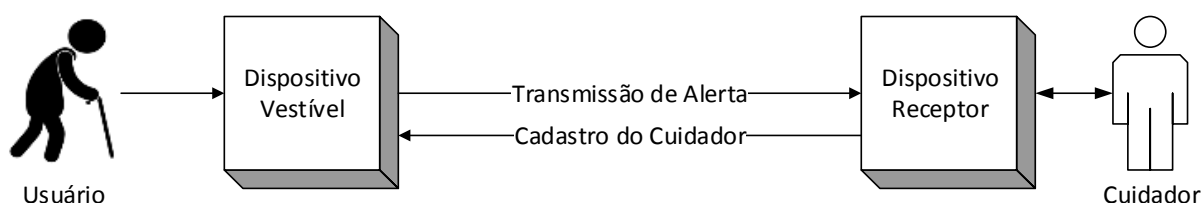
Descrição: O sistema irá sempre gerar alertas, mesmo que falso-positivos, no entanto, o número de falso-positivos será de, no máximo, um por semana de monitoramento.

No.: RNF05 – Desempenho

Descrição: O sistema elabora e envia as mensagens em menos de 30 segundos após ocorrido a queda.

3.2.4 FLUXO DA INFORMAÇÃO

Considerando os casos de uso e os requisitos apresentados anteriormente criou-se um diagrama, apresentado na Figura 3, indicando o caminho que a informação percorre, desde a movimentação do idoso até o alerta ao cuidador.

Figura 3 - Fluxo da informação

Fonte: HOSHINO e WAI (2014)

3.3 MÉTODOS

Após a definição e caracterização dos requisitos funcionais e não funcionais do dispositivo, identificaram-se as principais funcionalidades do dispositivo. Com isso, foi possível listar e eleger o método mais adequado para se atender à funcionalidade em questão do modo mais adequado, tendo em mente, o escopo do projeto.

3.3.1 DETECÇÃO DE QUEDA

Existem diversas maneiras de se detectar quedas. Seguindo o escopo deste projeto, a detecção de queda deveria ser feita de maneira autônoma e em qualquer ambiente, de tal forma que o monitoramento fixo citado anteriormente não seria aplicável. Desse modo, apenas os métodos de detecção de queda utilizando componentes eletrônicos que ficam atrelados ao corpo do usuário são válidos.

Entre os componentes encontrados que podem permitir a detecção de quedas, encontraram-se o acelerômetro, que seria utilizado para identificar acelerações atípicas ao cotidiano de um idoso e similares a queda; o giroscópio, que identificaria variações no eixo de coordenadas do corpo do idoso; o altímetro, que identificaria variações de altura do sensor; o magnetômetro, que agiria como uma bússola magnética. Segue abaixo a descrição de cada um dos métodos identificados:

3.3.1.1 ALTÍMETRO

Os altímetros elétricos conseguem estimar a altura em que o aparelho se encontra a partir da medição da pressão atmosférica do ambiente. No entanto, os altímetros

atuais possuem uma precisão de no máximo 3 metros, valor muito maior que o adotado nos requisitos.

3.3.1.2 MAGNETÔMETRO

Os magnetômetros são sensores capazes de medir campos magnéticos presentes ao seu redor. Devido a essa característica, é possível localizar o Norte magnético da terra e, conseqüentemente, fazer medidas de variações angulares. O problema desse dispositivo é que ele exige calibração toda vez que se muda de ambiente, ou que se sofra interferências de outros dispositivos, para que o Norte magnético seja localizado, não sendo muito prático.

3.3.1.3 GIROSCÓPIO

De acordo com a segunda lei de Newton, um objeto em movimento permanece em movimento, linear ou circular, a menos que uma força externa aja sobre o mesmo. E, no momento que o objeto sofre tal força, o sistema gera uma força de resistência contrária a ela. É com esse princípio que se baseia o funcionamento do giroscópio eletrônico. Esse componente é constituído por um conjunto de uma massa de silicone que permanece em constante vibração e um apoio que fixa essa massa ao restante do sistema. Quando o componente sofre uma variação angular, o silicone sofre deformações e, devido à sua propriedade piezoelétrica, produz corrente elétrica. Com isso, é possível associar diferentes amperagens de corrente com diferentes velocidades angulares.

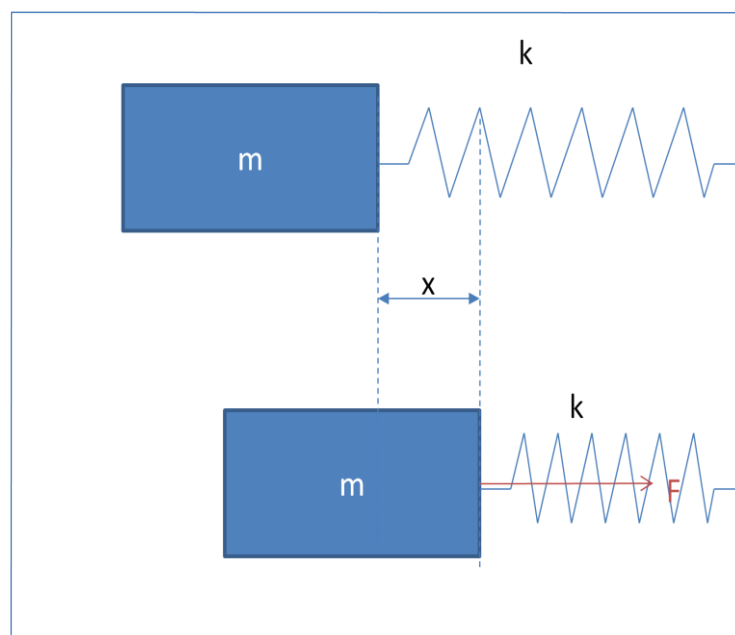
O uso do giroscópio no projeto é interessante na medida em que a partir dele, pode-se identificar as velocidades angulares às quais o eixo do corpo do usuário está sendo submetido ao longo do tempo. No entanto, o uso único do giroscópio não produz o número de parâmetros necessários para se caracterizar uma queda, uma vez que muitas das atividades do cotidiano envolvem mudanças bruscas na direção do eixo do corpo, tais como deitar-se, por exemplo. Assim, torna-se essencial o uso de outro componente em conjunto, o acelerômetro (BRENDLER, 2002).

3.3.1.4 ACELERÔMETRO

Acelerômetros são sensores capazes de medir acelerações de um dado corpo ao qual se encontram acoplados. Seu funcionamento baseia-se em correlacionar a medida de um parâmetro que seja proporcional a aceleração com a aceleração propriamente dita. Ou seja, a obtenção do dado é feita de maneira indireta.

Existem diversos tipos de acelerômetros e cada tipo adota um parâmetro distinto para se obter a aceleração. O exemplo clássico para se ilustrar seu funcionamento é o sistema massa-mola em que uma massa inercial aplica tensões às molas e essas, por sua vez, se deformam em um comprimento x para compensar tais deformações. Sabendo o coeficiente da mola e sua deformação, pode-se obter a força que a massa inercial está aplicando na mola (eq. 1) e, tendo a massa, finalmente pode-se obter a aceleração da massa inercial (eq. 3), que por sua vez, é a mesma que o corpo ao qual está acoplado.

Figura 4 - Sistema esquemático do funcionamento do acelerômetro.



Fonte: HOSHINO e WAI (2014)

$$F=k.x \quad (1)$$

$$R=m.a \quad (2)$$

$$F=R \rightarrow k.x=m.a$$

$$\therefore a(x)= (k/m).x \quad (3)$$

Além de acelerômetros baseado em sistemas massa-mola, existem diversos outros tipos, dos quais pode-se destacar os extensômetros, os piezoresistivos, os piezoelétricos e os de capacitância variável. Todos os quatro citados anteriormente podem ser classificados como acelerômetros mecânicos. No entanto, recentemente, acelerômetros eletromecânicos têm surgido e substituído os acelerômetros mecânicos. Tais acelerômetros são chamados de MEMS (Micro Electronic Mechanical System). O motivo para tal domínio decorre do fato de eles serem menores, mais eficientes e mais baratos, apesar de utilizar o mesmo método dos acelerômetros mecânicos. A seguir, será feita uma breve descrição de cada tipo e qual o escolhido para o projeto.

a) Extensômetros

Os extensômetros são sensores que medem acelerações baseando-se na deformação do corpo ao qual ele está acoplado e a variação de resistência elétrica do extensômetro resultante de sua deformação. Devido a essa característica, ele é amplamente aplicado em ensaios de corpo de prova e monitoramentos.

b) Acelerômetros Piezoresistivos

Os acelerômetros piezoresistivos são sensores que vão medir a aceleração baseando-se na variação de resistência de uma massa devido a sua deformação no instante em que se aplica uma força sobre o objeto monitorado. Sua principal característica é ser capaz de fazer medições em baixas frequências.

c) Acelerômetros de Capacitância Variável

O acelerômetro de capacitância variável é composto por um componente elétrico capaz de armazenar a carga que se encontra entre duas placas paralelas entre si. Como $C = \epsilon A / d$, sendo ϵ uma constante que depende do material dos componentes, A , a área das placas e d a distância entre as placas e o componente que armazena carga e como ϵA não varia, a capacitância dependerá da distância d . Assim, o sensor correlaciona o movimento das placas d com a aceleração que ocasionou tal movimento nas placas.

d) Acelerômetros Piezoelétricos

Existem na natureza alguns elementos que, quando sofrem deformação em sua estrutura, geram corrente elétrica. Tal fenômeno é chamado de efeito piezoelétrico. Os acelerômetros utilizam esse efeito para obter as acelerações.

3.3.1.5 ESCOLHA PARA DETECÇÃO

O método mais adequado escolhido para se fazer a detecção de quedas foi o uso conjunto do giroscópio com o acelerômetro. Tal escolha se justificou pelo fato do acelerômetro ser capaz de captar acelerações com uma precisão de até $0,001 \text{ m/s}^2$ e em diferentes eixos, sendo a aceleração uma das principais características de uma queda. Já o uso do giroscópio foi interessante na medida em que ele consegue identificar rotações nos eixos, melhorando, assim, a precisão na detecção, visto que um corpo caído tende a ter o seu eixo alterado.

3.3.2 LOCALIZAÇÃO – AMBIENTES EXTERNOS

Especificou-se duas precisões diferentes para a localização (coordenadas geográficas) a ser obtida pelo sistema. Essa diferença decorre do fato de 70% das quedas de idosos ocorrerem dentro de seus domicílios (HOSPITAL ALBERT EINSTEIN, 2012). Ou seja, a maioria dos casos ocorre em um ambiente previamente conhecido: o domicílio no qual os idosos habitam. Além disso, essas quedas não são igualmente distribuídas entre os cômodos do domicílio: há ambientes onde o idoso está mais propenso a sofrer quedas sérias, como, por exemplo, o banheiro. Por isso, é importante conseguir identificar qual o cômodo no qual o idoso se encontra no momento da queda. Esses fatores justificam a necessidade de maior precisão no ambiente domiciliar.

Já no ambiente externo e em outros ambientes internos, diferentes do domiciliar, aceita-se uma precisão menor devido à menor ocorrência de incidentes, à possibilidade de se visualizar a pessoa de uma distância maior (em ambientes externos) e à maior circulação de pessoas nesses ambientes, normalmente. No

entanto, esses ambientes não podem ser desprezados, já que 30% dos casos ainda é uma parcela relevante.

Em ambientes externos, tal como ruas, calçadas e parques há uma grande disponibilidade de mapas que podem ser usados para auxiliar na obtenção das coordenadas geográficas. Já em ambientes internos, os mapas são normalmente privados e, portanto não acessíveis ao uso público.

Há diversas alternativas que podem ser utilizadas para se obter a localização geográfica de um objeto automaticamente. Essas alternativas serão descritas e avaliadas a seguir, considerando os requisitos do projeto (RAMBLINGS, 2012).

3.3.2.1 GPS

O GPS é um sistema de navegação por satélite desenvolvido para uso militar a partir da década de 60, e só foi disponibilizado para uso civil em 1996. Atualmente é utilizado em inúmeros casos, como aviões, navios até smartphones e carros.

Para obter o conjunto de variáveis formado por Altitude, Longitude, Latitude e Tempo é necessário que o receptor obtenha sinais de 4 ou mais satélites, dentre os 24 satélites que em órbita. Após a recepção do sinal utiliza-se métodos, como a trilateração, para calcular o valor das variáveis. (BRAIN e HARRIS, 2014)

a) Pontos Positivos:

- O sistema é gratuito, há apenas o custo de aquisição do receptor.
- Precisão: 10-75m.

b) Pontos Negativos:

- O sistema não funciona dentro de ambientes internos, já que há a necessidade de ter pelo menos 4 satélites em seu campo de visão. É possível realizar uma aproximação com 3 satélites “visíveis”, supondo o valor da altitude como o último valor medido, por exemplo.
- Tempo de localização. Normalmente demora 1 minuto, mas pode ser muito maior.

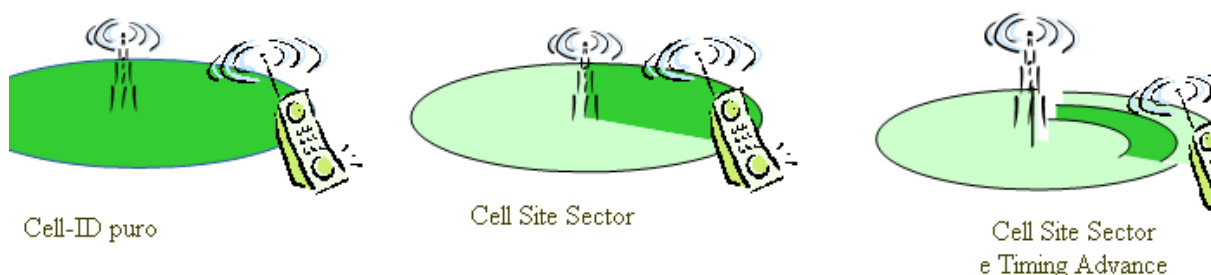
3.3.2.2 SINAL DE CELULAR

Quando o celular conecta-se à rede de telefonia, é possível obter o Cell ID e o Location Area Code (LAC) da célula à qual o celular está conectado, além de informações da operadora, Mobile Country Code (MCC) e Mobile Network Code (MNC). Este conjunto de informações é único e é atribuído a cada Base Transceiver Station (BTS), ou seja, a cada torre de telefonia móvel. Portanto, é possível descobrir as coordenadas geográficas da antena ao se buscar esse conjunto de informações em bancos de dados disponíveis na internet. A partir das coordenadas geográficas da torre e de outras informações, como o tempo de transmissão e a intensidade do sinal.

Há diversos algoritmos que podem ser utilizados para encontrar a localização a partir das variáveis fornecidas. Alguns desses métodos são (WIRELESSBR, 2014):

- Cell-ID: Busca-se apenas a coordenada da antena de telefonia móvel e se assume que o usuário se encontra dentro da área de alcance da antena.
- Cell-ID Timing Advance: Estima-se a que distância o dispositivo está da antena. Somando essa informação à localização da antena é possível estimar uma circunferência com a possível localização do usuário.
- Cell Site Sector: Neste método acrescenta-se mais uma informação aos inputs: o setor no qual o usuário se encontra. Um setor é uma fatia que abrange $\frac{1}{3}$ da área de alcance da antena.
- Enhanced Cell-ID Timing Advance: Combina-se diversos métodos de modo a tentar melhorar a precisão da localização.

Figura 5 – Representações de algoritmos de localização



Fonte: WIRELESSBR (2014)

a) Pontos Positivos:

- Extensa área de cobertura, equivalente à rede de telefonia móvel
- Possibilidade de uso em ambiente indoor

b) Pontos Negativos:

- Dificuldade de obter as coordenadas geográficas das antenas
 - Os maiores bancos de dados sobre antenas de telefonia são privados, e é preciso pagar para acessá-los. Exemplos: Location API (site: <http://location-api.com>) - \$100 por 2.000 acessos – e Unwired Labs (site: <http://unwiredlabs.com>) - \$145/mês para 30.000 acessos.
 - Há um grande banco de dados público, criado pelo upload de informações dos usuários, gerenciado pela OpenCellID. No entanto, as antenas de telefonia brasileira não são localizadas, já que os códigos não conferem (Operadoras testadas: Vivo e Claro)
 - A Anatel possui um banco de dados das antenas de telefonia do Brasil onde é possível obter as coordenadas geográficas da antena. No entanto, as antenas não estão identificadas pelo conjunto CellID, LAC, MCC e MNC. Elas são identificadas por um código próprio da Anatel.
 - O Google também possui um banco de dados utilizado por suas aplicações, como o GoogleMaps. Esse banco de dados não é disponibilizado publicamente, mas é possível acessá-lo através de um web service “secreto”. No entanto, o uso desse web service não é documentado publicamente e deveria ser restrito somente ao uso de aplicações Google, sendo disponibilizado, por exemplo, em bibliotecas do Android (TYLERBRINKS, 2014).
- Baixa precisão. A precisão fornecida por este método varia muito conforme o ambiente, podendo ser de algumas dezenas de metros a até centenas de metros, em ambientes urbanos.

O grande ponto negativo destacado foi a falta de uma solução eficaz para a obtenção das coordenadas geográficas das antenas de celular. No entanto, o Google disponibiliza comercialmente uma API (Google Maps Geolocation API) que

fornece a localização de um dispositivo/pessoa a partir das variáveis da antena de celular MCC, MNC, LAC, CellID e potência do sinal e de sinais de wi-fi captados. A desvantagem da API é que ela limita a 100 consultas ao servidor a cada 24 horas. Caso haja a necessidade de mais acessos é necessária a compra de uma licença para negócios. Mas, no escopo deste projeto, apenas para fins de teste, o limite de 100 consultas é razoável (GOOGLE DEVELOPERS, 2014).

3.3.2.3 WI-FI

O receptor de wi-fi obtém algumas informações das diversas redes das quais ele consegue captar o sinal. Estas informações são, por exemplo, a intensidade do sinal, o SSID e o MAC Address do Access Point. A partir dessas informações é possível mapear ambientes através de *fingerprints* para posteriormente localizar um dispositivo nesse ambiente. A solução citada anteriormente, do Google Maps Geolocation API, também consegue receber como inputs sinais de wi-fi de modo a obter melhor precisão na localização.

O sinal de wi-fi de 802.11 WLANs é detectado e pode ser mapeado e armazenado em bancos de dados para posterior consulta, a partir da qual será possível estimar a localização do usuário (WEXLER, 2006).

a) Pontos Positivos:

- Redes wi-fi são amplamente difundidas atualmente, sendo detectadas tanto em ambientes indoor quanto outdoor.
- A API do Google citada anteriormente pode utilizar os dados de redes wi-fi como input para melhorar a precisão da localização .

b) Pontos Negativos:

- Precisão variável: O sinal é fortemente influenciado por diversos tipos de interferência, como paredes ou outros sinais.
- É necessário um mapeamento prévio dos sinais de wi-fi para seu uso na localização (VAUGHAN-NICHOLS, 2011).

3.3.2.4 DEAD RECKONING

Dead Reckoning é o processo de se estimar a localização atual a partir de informações de uma localização anterior e um deslocamento.

Deve-se obter uma localização inicial a partir da qual serão calculadas as próximas localizações. Acrescenta-se a essa localização inicial um deslocamento, medido por giroscópios e acelerômetros, obtendo-se uma localização final. Esse processo é realizado inúmeras vezes de modo que se conheça a posição do usuário a cada instante (HEATHCOTE, 2004).

a) Pontos Positivos:

- Em teoria, não há a necessidade de múltiplos acessórios, basta uma localização exata e um acelerômetro/giroscópio. Poder-se-ia, por exemplo, sempre iniciar o dispositivo na mesma posição geográfica e o acelerômetro/giroscópio serviriam de input para o cálculo do deslocamento.

b) Pontos Negativos:

- Localização inicial deve ser muito precisa não inviabilizar o método.
- Neste método, os erros na medida do deslocamento se acumulam, até que os erros se tornam tão grandes a ponto de inutilizar a localização calculada. Os acelerômetros e giroscópios comumente utilizados possuem precisão da ordem de $\pm 2g$ (acelerômetros), precisão está ainda muito baixa para o uso do Dead Reckoning.

3.3.2.5 OUTRAS SOLUÇÕES

As soluções apresentadas anteriormente foram as que se apresentaram mais promissoras para serem implementadas no projeto. Outras soluções possíveis para resolver a localização do usuário são o RFID ou o Bluetooth, que são eliminadas devido ao baixo alcance, inviabilizando seu uso em ambientes externos.

Também é importante não descartar o uso de soluções híbridas, como GPS+GSM ou GSM+wi-fi, ou até mesmo GSM+mapa. O uso de um mapa pré-carregado

possibilita que se façam estimativas de locais possíveis para o usuário. Assim é possível melhorar a precisão do sistema ao se eliminar posições improváveis no mapa (HEATHCOTE, 2004).

3.3.2.6 ESCOLHA PARA LOCALIZAÇÃO EXTERNA

A partir das soluções pesquisadas e descritas acima, optou-se por utilizar o Google Maps Geolocation API, em sua licença de testes, com limite de 100 consultas/24 horas. Esta decisão é embasada em diversos argumentos:

Com a API não há a necessidade de implementação de algoritmos de triangularização ou outros de localização, de tal maneira que o desenvolvimento seja simplificado.

O GPS não funcionaria em ambientes internos e sua não-utilização reduz o total de componentes que compõem o dispositivo, reduzindo, assim, a complexidade do sistema e o consumo de bateria. Além disso, o sinal GSM possui grande cobertura. Segundo a ANATEL todos os municípios brasileiros possuem sinal de celular e operadoras como a Vivo atendem a mais de 91% da população (TELECO, 2014).

3.3.3 LOCALIZAÇÃO – AMBIENTE DOMICILIAR

Em certos ambientes internos a indicação precisa do local da queda pode prover informações importantes sobre o contexto no qual esta ocorreu. 70% das quedas de idosos ocorrem dentro de casa, e certos cômodos, como banheiros, oferecem maiores riscos que outros (HOSPITAL ALBERT EINSTEIN, 2012). Assim, a possibilidade de precisar o local da queda pode fornecer indicativos sobre a gravidade da queda, sobre a segurança nos cômodos da casa e, principalmente, se a queda ocorreu dentro do domicílio e não, por exemplo, em um domicílio vizinho.

Há grandes investimentos e avanços sendo realizados em sistemas de localização interna, conhecidos como Indoor Positioning System (IPS). Um IPS é uma rede de dispositivos usada para localizar objetos dentro de um ambiente fechado, como um prédio ou uma casa. A localização pode ser feita através de diversas tecnologias, ou

até mesmo soluções híbridas, mas em qualquer caso há a necessidade de um mapa preciso do ambiente.

O funcionamento desse sistema depende de emissores de sinais e de receptores destes sinais. A localização pode ser obtida através de sensores de proximidade, usando, por exemplo, RFID, ou criando-se mapas com os padrões do sinal em cada posição (*fingerprints*) (SCHUTZBERG, 2013).

3.3.3.1 GPS

Como mencionado na seção anterior o GPS não é adequado à ambientes internos devido à baixa potência do sinal, à necessidade de ter satélites em seu “campo de visão”, a interferências e reflexões no sinal causadas por paredes, telhados e outros objetos.

3.3.3.2 SINAL DE CELULAR

Como explicado na seção anterior o sinal de celular pode ser usado em ambientes internos, mas é de baixa precisão, ou seja, não é adequado para a obtenção de localizações precisas dentro de ambientes internos. Como mostrado na seção anterior, o dispositivo obterá uma localização aproximada através do sinal de celular, podendo esta localização servir de base para um refinamento proporcionado pelas soluções a serem apresentadas a seguir.

3.3.3.3 WI-FI

A tecnologia Wi-Fi foi mencionada para localização de baixa precisão, no entanto, há diversas empresas oferecendo soluções de localização interna com precisão na ordem metros. Uma empresa que tem investido nessa solução é o Google, que busca mapear e cadastrar todas as redes wi-fi no Google Location. Outras empresas líderes deste segmento são Navizon e Skyhook.

Para localizar o dispositivo considera-se os Wi-Fi Hotspots ou Wireless Access Points, identificados, normalmente, pelo SSID e MAC Address, e um mapa com os *fingerprints*.

a) Pontos positivos:

- Investimentos nessa área que demonstram o potencial desta solução.
- Popularidade do Wi-Fi em ambientes domésticos. Um indicativo disso é que atualmente os novos sistemas digitais de televisão já possuem Wi-Fi embutido

B) Pontos negativos:

- Variações e interferências no sinal, causadas por paredes, móveis e outros fontes de sinais eletromagnéticos.
- Necessidade de mapeamento prévio para a criação dos *fingerprints*.

3.3.3.4 BLUETOOTH

O IPS baseado em Bluetooth utiliza o mesmo conceito do Wi-Fi, criando *fingerprints* conforme a intensidade do sinal Bluetooth. A NOKIA tem investido no desenvolvimento de soluções de localização baseadas nesta tecnologia. A diferença para o Wi-Fi é que o alcance do sinal Bluetooth é menor, cobrindo apenas alguns metros quadrados, o que gera uma maior necessidade por emissores de sinal (Access Points), mas que aumenta a precisão do sistema.

3.3.3.5 OUTRAS SOLUÇÕES

Além de soluções que se baseiam no mapeamento de *fingerprints*, como Wi-Fi ou Bluetooth, há soluções baseadas em *Dead Reckoning* ou em proximidade (RFID, NFC). Mesmo nas soluções baseadas em *fingerprints* há outras maneiras de se fazer o mapeamento. Pode-se, por exemplo, mapear padrões de luz emitidos por LEDs programados (BYTELIGHT, 2014), mapear padrões de campo magnético, ou mesmo outros padrões de sinal de rádio como o ZigBee.

Como mencionado anteriormente, *Dead Reckoning* é inviável devido a alta precisão necessária dos sensores de movimento do dispositivo. Para as soluções baseadas em proximidade as desvantagens são a própria necessidade de uma pequena distância entre emissor e receptor e a necessidade de emissores específicos, não encontrados normalmente em ambiente domiciliares.

E as outras soluções apresentadas tem como desvantagens a necessidade de mapeamento prévio, típico de soluções por *fingerprint*, e/ou a necessidade de equipamentos específicos.

3.3.3.6 ESCOLHA PARA LOCALIZAÇÃO INTERNA

As soluções presentes atualmente para a localização interna de precisão necessitam de um mapa do ambiente e de algum trabalho prévio no local, seja ele a instalação de dispositivos e/ou a realização de uma série de medições. Essa necessidade de trabalho prévio prejudica a adoção de qualquer solução para a localização precisa em ambientes internos. No entanto, devido à complexidade de implementar uma solução para esse tipo de localização frente ao escopo deste projeto, optou-se por não implementar esta funcionalidade, sendo válida sua inclusão em trabalhos futuros.

3.3.4 BATERIA

Para a escolha do tipo de bateria considerou-se os requisitos funcionais de peso e tamanho, além da necessidade de ser recarregável. Há diversos tipos de bateria que serão descritos brevemente a seguir. Para auxiliar a decisão utilizou-se o dado de densidade de potência das baterias, que indica quanto de energia é fornecida por Kg (ADAFRUIT, 2013).

3.3.4.1 ALCALINA

Baterias e pilhas alcalinas são muito comuns e embora não recarregáveis são fáceis de se encontrar no mercado. Possuem uma densidade de potência de 100Wh/kg.

3.3.4.2 NI-CAD E NI-MH

Estas baterias são recarregáveis e podem seguir os padrões de tamanho das alcalinas, podendo substituí-las. Elas sofrem com “efeito memória”, de modo que parece que as baterias passam a possuir menor capacidade com o tempo. A densidade destas baterias está entre 60Wh/kg e 100Wh/kg.

3.3.4.3 LI-ION E LI-PO

Estes dois tipos de baterias são o padrão tecnológico atual, sendo utilizadas em notebooks e celulares. Ambas utilizam o Lítio como seu elemento base, sendo ele o metal mais leve conhecido. Também são recarregáveis, não sofrem de “efeito memória”, possuem desempenho energético e ciclo de vida similares.

Dentre as diferenças destaca-se que a bateria de Li-Po é feita com polímeros, que são materiais maleáveis, desse modo a bateria pode ter um formato diferente do cilíndrico das Li-Ion. Além disso as células de Li-Po são envolvidas por um material flexível e mais leve do que o material rígido que encobre as células de Li-ion, fazendo com que as baterias de Li-Po sejam mais leves do que as de Li-Ion.

Em relação à densidade energética a bateria de Li-Po apresenta melhor densidade, de tal forma que essas baterias podem ser mais finas que as de Li-Ion, oferecendo a mesma carga. Já em relação ao quesito preço a bateria de Li-Ion é mais barata do que a de Li-Po (CIRIACO, 2013).

3.3.4.4 OUTRAS

Há diversas outras opções de baterias como as baterias seladas que são recarregáveis, baratas, mas grandes e pesadas. Também há outras baterias não-recarregáveis, como as utilizadas em relógios de pulso. De qualquer forma, essas outras opções não atendem os requisitos citados no início desta seção.

3.3.4.5 ESCOLHA DA BATERIA

Independente da bateria deve-se tomar precauções em relação à segurança no manuseio e uso delas, portanto não consideramos este como um fator importante na escolha.

Dentre os tipos de bateria apresentados o que mais se adequa a este projeto é a bateria de Li-Po, já que ela é (i) fina, (ii) leve, (iii) recarregável e (iv) possui boa capacidade energética para este projeto - pode-se encontrar baterias de diversas capacidade como 1300mAh, 2600mAh, 6600mAh ou outras. Outra vantagem que será mais clara adiante é a tensão de saída. As mais comuns são as de 3,7-4,2V, sendo ideais para trabalhar com circuitos como os que serão utilizados no projeto (ADAFRUIT, 2013).

3.3.5 PERCEPÇÃO DE USO

Para o dispositivo, seria muito importante que se tivesse alguma maneira de se apenas fazer as detecções de queda e localização nos casos em que o dispositivo estivesse em uso. Isso para que gerações de falso-alertas por conta da queda do dispositivo sozinho não fossem geradas e também por conta da economia de bateria, visto que o dispositivo apenas funcionaria quando em uso.

3.3.5.1 ESCOLHA DO MÉTODO DE PERCEPÇÃO DE USO

Para atender a essa funcionalidade, optou-se por um método físico, por ser mais simples e mais confiável de se fazer. O método adotado foi através da utilização de dois fios conectores que obrigatoriamente deveriam ser ligados um no outro quando o dispositivo estivesse em uso. Assim, a partir dessa conexão, seria possível gerar uma corrente elétrica que acionaria o dispositivo, indicando que o mesmo encontrava-se em uso.

Por questões de facilidade em se fazer demonstrações e testes do dispositivo desenvolvido, tal funcionalidade não foi implementada no projeto. No entanto, é importante notar que ela é viável.

3.3.6 ALERTAS

Conforme especificado anteriormente, o dispositivo deve enviar um alerta diretamente aos celulares cadastrados, informando a ocorrência da queda, a localização e o horário da queda ou informando o nível de bateria do dispositivo. De forma a atender essa funcionalidade analisou-se as possibilidades de comunicação com celulares.

A solução deve poder ser utilizada com a maior parte dos celulares/smartphones e deve possibilitar que o alerta seja entregue em menos de 5 minutos.

3.3.6.1 INTERNET MÓVEL

Com a adoção dos smartphones o uso da internet móvel tornou-se muito mais difundida. Segunda pesquisa realizada pelo Google em 2012, 42% dos usuários acessam a internet em seus smartphones ao menos uma vez por dia. Ao mesmo tempo em que este número mostra a adesão dos usuários à internet, ele mostra que o uso da internet ainda não é constante, seja pela falta de necessidade desse uso, pelo consumo extra de bateria ou pelos custos extras. Para o envio de alerta, portanto, é improvável que todo alerta seja entregue em menos de 5 minutos já que depende da conectividade do celular com a internet (GOOGLE INC., 2012).

Como vantagem à esta solução, através da internet ou de aplicativos que utilizam a internet, é possível fornecer ao usuário um conteúdo rico em informações. No caso deste projeto, por exemplo, a localização da queda poderia ser mostrada em um mapa. Ou seja, a conexão com a internet permitiria a transmissão de mais dados do dispositivo ao celular (SANDOVAL, 2014).

3.3.6.2 MENSAGEM DE TEXTO (SMS)

O serviço de envio de mensagens de texto popularizou-se com o celular e, mesmo apresentando certo declínio com advento de serviços de mensagens pela internet, ainda está presente em todos os celulares e smartphones. Assim, esta solução funcionaria com qualquer celular do mercado, basta que este esteja conectado à

rede de telefonia móvel. Além disso, a Anatel publicou em 2011 uma resolução que estabelece parâmetros de qualidade para o envio de mensagens de celular, exigindo que um SMS não poderá demorar mais que 60 segundos para ser entregue, em 95% dos casos (ANATEL, 2011).

Como desvantagem à esta tecnologia está o limite de 160 caracteres alfanuméricos por mensagem, restringindo a quantidade e complexidade das informações que podem ser enviadas. Esta limitação pode ser contornada com o envio de múltiplas mensagens, mas de qualquer jeito a apresentação do conteúdo ainda ocorrerá por texto.

3.3.6.3 USSD

O USSD (Unstructured Supplementary Service Data) é um protocolo usado em redes GSM orientado a conexão. Um exemplo de USSD é representado abaixo:

Figura 6 - Exemplo de serviço USSD



Fonte: SSCI (2014)

Este serviço permite o envio rápido de curtas mensagens e de interação com o usuário. No entanto, as mensagens não são armazenadas, e caso ocorra algum erro no envio da mensagem, esta será simplesmente perdida, diferentemente do SMS, por exemplo, que é baseado em “store & forward”. Outra característica é que não é possível realizar envio de mensagens de celular para celular, de modo que não seria

possível enviar a mensagem do dispositivo até o celular do cuidador, devendo a mensagem passar por uma central (FRIZZO, 2014).

3.3.6.4 OUTROS

Há outras formas de comunicação permitidas pelos celulares como, por exemplo, Bluetooth e NFC, mas que são de curto alcance e, portanto inviáveis para este projeto. Outra solução seria o uso de mensagens áudio que seriam transmitidas através de ligação, no entanto, há como desvantagens para esta solução a impossibilidade de armazenamento da ligação, a menos que esta seja armazenada no correio de voz, e a complexidade em criar a mensagem de voz, uma vez que este projeto busca criar um dispositivo de usabilidade simples e de baixo custo.

3.3.6.5 ESCOLHA DO ALERTA

Das soluções apresentadas o uso de internet e de SMS se mostraram mais viáveis, cada um com suas qualidades e obstáculos. A internet proporcionaria a transmissão de uma maior quantidade de informações e a apresentação mais amigável destas. Enquanto que o SMS garantiria a entrega rápida do alerta em um maior número de casos.

Ambas as tecnologias podem ser implementadas com a solução escolhida anteriormente para a localização, ou seja, o uso do GSM, já que a maior parte dos módulos GSM desenvolvidos atualmente permite também a conexão com a internet. No entanto, uso da internet dependeria de uma conexão EDGE/3G/4G tanto no dispositivo quanto no celular do cuidador, além de um web service que possibilite a transmissão do alerta. Enquanto que o uso do SMS dependeria apenas da conexão com a rede de telefonia móvel.

Desse modo, a solução escolhida foi o uso do SMS para o envio de alertas do dispositivo aos celulares cadastrados, pois (i) possibilita envio rápido de mensagens, regulado pela Anatel, (ii) necessita apenas da cobertura de telefonia móvel e (iii) o envio é feito diretamente entre dispositivo e celular (GOOGLE INC., 2012).

3.3.7 CADASTRO DE TELEFONES

Como descrito nos requisitos funcionais o sistema deve ser capaz de receber pedidos de cadastro de telefone. De modo a atender este requisito considerou-se diversas soluções que permitam uma interação com o sistema de modo a realizar o cadastro.

3.3.7.1 SMS

Para o envio de alertas optou-se pela solução baseada em SMS. Esta tecnologia também pode ser utilizada para configurar o dispositivo, já que este terá um módulo GSM que precisa de um SIM card com número de telefone válido para funcionar. Esta solução já foi utilizada por outros sistemas (WEN-TSAI SUNG, 2013). Neste método seria possível enviar uma mensagem com os números de telefone a serem cadastrados no sistema ou enviar uma mensagem de cadastro a partir de cada telefone a ser cadastrado.

Esta é uma solução rápida e simples, mas que depende de algum tipo de formato específico para a mensagem de cadastro. Assim, a interface com o usuário pode ser prejudicada caso o formato da mensagem não seja bem estruturado.

3.3.7.2 USSD

Esta tecnologia também foi enunciada na seção anterior. A maior desvantagem do USSD, no caso deste projeto, é o fato de não permitir uma conexão direta entre o dispositivo e o celular.

3.3.7.3 BLUETOOTH

A tecnologia Bluetooth é de baixo alcance (aproximadamente de 1 a 10 metros), de modo que exige a proximidade física entre o sistema e o celular. Além disso, nem todos os celulares mais antigos possuem Bluetooth e seria necessária a utilização de algum sistema para contornar o pareamento dos dispositivos. A vantagem do

Bluetooth se concentra em sua conexão rápida, podendo chegar a 24Mbit/s, fator que não seria necessário no simples cadastro de números de telefone (ALECRIM, 2013).

3.3.7.4 INTERNET

Seria possível a utilização da internet para transmitir o número do telefone a ser cadastrado ao sistema. Esta solução dependeria de uma conexão à internet, que poderia ser garantida pela internet móvel provida pelo módulo GSM do dispositivo, e como seria uma etapa de configuração pode-se assumir que o dispositivo esteja em um ambiente controlado com acesso à internet.

O envio do pedido de cadastro poderia ser feito através de um site ou aplicativo, de modo que a interface com o usuário pode ser trabalhada de modo a ser mais simples e bem estruturada.

Como desvantagem deste sistema apresenta-se a necessidade de utilização de um web service para intermediar a comunicação e a criação de uma interface extra com o usuário (site ou aplicativo).

3.3.7.5 INTERFACE FÍSICA

Outra solução seria a inserção no dispositivo de elementos físicos que possam interfacear com o usuário, de modo a permitir o cadastro de telefones. Esses elementos podem ser dos mais diversos:

- Botões e visor gráfico
- Botões e alto-falante para ditar os números cadastrados, de modo que o usuário possa conferi-los
- Apenas botões, mas neste caso o usuário teria que fazer algum teste para verificar que o número foi corretamente cadastrado
- Microfone e alto-falante, de modo que o cadastro seja todo feito por comandos de voz. Esta solução exigiria maior capacidade de processamento e de um sistema de reconhecimento de voz.

De maneira geral, a utilização de qualquer uma dessas soluções implicaria na necessidade de mais componentes no dispositivo, além de algumas não permitirem uma interface amigável com o usuário.

3.3.7.6 OUTROS

Outra solução que pode ser apresentada neste caso é o cadastro do telefone em código (hardcoded). No entanto esta opção possui diversas desvantagens como o fato de ser cara, já que implica em personalizar o software de cada dispositivo feito, não ser possível alterar o número de telefone sem ter que mexer diretamente no código e usar uma interface com o usuário de difícil utilização.

3.3.7.7 ESCOLHA DO CADASTRO DE TELEFONES

A solução escolhida, e implementada, é o uso de mensagens SMS para o cadastro de números de telefone. Em uma versão futura do sistema é possível que a interface web/aplicativo torne-se mais atraente, mas no estágio atual do projeto o SMS mostrou-se melhor já que (i) não exige um novo hardware, utilizando o mesmo componente da localização e do envio de alertas, (ii) utiliza a mesma tecnologia do envio de alertas, favorecendo a curva de aprendizado e (iii) permite a configuração dos números cadastrados a qualquer momento.

3.3.8 INTERFACE PARA ALERTAS E LOCALIZAÇÃO

Para o cadastro de número de celular e para o recebimento de alertas, é possível a utilização de qualquer celular, já que ambos utilizam SMS. No entanto, para a visualização da localização da queda e para obter uma notificação mais visível do alerta, optou-se pela criação de um aplicativo no celular.

Esse aplicativo foi desenvolvido apenas para smartphones por diversas razões: (i) maior facilidade de desenvolvimento, (ii) facilidade de visualização de mapas e (iii) mais de 75% dos celulares vendidos no Brasil atualmente são smartphones (G1, 2014).

Devido a restrições de tempo, e por não se tratar do foco deste projeto, o aplicativo foi desenvolvido para apenas uma plataforma. Esta plataforma foi escolhida dentre as três principais atualmente: Android, iOS e Windows Phone.

Para a escolha da plataforma considerou-se os fatores e características mostrados na tabela a seguir:

Tabela 1 - Comparativo de plataformas para smartphones

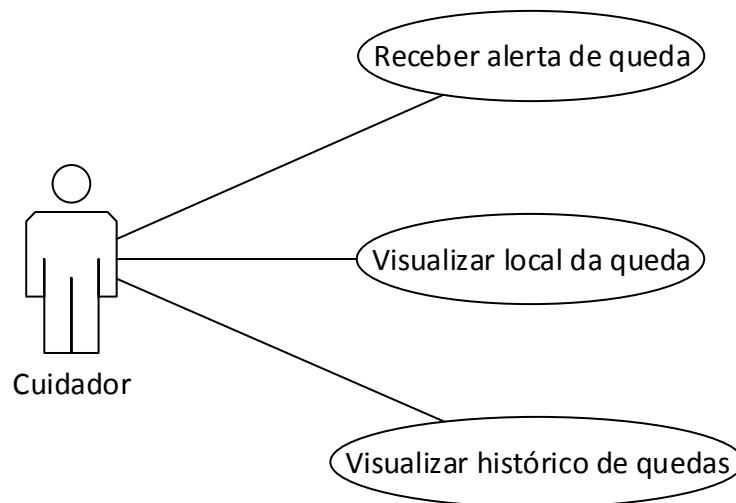
	Android	iOS	Windows Phone
Empresa	Google	Apple	Microsoft
Market Share (EUA)	78,9%	14,9%	3,9%
Número de tutoriais (busca no Google)	51 milhões de links	33 milhões de links	31 milhões de links
Variedade de celulares	Maior variedade de celulares e fabricantes	Apenas iPhone	Seleção crescente de celulares, mas majoritariamente fabricados pela Nokia
Outros	A API de localização é do Google, de modo que a integração é facilitada	Dificuldade de desenvolvimento – Falta de acesso a iPhones e computadores da Apple	Falta de acesso a celulares com Windows Phone

Fonte: G1 (2014) e BAMBURIC (2014)

Considerando as informações apresentadas na tabela, optou-se pela utilização da plataforma Android, principalmente devido a (i) facilidade de desenvolvimento, (ii) quantidade de tutoriais e (iii) *market share*.

3.3.8.1 CASOS DE USO – APLICATIVO

Tomando como base o modelo UML, foi feito um diagrama de casos de uso para o aplicativo. O diagrama se encontra na Figura 7.

Figura 7 - Casos de uso - aplicativo

Fonte: HOSHINO e WAI (2014)

Caso de uso 1: Receber alerta de queda

Descrição: Este caso de uso descreve o processo de recebimento do SMS que alerta a ocorrência de um evento de queda

Evento iniciador: Um SMS de alerta de queda é recebido pelo smartphone

Atores: Cuidador

Pré-condição: Smartphone ligado

Cenário:

- O aplicativo recebe o SMS e notifica o cuidador sobre a queda
- O cuidador vê/abre a notificação

Pós-condição: Cuidador notificado sobre a queda

Caso de uso 2: Visualizar local de queda

Descrição: Este caso de uso faz o processo de visualização do local de queda

Evento iniciador: O aplicativo é iniciado

Atores: Cuidador

Pré-condição: Smartphone ligado e com acesso à internet

Cenário:

- O aplicativo mostra em uma mapa as coordenadas do locais de queda
- O cuidador pode selecionar algum local
- O aplicativo mostra informações sobre o local selecionado

Pós-condição: Mapa com os locais de queda

Caso de uso 3: Visualizar histórico de quedas

Descrição: Este caso de uso descreve o processo de visualização do histórico de quedas

Evento iniciador: Cuidador seleciona opção de histórico

Atores: Cuidador

Pré-condição: Aplicativo rodando

Cenário:

- O aplicativo mostra uma lista com todas as ocorrências de queda registradas

Pós-condição: Lista com as quedas

3.3.8.2 REQUISITOS FUNCIONAIS - APLICATIVO

No.: RF-App-01 – Identificação de SMS

Descrição: O aplicativo é capaz de identificar, dentre todos os SMS recebidos pelo smartphone, quais são referentes a alertas de queda.

No.: RF-App-02 – Obtenção das coordenadas geográficas da queda

Descrição: O aplicativo é capaz de obter as coordenadas geográficas, latitude e longitude, e também a precisão dessas coordenadas, da queda, a partir dos dados contidos no SMS de alerta de queda.

No.: RF-App-03 – Visualização do local da queda

Descrição: O aplicativo é capaz de indicar o local da queda através de um marcador no mapa. Além disso, a precisão das coordenadas deve ser mostrada como um círculo, de raio igual à precisão, em volta do marcador.

No.: RF-App-04 – Visualização do histórico de quedas

Descrição: O aplicativo é capaz de mostrar um histórico com todos os registros de queda, indicando a data da queda e o número do dispositivo que gerou a queda (número do SIM card).

No.: RF-App-05 – Visualização de informações da queda

Descrição: O aplicativo mostra junto ao marcador do local da queda algumas informações sobre a queda: data e hora do recebimento do SMS e número do dispositivo (número do SIM card).

No.: RF-App-06 – Notificação ao usuário sobre evento de queda

Descrição: O aplicativo mostra uma notificação ao usuário informando sobre a queda, após o recebimento do SMS de alerta de queda.

3.3.8.3 REQUISITOS NÃO-FUNCIONAIS - APLICATIVO

No.: RNF-App-01 – Utilização da plataforma Android

Descrição: O aplicativo deve ser desenvolvido para rodar em dispositivos Android.

No.: RNF-App-02 – Utilização da Geolocation API

Descrição: O aplicativo deve utilizar a Geolocation API para obter as coordenadas geográficas e a precisão do local da queda.

No.: RNF-App-03 – Visualização das últimas 5 quedas

Descrição: O aplicativo deve mostrar no mapa somente as últimas 5 quedas registradas.

No.: RNF-App-04 – Histórico deve conter todos os registros de queda

Descrição: O aplicativo deve mostrar no histórico todos os registros de queda recebidos pelo smartphone, desde que estes registros não sejam apagados da memória do smartphone.

3.3.9 PROCESSADOR

Para o desenvolvimento deste projeto, como se trata de um protótipo inicial, focou-se em tecnologias consagradas para o estilo DIY (*Do it yourself*), representada majoritariamente pela plataforma de desenvolvimento Arduino. A grande limitação

apresentada pelo Arduino é o fato de se tratar principalmente de um microcontrolador, apresentando, portanto, baixa capacidade de processamento e pouca memória. Como opções para contornar essas limitações pesquisou-se também as plataformas Raspberry PI e BeagleBoard, que utilizam o conceito de *System-on-a-chip*.

Essas duas outras plataformas são bem diferentes do Arduino por se tratarem de pequenos computadores. Elas se baseiam em microprocessadores e não em microcontroladores. Nessas duas plataformas a capacidade de processamento é muito maior e elas são, basicamente, pequenos computadores, que rodam sistemas operacionais como Linux. Consideraram-se os modelos mais recentes e de tamanhos e preços comparáveis de cada plataforma.

Para analisar essas três plataformas consideraram-se quesitos como preço, disponibilidade de material para aprendizado e desenvolvimento, facilidade de uso, tamanho, peso, disponibilidade de acessórios e módulos complementares.

A existência de módulos para a plataforma é de grande importância já que nenhuma das plataformas apresentava todos os componentes a serem utilizados no projeto:

- Acelerômetro
- GSM

A existência de módulos (ou *shields*) que se comunicassem com a plataforma é essencial para simplificar o projeto e permitir seu rápido desenvolvimento.

A seguir serão comparadas as três plataformas.

3.3.9.1 ARDUINO

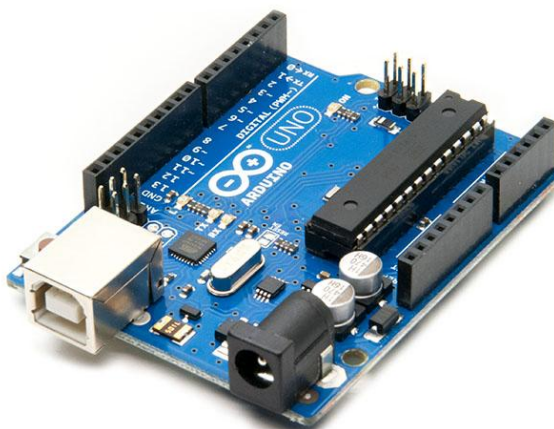
Arduino é um microcontrolador montado em uma placa única baseada, normalmente, em um microcontrolador Atmel AVR de 8-bits. É uma plataforma open-source, de modo que projetos desenvolvidos no Arduino podem ser migrados para projetos de específicos, de grande escala, caso necessário.

Atualmente há diversas placas Arduino que variam em quesitos como tamanho, capacidade e preço. Para este projeto considerou-se que o Arduino Uno, e outras placas de mesmo tamanho, como o Yùn, seriam suficientemente pequenas para

serem utilizadas, já que possuem aproximadamente o tamanho de um cartão de crédito.

A grande vantagem do Arduino é sua popularidade. Somente no fórum oficial do Arduino há mais de 200 mil membros e 1,5 milhão de postagens em 200 mil tópicos. E até 2013, mais de 700 mil placas oficiais haviam sido comercializadas (ARDUINO, 2014).

Figura 8 - Arduino Uno v3



Fonte: ARDUINO (2014)

3.3.9.2 RASPBERRY PI

O Raspberry Pi é um computador de baixo custo desenvolvido por uma fundação educacional no Reino Unido com o objetivo de promover a educação relacionada à computação. Como um computador ele pode ser conectado a um monitor, teclado ou mouse e roda sistemas operacionais baseados em Linux.

O Raspberry Pi é formado por um processador ARM de 700 MHz, 512 MB de memória RAM e uma unidade de processamento gráfico, possuindo uma capacidade de processamento muito superior ao Arduino Uno.

Em relação ao tamanho da comunidade ativa percebe-se que esta é menor que a do Arduino. O fórum oficial do Raspberry Pi possui pouco menos de 100 mil membros e mais de 500 mil postagens em mais de 70 mil tópicos (RASPBERRY PI FOUNDATION, 2014).

Figura 9 - Raspberry Pi Modelo B

Fonte: RASPBERRY PI FOUNDATION (2014)

3.3.9.3 BEAGLEBOARD

O BeagleBoard também é um computador de baixo custo desenvolvido por uma parceria entre Texas Instruments e Digi-Key. Tal como o Arduino ele é de hardware livre. Mas em termos de processamento assemelha-se ao Raspberry Pi, com um processador Sitara ARM de 1GHz e 512 MB de Ram. Ele também roda sistemas operacionais baseados em Linux.

Em relação ao tamanho da comunidade o BeagleBoard apresenta a menor delas, possuindo menos de 16 mil tópicos no fórum oficial, acessado em 3 de junho de 2014 (BEAGLEBOARD FOUNDATION, 2014).

Figura 10 - BeagleBone Black

Fonte: BEAGLEBOARD FOUNDATION (2014)

3.3.9.4 OUTROS

Outra opção que foi cogitada seria a compra de um processador de empresas como Texas Instruments ou STMicroelectronics e a criação de todo o circuito para utilizá-lo. Mas isso requereria uma grande alocação de tempo e esforço apenas para o desenvolvimento e teste do circuito. Além disso seria necessário garantir que cada módulo se comunicasse com o circuito desenvolvido, já que este módulo poderia ser comprado pronto ou criado também.

Outra opção que não envolveria o desenvolvimento de todo o circuito seria o uso de uma placa de desenvolvimento fornecida por essas empresas. No entanto, essa seria uma solução como a de utilizar uma das plataformas descritas anteriormente, mas sem a mesma disponibilidade de material, exemplos e acessórios, além dessas placas de desenvolvimento serem normalmente mais caras.

Deste modo, o desenvolvimento de todo o circuito a partir de um microprocessador foi desconsiderado para este projeto, mas continuará sendo uma opção para desenvolvimentos futuros, já que este desenvolvimento possibilita uma minimização do circuito, tornando-o mais simples, mais leve e sem componentes desnecessários.

3.3.9.5 ESCOLHA DO PROCESSADOR

Criou-se um comparativo para auxiliar na escolha do processador a ser utilizado.

Tabela 2 - Comparativo de processadores

Fator	Arduino	Raspberry Pi Modelo B	BeagleBone Black
Preço (US\$)	25-65	35	45
Tamanho (mm)	68,6 x 53,3	85,60 x 56	86,40 x 53,3
Peso	28g-41g	45g	40g
Disponibilidade de documentação e exemplos	Alta	Intermediária	Pequena
Comunidade desenvolvedora	Grande	Intermediária	Pequena
Processador	ATmega328P (8-bit)	ARM1176JZF-S core (32-bit)	TI OMAP3530
Clock	16MHz	700 MHz	1GHz
Memória (RAM)	2 kB – 2,5kB	512 MB	512 MB
Disponibilidade de acessórios e módulos	Alta	Média	Média
Hardware Open-source	Sim	Não	Sim
Análise	Capacidade de processamento muito baixa	Processamento acima do necessário e não é open-source	Processamento acima do necessário, falta de documentação, tutoriais e acessórios

Fonte: HOSHINO e WAI (2014) e ALLAN (2013)

Levando-se em conta as necessidades do projeto optou-se pelo Arduino, pois ele apresenta (i) comunidade ativa, (ii) alta disponibilidade e diversidade de acessórios e módulos, (iii) boa capacidade de processamento e (iv) hardware open-source. Sua menor capacidade de processamento não é um impeditivo ao projeto à medida em que este conta com um escopo limitado e que não depende do processamento de muitos dados, como imagens, vídeos, etc.

3.4 TESTES E SIMULAÇÕES

Alguns testes foram realizados para corroborar as escolhas apresentadas, antes de se iniciar o processo de desenvolvimento do projeto. A Geolocation API do Google, escolhida para a obtenção das coordenadas geográficas do evento de queda, foi testada nesta etapa.

3.4.1 GEOLOCATION API

A Geolocation API do Google, utilizada para a obtenção das coordenadas geográficas do dispositivo a partir dos dados das antenas foi testada separadamente para verificar sua adequação ao Projeto.

a) Teste do uso da API

Para poder utilizar esta API é necessário cadastrar uma conta para débitos, que dependem da quantidade de demandas realizadas ao servidor do Google. Os objetivos deste teste foram: (i) verificar que a chave de registro obtida no cadastro era válida para o uso da API, (ii) verificar se as 100 demandas diárias gratuitas seriam suficientes para o desenvolvimento do projeto e (iii) verificar quais os parâmetros e formatos necessários para o uso da API.

Neste teste criou-se um simples aplicativo de celular, baseado em Android, que pudesse enviar uma demanda ao servidor do Google e obter a resposta com as coordenadas geográficas e a precisão destas. Com este aplicativo foi possível testar a validade da chave de registro, verificar como se dava as contagens das demandas – por exemplo, mesmo que, por qualquer motivo, o resultado voltasse como “Erro” contabilizava-se uma demanda – e também permitiu explorar a semântica necessária às demandas (realizadas no formato JSON).

Assim, este teste mostrou que seria possível a utilização desta API no desenvolvimento do projeto.

b) Teste dos resultados da API

Com o acesso já garantido à API, realizou-se mais testes para verificar a qualidade dos resultados fornecidos por ela. A partir destes testes descobriu-se diversas limitações da API que não foram expostas na documentação (GOOGLE DEVELOPERS, 2014). Os testes foram realizados variando-se a quantidade de antenas de celular utilizadas, a quantidade de pontos de acesso Wi-Fi e a origem destes dados. Utilizou-se dados obtidos experimentalmente, de antenas localizadas próximas à Cidade Universitária e dados fornecidos como exemplo pelo Google, cuja antena se localiza em sua sede, na Califórnia, EUA.

A seleção de dados utilizada é apresentada na tabela a seguir.

Tabela 3 - Geolocation - Dados utilizados

	Dados experimentais	Dados de exemplo do Google
Localização esperada	Cidade Universitária	Sede do Google, Califórnia, EUA
Antenas (MCC, MNC, LAC, CellID)	Antena A (724,10,9511,13793) Antena B (724,10,9511,13796)	Antena C (310,260,40495,39627456)
Pontos de acesso Wi-fi (MAC Address)	MAC 1 (30:85:A9:69:F8:AF) – Rede LAB_DIDÁTICOS MAC 2 (00:25:90:D2:42:1F) – Rede PoliSemFio	MAC 3 (01:23:45:67:89:AB) MAC 4 (01:23:45:67:89:AC)

Fonte: HOSHINO e WAI (2014)

Os testes realizados e seus resultados são mostrados na próxima tabela, onde se indica quais antenas e/ou pontos de acesso foram utilizados.

Tabela 4 - Teste de localização - API

Teste	Antenas	Wi-Fi	Latitude,Longitude	Precisão (m)	Local
1	A	-	-23.559545800000002, -46.730998	876,0	Cidade Universitária
2	A, B	-	-23.559545800000002, -46.730998	876,0	Cidade Universitária
3	B	-	-23.559219, -46.730295	835,0	Cidade Universitária
4	B,A	-	-23.559545800000002, -46.730998	876,0	Cidade Universitária
5	A	1,2	-23.559545800000002, -46.730998	876,0	Cidade Universitária
6	-	1,2	-23.5169967, -46.7235716	512.0	Vila Anastácio – Local do teste
7	C	3,4	37.4248297, - 122.07346549999998	1145.0	Califórnia, EUA
8	C	-	37.4248297, - 122.07346549999998	1145.0	Califórnia, EUA
9	-	3,4	-23.5169967, -46.7235716	512.0	Vila Anastácio – Local do teste

Fonte: HOSHINO e WAI (2014)

A partir destes testes percebe-se que a API, em sua versão gratuita, não retorna os dados como esperado. A documentação indica que a API pode receber zero ou mais antenas de celular, no entanto, o envio de 2 ou mais antenas não causa alterações no resultado, sendo considerado apenas os dados da primeira (testes 1-4 e 8).

Já em relação ao Wi-Fi, a documentação exige 0, 2 ou mais MAC Address, no entanto, quando apenas os dados de Wi-Fi foram enviados a localização obtida não foi aquela dos pontos de acesso da rede, e sim a localização na qual o teste estava sendo realizado (testes 6 e 9). Este caso indica fortemente a incoerência da API, já

que dados de Wi-Fi, disponibilizados como exemplo pelo Google, retornam uma localidade no Brasil.

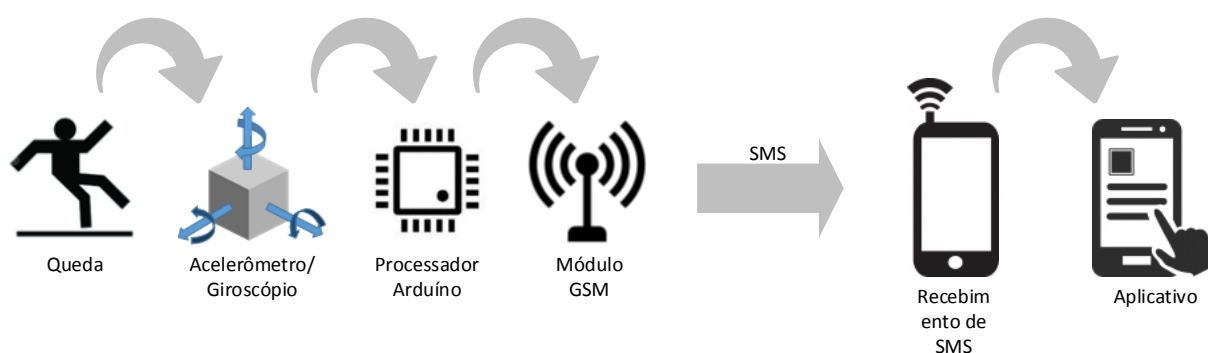
E, caso haja o envio tanto de dados de antenas de celular e dados de Wi-Fi, apenas os dados da primeira antena de celular são considerados (testes 5 e 7).

A partir destes testes optou-se por não utilizar dados de Wi-Fi para a localização, já que estes não são utilizados na Geolocation API, ao menos em sua versão gratuita, que é a que foi testada.

3.5 ESTRUTURA

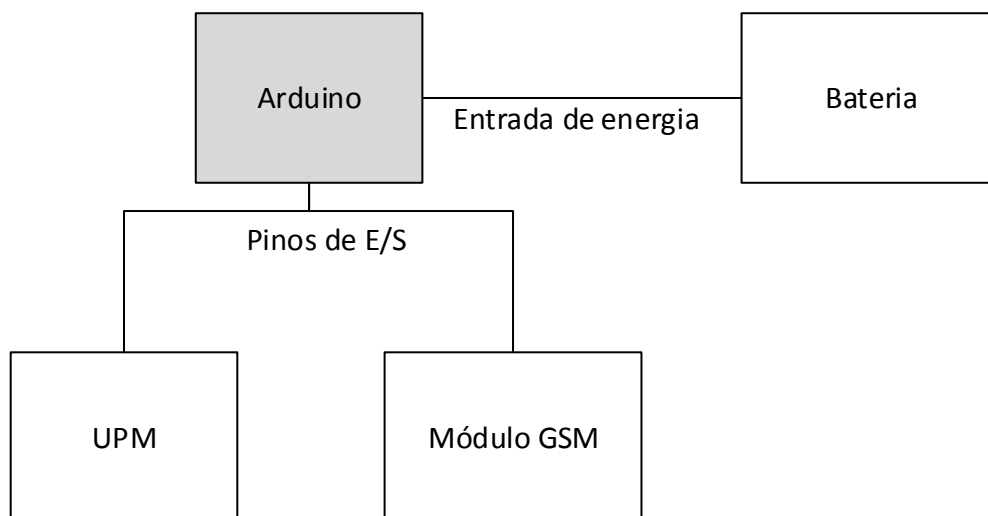
Após a definição dos tipos de componentes a serem utilizados, pode-se atualizar o caminho que a informação percorrerá desde a queda, até a visualização do alerta e do local da queda.

Figura 11 - Caminho da informação - especificação



Fonte: HOSHINO e WAI (2014)

A estrutura escolhida para o dispositivo, conforme a escolha dos componentes é a seguinte:

Figura 12 - Estrutura do dispositivo

Fonte: HOSHINO e WAI (2014)

A abordagem escolhida para esse projeto foi de uma estrutura modular, em que cada componente seria capaz de se comunicar com o microprocessador de maneira independente para que posteriormente fosse feita a integração completa do sistema. Tal escolha deveu-se a maior facilidade que existe em manipular cada componente de maneira independente, na possibilidade de se dividir o teste e a manipulação dos componentes a cada um dos membros do grupo e à existência de inúmeros módulos disponíveis no mercado e que atendem aos métodos escolhidos para se obterem os dados.

O projeto foi composto por um módulo capaz de fazer a medição de acelerações lineares e velocidades angulares, um módulo capaz de captar sinais de antenas ao seu redor e pegar as suas informações, de ler cartão SIM e de enviar mensagens SMS e, para integrar os dois módulos, um microprocessador.

4 IMPLEMENTAÇÃO

Nesta seção será descrito todo o procedimento do projeto, começando pela escolha dos componentes utilizados com base nos métodos escolhidos na etapa de desenvolvimento. Em seguida, a descrição dos softwares utilizados em cada módulo para que os mesmos realizassem corretamente as funcionalidades que lhe foram atribuídas e para garantir a comunicação correta entre o módulo e o processador. Posteriormente, a integração física entre os módulos e o processador a partir da montagem de protótipos e os testes finais do dispositivo obtido ao fim do projeto, tais como análises a respeito dos resultados adquiridos com os testes. Finalmente, as conclusões a respeito do projeto e trabalhos que seriam interessantes a ser implementados para complementar e melhorar o projeto.

4.1 ESCOLHA DOS COMPONENTES

Com base nas funcionalidades definidas nos requisitos do projeto e nos métodos adotados no desenvolvimento para atender a essas funcionalidades, foram escolhidos quais componentes seriam de fato comprados e utilizados no projeto. Para essa escolha, além da funcionalidade, também foi levado em conta atributos como preço e disponibilidade no mercado.

4.1.1 SMARTPHONE COMO DETECTOR DE QUEDA

Analisando as funcionalidades do projeto, notaram-se diversos métodos que também estão presente em *smartphones* atuais. Devido a isso, foi natural a dúvida do por que não utilizar o *smartphone* para se detectar quedas ao invés de se implementar um dispositivo unicamente com esse intuito. Para essa análise, foram analisados diversos pontos apresentados a seguir.

4.1.1.1 USO

Segundo dados do IBGE de 2011, a porcentagem de pessoas com idade acima de 60 anos e que utiliza celular é de 43,9% do total de idosos, ou seja, menos da metade. O motivo disso está, em parte, no fato de uma parcela desse grupo (aproximadamente 24%) possuir certa relutância em adotar novas tecnologias, da não necessidade de se utilizar o aparelho celular e a algumas complicações físico-motoras tais como a problemas para ler caracteres na tela devido à visão prejudicada ou dificuldade em digitar textos ocasionados por possíveis tremedeiras do corpo (IBGE, 2013) (ALENCAR, 2013).

4.1.1.2 ABRANGÊNCIA

Uma grande limitação do *smartphone* como um detector de queda seria a necessidade de o aparelho estar junto ao corpo para que seja o monitoramento. Para ambientes fora de casa, o seu uso é válido, no entanto, no ambiente doméstico, é menos usual carregar o celular consigo mesmo. Além disso, em situações como o banho, o uso do celular é inviável.

4.1.1.3 PREÇO

Os preços de *smartphones* no mercado que possuem potencial para detecção de quedas é de, no mínimo, R\$400,00. A ideia de se fazer um dispositivo é eliminar funcionalidades presentes no celular que seriam irrelevantes para a detecção de queda, por exemplo, a tela. Consequentemente, é possível reduzir os custos de produção.

4.1.1.4 BATERIA

Devido à maior complexidade do celular, a bateria tem capacidade de duração de, na média, um dia de uso. Porém, utilizando componentes mais simples ou eliminando alguns mais complexos, estima-se que a duração aumente, conforme

mentionado anteriormente. É importante mencionar que apenas a tela do celular consome, em média, 30% da bateria.

4.1.2 ARDUINO

Há mais de 20 placas Arduino no mercado então, para se escolher a mais adequada ao projeto, listou-se as características mais relevantes das principais placas disponíveis. Como todas as placas são baseadas em processadores Atmega de 16MHz e no máximo 2,5KB de SRAM, a capacidade de processamento não foi considerada como fator relevante de escolha (CASTLE, 2013).

Tabela 5 - Comparativo Arduino

Arduino	# páginas no Fórum oficial	# periféricos, oficiais e de terceiros, na loja oficial¹	Extras	Preço (US\$)
Uno	213.000	28	Versão de entrada	25
Leonardo	19.900	28	Compatibilidade com USB embutida	25
Due	120.000	3	Tamanho maior	50
Micro	44.500	0	Tamanho menor	27
Lilypad	3.600	1	Costurável	22
Esplora	34.100	1	Montagem como joystick	60
Yún	44.400	28	Wi-Fi embutido	65

Fonte: ARDUINO (2014)

Após a comparação entre as placas disponíveis escolheu-se o Arduino Uno, principalmente por ter (i) maior quantidade de tutoriais e informações, (ii) maior número de periféricos compatíveis, (iii) maior facilidade de uso e (iv) preço acessível.

¹ Computou-se os periféricos apenas pelo seu tamanho – quer permitiria um encaixe fácil com a placa. No entanto, normalmente os periféricos são feitos para o Uno, sendo necessário alguns ajustes para seu uso com o Leonardo e o Yún

4.1.3 DETECTOR DE QUEDA

O componente escolhido para que seja feita a detecção de queda foi o MPU-6050. O motivo para a escolha desse componente foi devido ao seu(i) preço (\$5,00), (ii) possuir um acelerômetro e um giroscópio integrados, (iii) ser compatível com o Arduino, (iv) possuir um processador de movimento próprio, capaz de gerar interrupções com base nas medições de suas medições.

A unidade de processamento de movimento escolhida para o projeto foi o MPU-6050 (Motion Processing Unit). O módulo é composto por um acelerômetro 3-axial MEMS, um giroscópio 3-axial MEMS e um processador digital de movimento. Ele possui um conversor analógico-digital de 16 bits para cada canal e utiliza o I2C-bus para fazer a interface com o Arduino. Além disso, o módulo também possui um termômetro para que eventuais desvios de leitura nos valores, ocasionados pela variação de temperatura do componente, possam ser corrigidos adequadamente.

4.1.3.1 ACELERÔMETRO

O acelerômetro consiste em um MEMS (microelectromechanical system) 3-axial. Ele possui quatro faixas de operação, sendo elas: $\pm 2g$, $\pm 4g$, $\pm 8g$ e $\pm 16g$. Além disso, ele possui dois modos de operação: o padrão que opera com uma corrente de 500 μA e a de baixo consumo que opera com uma corrente de 40 μA com uma frequência de 10 Hz.

4.1.3.2 GIROSCÓPIO

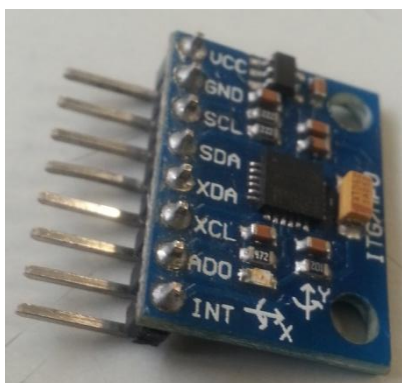
O giroscópio consiste em um MEMS 3-axial também, sendo os seus eixos de coordenadas comuns ao do acelerômetro. Suas faixas de operação são: ± 250 , ± 500 , ± 1000 , e $\pm 2000^\circ/\text{sec}$. O giroscópio também possui dois modos de operação: o padrão que opera com uma corrente de 5 mA e o modo de repouso que opera com uma corrente de 5 μA .

4.1.3.3 PROCESSADOR DE MOVIMENTO

O MPU-6050 possui um processador interno capaz de fazer a leitura e análise dos seis dados obtidos do acelerômetro e do giroscópio e, dependendo de sua configuração, gerar interrupções específicas que são enviadas pelo pino de interrupção INT. Existem três tipos de interrupções no processador: o Free fall interrupt, que é gerada quando o dispositivo detecta que ele encontra-se em queda livre, o Motion interrupt, que é gerada quando o dispositivo detecta algum movimento em qualquer um de seus eixos, e o Zero motion interrupt, que é gerada quando o dispositivo não detecta movimentos em nenhum dos seus três eixos. Devido ao foco do projeto, que é a detecção de queda, optou-se por utilizar a interrupção Free fall interrupt.

O Free fall interrupt é composto por três registradores: FF_TRH, FF_DUR e o MOT_DETECT_CTRL. Para essa interrupção, a queda começa a ser detectada no momento em que o valor absoluto da aceleração dos três eixos fica abaixo do valor definido no registrador FF_TRH. Para cada valor absoluto abaixo do definido no FF_TRH, um contador é incrementado, já para cada valor acima do, o contador é decrementado. A interrupção de queda apenas será emitida quando o valor do contador se igualar ao valor definido no registrador FF_DUR. Opcionalmente, é possível também configurar a velocidade de decremento do contador através do registrador MOT_DETECT_CTRL. Com esse registrador, pode-se configurar o decremento para zerar o contador diretamente ou decrementá-lo em passos de 1, 2 ou 4. (INVENSENSE INC., 2010)

Figura 13 - MPU-6050 – vista frontal.



Fonte: HOSHINO e WAI (2014)

Figura 14 - MPU-6050 – vista posterior

Fonte: HOSHINO e WAI (2014)

4.1.4 MÓDULO GSM

O módulo escolhido para o projeto foi o Shield Sim900, compatível com Arduino. Esta escolha ocorreu pois o módulo oficial GSM do Arduino estava indisponível para compra e, portanto, optou-se por comprar o módulo GSM no mercado brasileiro. Este módulo apresentou o melhor preço e atende aos requisitos ao ser compatível com o Arduino e possibilitar a conexão à rede GSM brasileira.

Figura 15 - Módulo GSM Sim900

Fonte: NEWHARDWARE (2014)

4.1.5 BATERIA

A bateria escolhida para o projeto foi uma de li-po, conforme especificado no projeto, de 7,4 V e 800mAh. Essa bateria foi a escolhida por se tratar de um protótipo, no qual a duração da bateria não é o foco e pelo preço mais acessível (R\$130,00). Em um produto final o ideal seria uma bateria de maior capacidade para reduzir a necessidade de recarregar constantemente o dispositivo. A tensão da bateria deve ser acima de 5V para alimentar a placa Arduino e o módulo GSM utilizados.

4.1.6 APLICATIVO

Para a escolha da versão mínima do Android suportada pelo aplicativo consideraram-se dois fatores: (i) facilidade de programação e (ii) número de usuários.

Várias funcionalidades foram adicionadas e simplificadas a partir da versão 4.0 do Android. Diversas bibliotecas foram substituídas de modo a lidar melhor com os dispositivos mais novos, com maior processamento, além de facilitar o desenvolvimento de aplicativos.

A tabela a seguir indica o percentual que cada versão do Android representa do total de usuários da Google Play Store em Setembro de 2014.

Tabela 6 - Distribuição das Versões Android

Version	Codename	API	Distribution
2.2	Froyo	8	0.7%
2.3.3 - 2.3.7	Gingerbread	10	11.4%
4.0.3 - 4.0.4	Ice Cream Sandwich	15	9.6%
4.1.x	Jelly Bean	16	25.1%
4.2.x		17	20.7%
4.3		18	8.0%
4.4	KitKat	19	24.5%

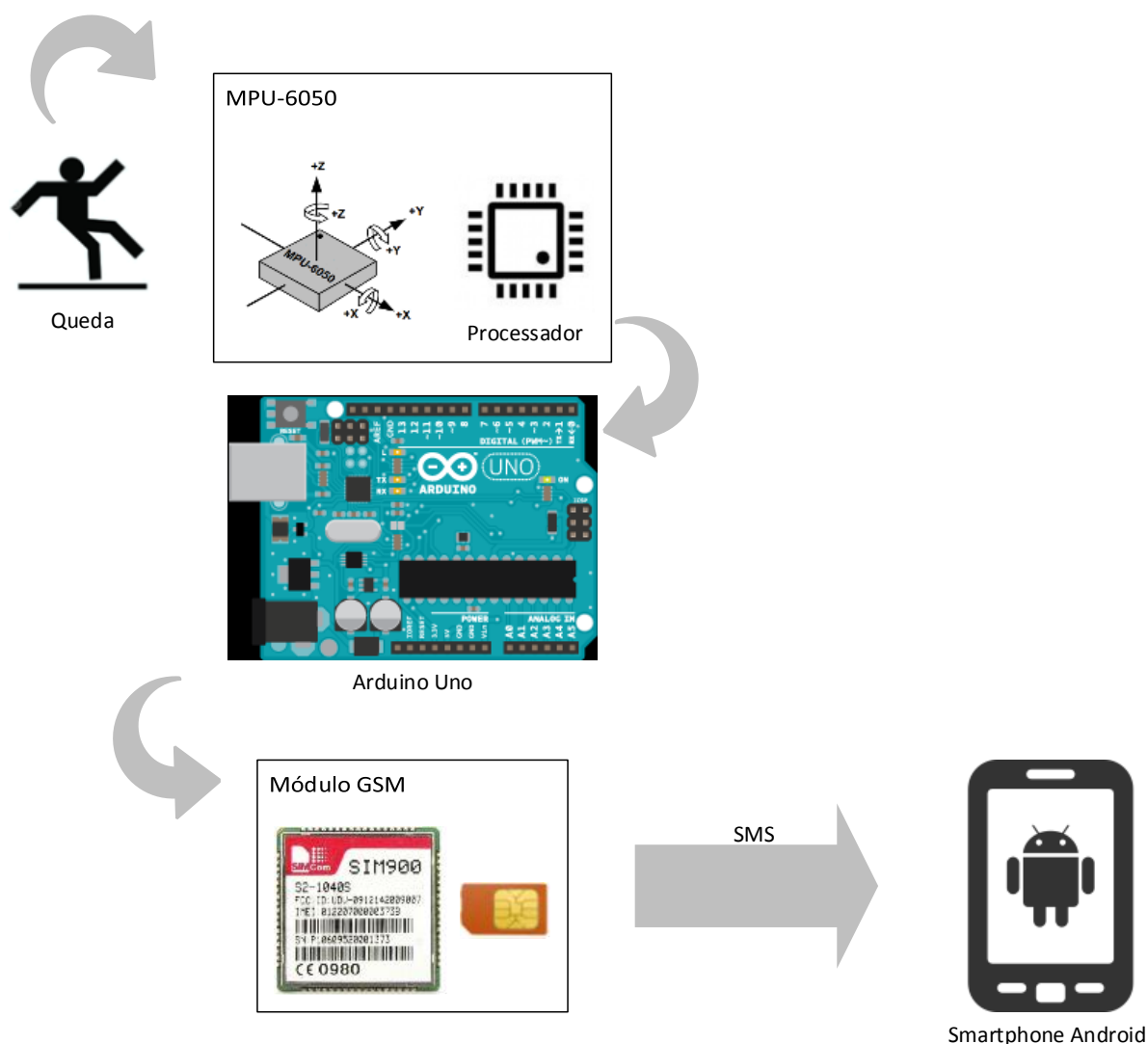
Fonte: GOOGLE DEVELOPERS (2014)

Considerando a facilidade de desenvolvimento e o percentual de cada versão, optou-se por realizar o aplicativo deste projeto em versão mínima 4.1.x, de modo a rodar em mais de 75% dos dispositivos Android. Posteriormente é possível utilizar bibliotecas de compatibilização com as versões mais antigas.

4.1.6.1 FLUXO FINAL

Uma vez definido todos os componentes e o modo de como eles iriam atuar, foi possível desenhar um fluxo de detecção de queda final. Assim, resumidamente, uma vez que o usuário do dispositivo sofre um acidente de queda, o MPU-6050 identifica essa queda e gera uma interrupção no processador. O processador, por sua vez, aciona o módulo GSM, espera que ele colete os dados das antenas presentes em suas proximidades e faz com que ele envie um SMS contendo esses dados para o número cadastrado em seu sistema. Uma vez recebido o SMS pelo número cadastrado, o aplicativo desenvolvido, presente nesse celular, lê esse SMS, extrai os dados das antenas enviadas e faz a localização do usuário a partir desses dados, mostrando em um mapa o local aonde ocorreu o acidente. Esse fluxo de trabalho foi descrito na Figura 16.

Figura 16 – Fluxo final



Fonte: HOSHINO e WAI (2014)

4.2 SOFTWARES DOS MÓDULOS

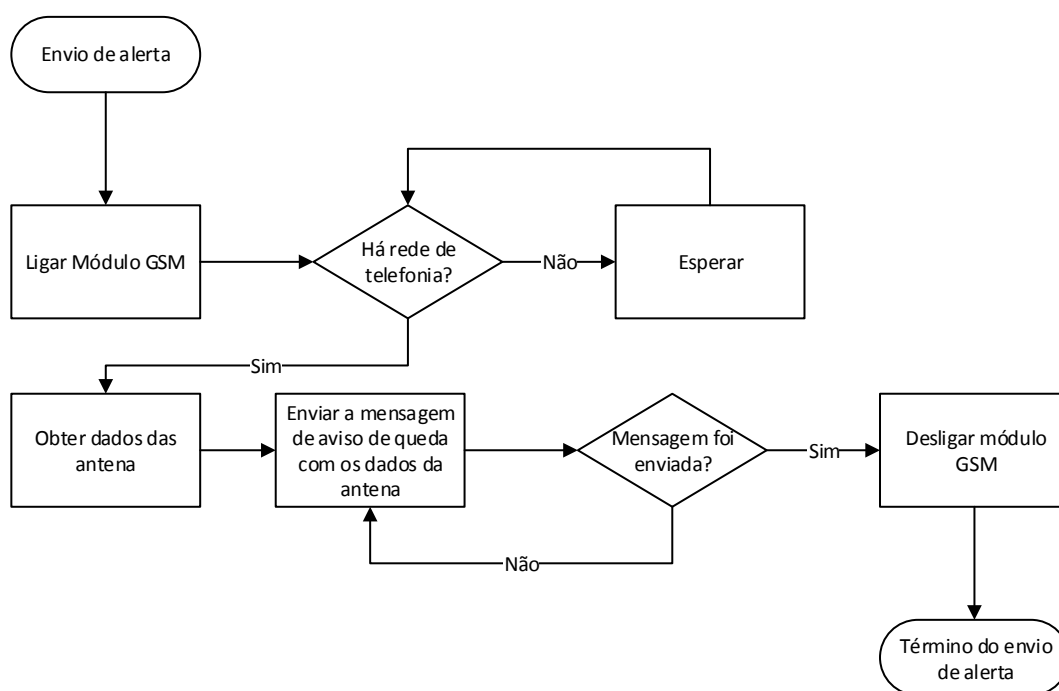
Para que cada componente funcionasse corretamente, executasse de maneira adequada as funções que lhe foram atribuídas e conseguisse se comunicar com o processador foi necessário inserir, através do processador, códigos de programa (software). A seguir é descrito os programas feitos para cada um dos módulos, junto com a respectiva funcionalidade ao qual o programa se refere.

4.2.1 MÓDULO GSM

O módulo GSM é responsável pelo envio das mensagens de alerta aos celulares cadastrados. Para que tal envio seja possível, após a detecção da queda deve-se: (i) ligar automaticamente o módulo, (ii) obter os dados das antenas telefônicas, (iii) enviar a mensagem, (iv) desligar o módulo, e (v) verificar que cada etapa foi bem-sucedido, lendo as respostas enviadas pelo módulo após cada comando.

O fluxograma do funcionamento do GSM é mostrado a seguir:

Figura 17 - Fluxograma do GSM



Fonte: HOSHINO e WAI (2014)

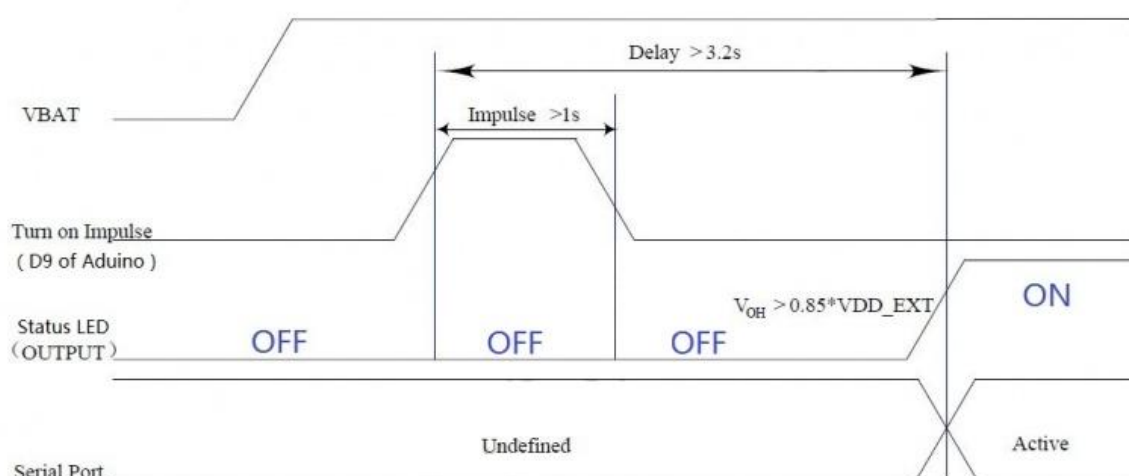
Esse fluxograma mostra o funcionamento no caso do envio da mensagem de alerta de queda. Além disso, o módulo GSM é necessário para o cadastramento de celulares e para o envio de alertas de bateria baixa.

4.2.1.1 LIGAR

É possível ligar o módulo automaticamente utilizando um de seus pinos de entrada. Assim, utilizou-se uma rotina para ligar o módulo GSM apenas quando necessário. Sem esta rotina, o módulo deveria ser ligado manualmente, através de um botão, o que inviabilizaria o uso do dispositivo, seja pela necessidade de pressionar o botão ou pela necessidade de manter o módulo ligado continuamente, o que consumiria muita bateria.

A rotina utilizada foi obtida de COOKING HACKS (2014) e tem como base a curva necessária para o acionamento do módulo GSM, que é mostrada a seguir.

Figura 18 - Software Trigger GPRS



Fonte: COOKING HACKS (2014)

Esta mesma rotina é responsável pelo desligamento do módulo GSM. Assim, após o envio do SMS o módulo é desligado para poupar bateria.

4.2.1.2 COMUNICAÇÃO SERIAL

Para a interação com o módulo GSM foi necessário o uso de uma série de comandos específicos. Os comandos utilizados para o chip SIM900 do módulo GSM

foram comandos AT comumente utilizados em comunicações de telefonia. A partir destes comandos foi possível verificar se há rede disponível, verificar quais os dados das antenas de celulares, enviar mensagens de texto, realizar chamadas de voz, desligar o módulo, entre outros.

Para se testar o uso e sintaxe dos comandos criou-se um programa no Arduino que teve como base os exemplos disponíveis em (GPRS/GSM Quadband Module for Arduino, 2014). Inicialmente realizou-se uma conexão serial direta entre o terminal do computador e o módulo GSM, ligando diretamente os pinos RX e TX do Arduino e do GSM. A partir dessa conexão foi possível testar os comandos AT que seriam necessários ao projeto, como o envio de mensagens, o desligamento do módulo, a obtenção de dados da rede de telefonia móvel, entre outros. Para consultar os comandos utilizou-se o manual disponível em (SIM900 AT Command Manual, 2010). No entanto, a conexão serial direta não é viável ao projeto, já que a conexão do módulo GSM deve ser feita com o processador do Arduino e não com um computador. Essa outra conexão serial foi realizada com a biblioteca Software Serial do Arduino (ARDUINO, 2014), que permite que o Arduino tenha duas ou mais conexões seriais ativas, neste caso uma com computador, necessária para a depuração do projeto durante o desenvolvimento, e outra com o módulo GSM. Ou seja, a biblioteca permite que os pinos RX e TX do módulo GSM sejam ligados em outras portas digitais do Arduino.

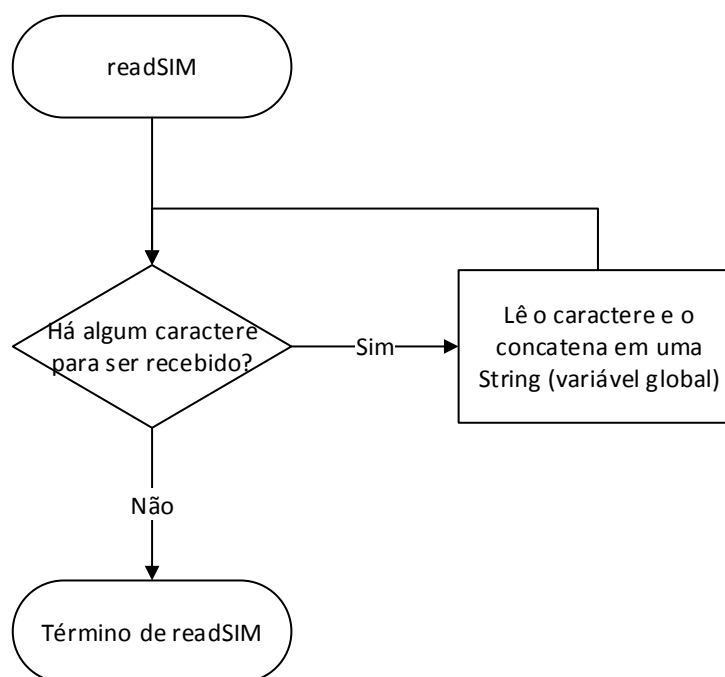
Devido à utilização da biblioteca Software Serial foi necessário uma atenção extra em relação ao *baud rate* utilizado na conexão com o módulo GSM. O componente escolhido opera, por padrão, a 19.200 bps, no entanto, a biblioteca Software Serial opera a, no máximo, 9.600 bps. Desse modo, foi necessário ajustar a configuração do módulo GSM através do comando AT correspondente.

4.2.1.3 LER RESPOSTAS

Cada comando AT retorna uma resposta indicando o resultado do comando. Tal resposta pode ser, por exemplo, um “OK”, um “ERROR”, ou uma cadeia de caracteres. Deste modo, a leitura dessas respostas é essencial para o uso de certos comandos, como a obtenção de dados das antenas, ou para verificar o sucesso de certos comandos, como o de envio de mensagens.

A função criada responsável pela leitura das respostas se baseia na leitura dos caracteres enviados pela serial que usa a biblioteca Software Serial e segue o fluxograma da Figura 19. A função recebe caracteres do SIM900 e os guarda em uma String. Esta função é chamada quando se sabe que deveria haver alguma resposta do SIM900, por exemplo, após cada envio de comando AT. Mesmo que o comando não seja bem sucedido, sempre há alguma resposta.

Figura 19 - Fluxograma da leitura da serial



Fonte: HOSHINO e WAI (2014)

4.2.1.4 DISPONIBILIDADE DO MÓDULO

Após o acionamento do módulo GSM, mas antes do envio/recebimento de mensagens de texto, verifica-se a existência de rede de telefonia válida. Esta verificação é feita através do seguinte comando AT:

AT+COPS?

Este comando retorna o operador ao qual o módulo está atualmente conectado, como por exemplo, a operadora VIVO. A resposta do comando segue o formato:

```
+COPS: 0,0,"VIVO"  
OK
```

Assim, uma resposta bem-sucedida a este comando indica que o módulo GSM está conectado à rede e pode, conseqüentemente, realizar o envio/recebimento de mensagens de texto.

4.2.1.5 ENVIO DE MENSAGENS

Com o uso da conexão serial entre o Arduino e o módulo GSM, descrita anteriormente, já foi possível enviar mensagens manualmente, com o uso de comandos AT. No entanto, para a realização deste projeto, este envio deveria ser automatizado. Portanto, criou-se uma função no Arduino para efetuar o envio de mensagens.

Esta função é responsável por enviar os comandos AT necessários. Basicamente envia-se um comando informando o número ao qual se deseja enviar o SMS e, depois, outro comando informando o texto do SMS. Entre esses dois comandos é necessário que haja um intervalo de espera, de modo a permitir o processamento do comando.

O comando AT utilizado para o envio de mensagens é:

```
AT+CMGS=<número de celular><Carriage Return><Texto><ctrl-Z>
```

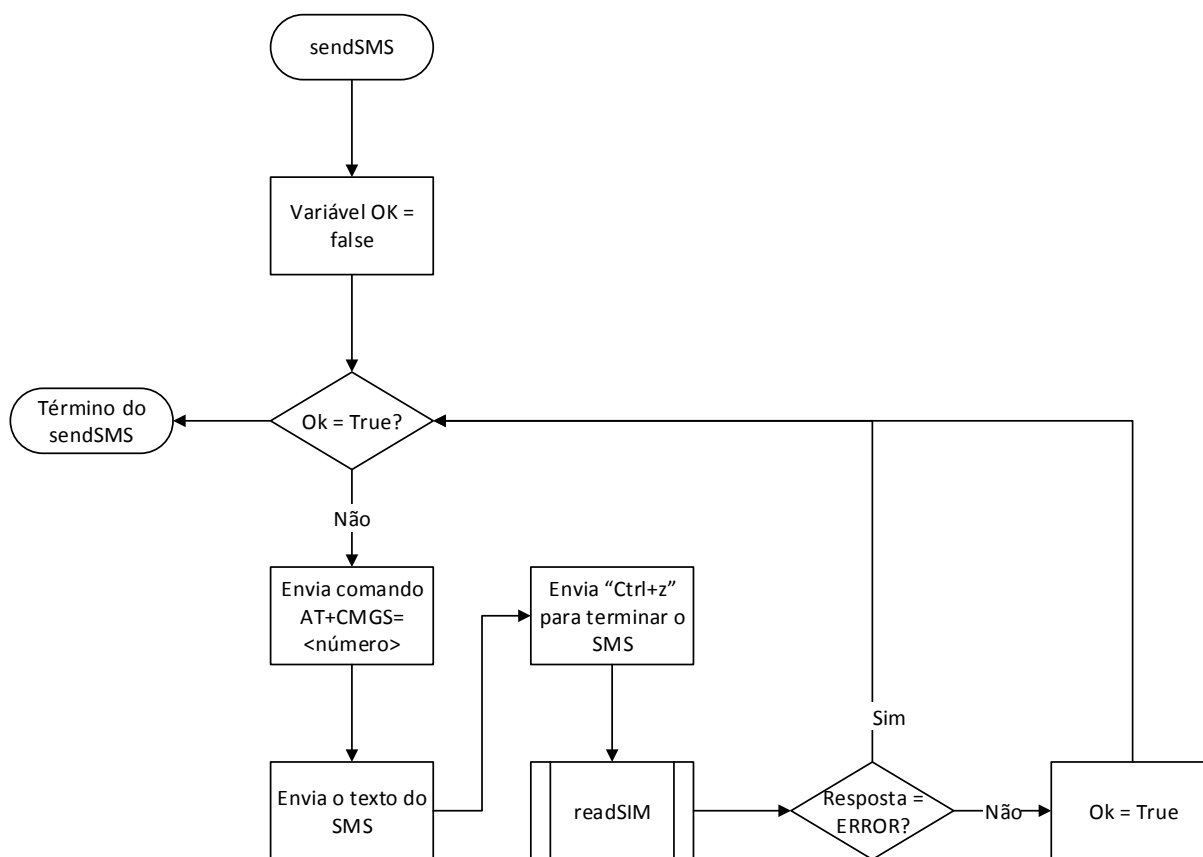
Caso o envio seja bem sucedido a resposta é:

```
+CMGS=<referência da mensagem>  
OK
```

Caso o envio falhe, a resposta é:

```
+CMS ERROR
```

A rotina criada para o envio automatizado de mensagens é explicada na Figura 20.

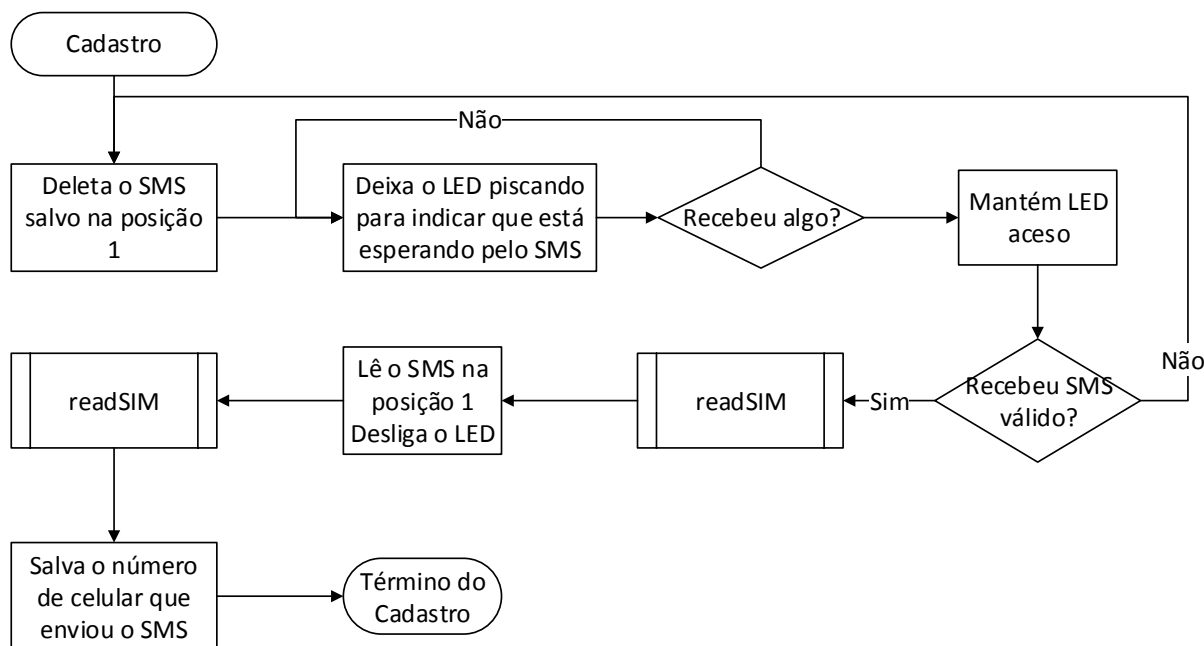
Figura 20 - Fluxograma do envio de SMS

Fonte: HOSHINO e WAI (2014)

4.2.1.6 CADASTRO DE CELULAR

O recebimento de mensagens de texto é necessário para o cadastro do número de telefone para o qual os alertas serão enviados, conforme o Caso de Uso 4. O módulo GSM utilizado pode armazenar até 30 SMS. Quando uma nova mensagem é recebida pelo módulo, ela é armazenada nessa memória, na primeira posição livre de memória.

O funcionamento da rotina de recebimento de SMS é mostrado no fluxograma a Figura 21.

Figura 21 - Fluxograma do cadastro de celular

Fonte: HOSHINO e WAI (2014)

Os comandos utilizados nesta rotina serão descritos a seguir.

Para liberar uma posição de memória, apaga-se o SMS que estiver armazenado na posição 1 com o seguinte comando:

$$AT+CMGD=1$$

Quando algum SMS é recebido pelo módulo há a transmissão de dados do módulo para o Arduino, indicando uma nova mensagem. Assim, o programa espera por essa comunicação iniciado pelo módulo GSM.

Após o recebimento do SMS, este pode ser lido na posição de memória 1, liberada anteriormente, com o seguinte comando:

$$AT+CMGR=1$$

Na leitura do SMS, o número de celular do emissor é mostrado, e é este número que será armazenado no programa como o celular cadastrado para o envio de mensagens de alerta. Também é realizada uma verificação do texto da mensagem para garantir que o SMS tenha tido mesmo o propósito de realizar o cadastro do

telefone. Assim, o SMS de cadastro deve iniciar com “SALUS” de modo a ser considerado válido.

Após o cadastro do número, o dispositivo envia um SMS, confirmando o cadastro, para o número recém-cadastrado. O estado atual do projeto considera apenas um número de celular cadastrado, de tal forma que um novo cadastro apaga o número antigo.

4.2.1.7 OBTENÇÃO DOS DADOS DAS ANTENAS

Para a obtenção da localização do dispositivo é necessária a obtenção dos dados das antenas próximas ao dispositivo. Os dados necessários são:

- MCC – Mobile Country Code. Ex: 724 (Brasil)
- MNC – Mobile Network Code. Ex: 10 (Vivo)
- LAC – Location Area Code. Ex: 0x2527
- CellID – Identificação da antena. Ex: 0x35e1

Este conjunto de quatro variáveis indica unicamente uma antena de celular. Utilizou-se como exemplo uma antena da Vivo localizada dentro da Cidade Universitária. Esses dados podem ser obtidos através da ativação do *Engineering Mode* e posterior leitura dos dados.

Ativação do *Engineering Mode* é realizada pelo seguinte comando AT:

AT+CENG=1,1

O primeiro 1 indica a ativação do modo engenheiro e o segundo 1 faz com que os CellIDs de todas as antenas sejam mostrados. O *default* é mostrar apenas o CellID da antena à qual se está conectado.

Após a ativação do modo de engenheiro pode-se obter os dados das antenas próximas ao dispositivo pelo comando:

AT+CENG?

Esse comando gera, por exemplo, a seguinte resposta:

+CENG: 1,1

+CENG:0,"0130,49,99,**724,10**,09,**35e1,15,05,2527,255**"

+CENG:1,"0789,43,09,35e4,724,10,2527"

+CENG:2,"0129,32,255,ffff,000,00,0"

+CENG:3,"0132,33,44,35e0,724,10,2527"

+CENG:4,"0134,32,22,ffff,724,10,2527"

+CENG:5,"0790,28,255,ffff,000,00,0"

+CENG:6,"0788,27,255,ffff,000,00,0"

OK

Os valores em negrito foram os utilizados no exemplo. Após a leitura dessa resposta realiza-se um *parsing* da mensagem de modo a obter apenas os dados relevantes ao projeto. Estes dados serão então enviados por SMS ao cuidador do usuário.

4.2.2 UNIDADE DE PROCESSAMENTO DE MOVIMENTO

A abordagem escolhida para realizar a detecção de quedas foi a leitura e a análise dos dados de aceleração obtidos com o MPU-6050. Para esse método, primeiramente é feita a configuração da faixa de amplitude em que o acelerômetro operaria. Em seguida, os dados obtidos pelo acelerômetro são armazenados em três variáveis do sistema, são convertidos para valores com unidade em g e então é feito o cálculo do módulo dessas acelerações.

Para a conversão dos dados, os valores lidos são divididos por um fator de conversão = 3963,08. Esse fator representa o valor da gravidade captado pelo dispositivo. Ele foi obtido experimentalmente a partir da média dos dados obtidos com o MPU-6050 em repouso, estando sujeito somente à gravidade.

$$AcelMod = \sqrt{acelx^2 + acely^2 + acelz^2}$$

Acelx = aceleração em x, convertida em g;

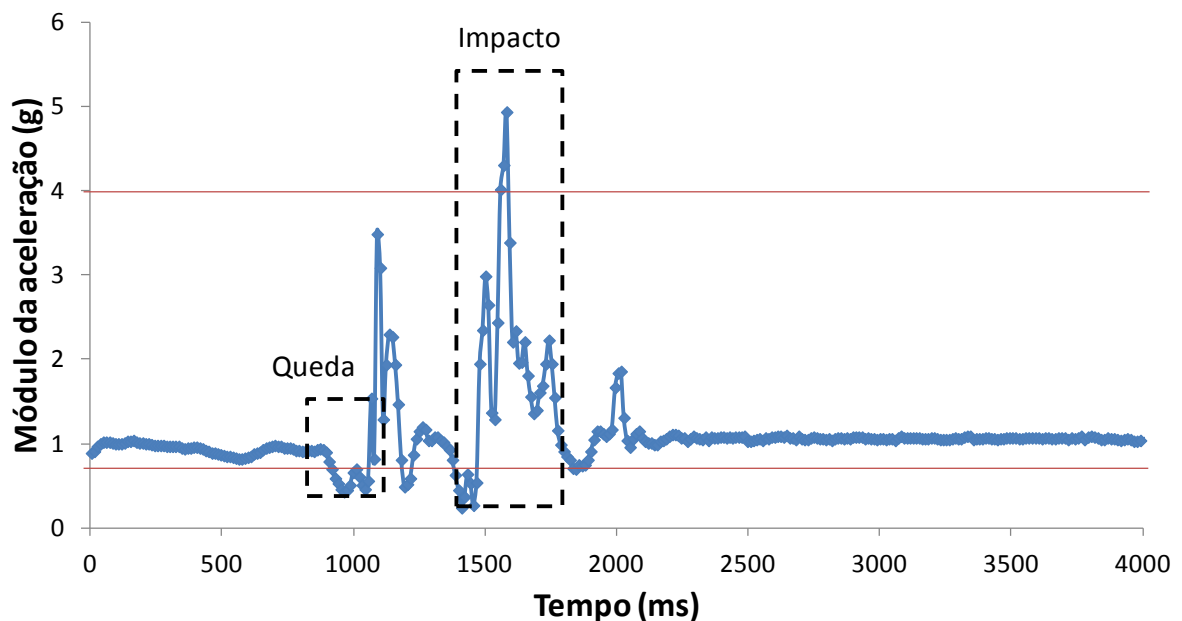
Acely = aceleração em y, convertida em g;

Acelz = aceleração em z, convertida em g;

AcelMod = módulo das acelerações, em g.

Utilizando esses dados, verificou-se que todas as quedas possuem um comportamento similar em suas variações de aceleração. Independentemente do modo como se sofre a queda, é possível dividir a queda em duas partes definidas como “queda” e “impacto”. A “queda”, é caracterizada pela presença de um vale de aceleração. Já o “impacto”, é caracterizado pela presença de picos que sempre ocorrem pouco tempo depois do vale (ordem de milissegundo).

Figura 22 - Perfil de queda.

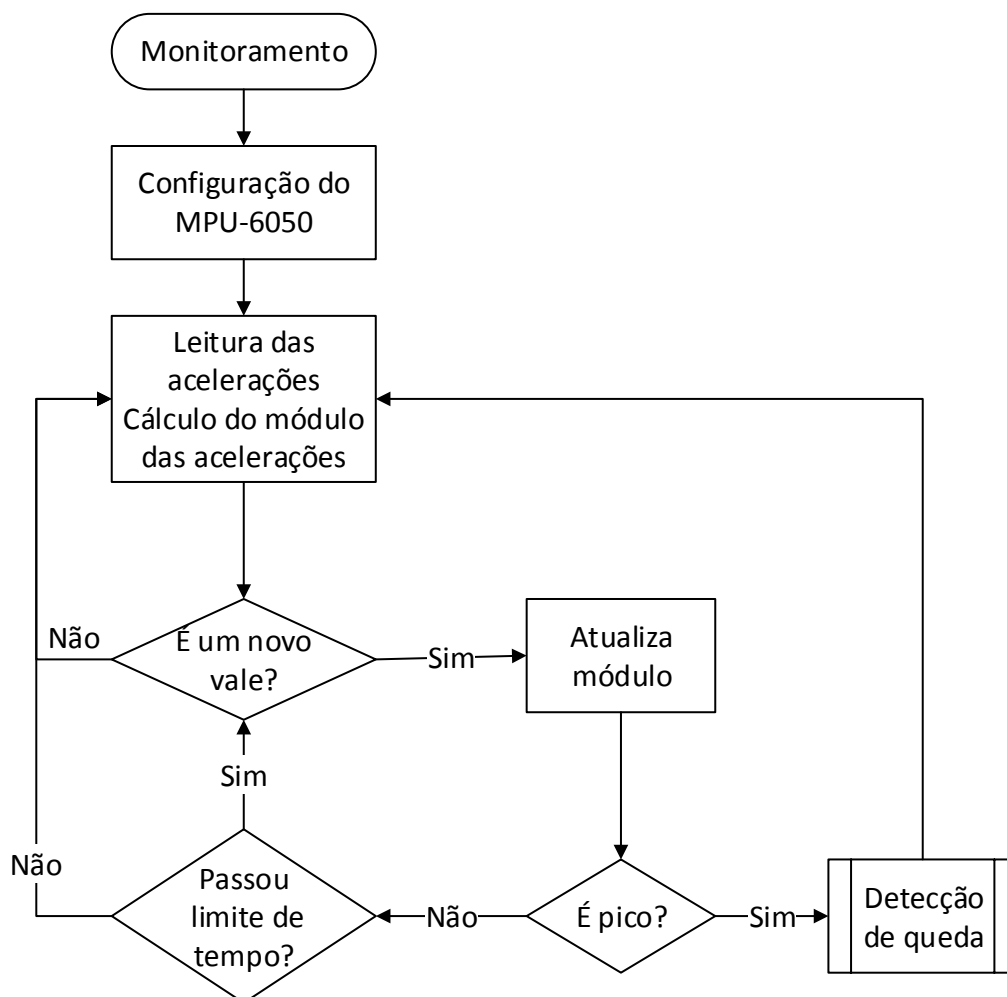


Fonte: HOSHINO e WAI (2014)

Assim, com o valor AcelMod obtido, o sistema passa a tentar identificar vales menores do que 0,7g. Uma vez encontrado o vale, o sistema então passa a buscar a presença de um pico maior que 4,0 g dentro do intervalo de 800 ms desde a identificação do vale. Se essas condições não são respeitadas, o monitoramento recomeça. Do contrário, o sistema identifica a atividade como sendo uma queda e

aciona a rotina de envio de alerta de queda. A Figura 23 mostra o fluxograma do programa.

Figura 23 - Fluxograma de detecção de queda



Fonte: HOSHINO e WAI (2014)

c) Protocolo de comunicação

O protocolo de comunicação que utilizado na comunicação entre o Arduino e o MPU-6050 foi o I2C. Para a configuração desse protocolo, utilizou-se como uma das bibliotecas prontas do Arduino como referência.

d) Registradores

O modulo MPU-6050 possui 107 registradores em seu processador interno. Assim, foi necessário que alguns de seus registradores fossem configurados

adequadamente para que a comunicação com o microprocessador fosse realizada. Para fazer tal configuração, foi utilizada, como referência, uma biblioteca de configuração de todos os registradores do MPU-6050 chamada “MPU6050.h”. Com base nessa biblioteca, tomou-se o cuidado de se ajustar os registradores que seriam utilizados nesse projeto.

Configuraram-se os registradores:

- Register 28 – Accelerometer Configuration (ACCEL_CONFIG)

Tabela 7 - Mapa do registrador 28.

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
28	XG_ST	YG_ST	ZG_ST	FS_SEL[1:0]		-	-	-

Fonte: INVENSENSE INC. (2010)

O registrador 28 é responsável pela configuração de aquisição dos dados de aceleração. No entanto, o registrador 28 é responsável pela configuração do acelerômetro. Esse registrador também possui bits de teste que não foram utilizados no projeto (bits 5 a 7) e três bits vazios (bits 0 a 2). Os bits que foram configurados foram os bits 3 e 4 que definem a amplitude máxima de escala do acelerômetro. Para os testes de queda, foi utilizada uma configuração de $\pm 4g$. Assim, a configuração do registrador ficou:

ACCEL_CONFIG = xxx01xxx

Para a configuração desse registrador, foi necessário utilizar uma máscara para que apenas os pinos FS_SEL<1:0> fossem alterados.

Tabela 8 - Configuração da amplitude máxima do acelerômetro.

FS_SEL	Full Scale Range (g)
0	± 2
1	± 4
2	± 8
3	± 16

Fonte: INVENSENSE INC. (2010)

Os registradores explicados a seguir são relacionados à emissão de interrupções. O MPU-6050 possui três tipos de interrupções no processador: o Free fall interrupt, que é gerada quando o dispositivo detecta que ele encontra-se em queda livre, o Motion interrupt, que é gerada quando o dispositivo detecta algum movimento em qualquer um de seus eixos, e o Zero motion interrupt, que é gerada quando o dispositivo não detecta movimentos em nenhum dos seus três eixos. Devido ao foco do projeto, que é a detecção de queda, optou-se por utilizar a interrupção Free fall interrupt.

- Register 29 – Free Fall Acceleration Threshold (FF_THR)

Tabela 9 - Mapa do registrador 29.

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
29	FF_THR[7:0]							

Fonte: INVENSENSE INC. (2010)

O registrador 29 é utilizado para se definir o parâmetro limiar de aceleração para o qual o dispositivo reconhece como queda livre, sendo tal aceleração o valor absoluto das acelerações presentes nos três eixos do dispositivo.

Para que a interrupção de Free Fall detection seja ativada, existem duas condições que devem ser atendidas: a diferença da aceleração lida e a ação da gravidade deve ser menor que a especificada no registrador 29 e tal aceleração deve ser mantida por um tempo definido pelo registrador 30.

Segundo informações obtidos da comunidade do Arduino, pode-se considerar LSB (least significant bit) = 2 mg. Ou seja, para cada um valor decimal de bit, considera-se 2 mg.

A partir de dados experimentais e visando obter o maior número de quedas detectáveis, utilizou-se o menor valor de aceleração obtido dos testes de queda realizados no projeto. Para o cálculo desse valor, foram feitas as seguintes contas:

Menor valor obtido nos teste = 7.98 m/s^2

Valor da gravidade = 9.81 m/s^2

Diferença de aceleração = $1.83 \text{ m/s}^2 = 190 \text{ mg}$

Valor máximo do registrador = 255 (valor dos bits, em decimal) = 510 mg.

Parâmetro de queda = 190 mg = 95

Logo, o registrador foi configurado como:

FF_THR = 01011111

- Register 30 – Free Fall Duration (FF_DUR)

Tabela 10 - Mapa do registrador 30.

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
30	FF_DUR[7:0]							

Fonte: INVENSENSE INC. (2010)

O registrador é utilizado para se definir o parâmetro de duração de aceleração que o dispositivo deve contabilizar para poder caracterizar a ação como sendo uma queda. Então esse registrador também foi configurado para que a segunda condição descrita anteriormente fosse parametrizada (tempo que a aceleração lida deve se manter abaixo da aceleração referencia).

Para esse registrador, LSB (least significant bit) = 1 ms.

Assim como o registrador 29, visou-se obter o maior número de quedas. Sendo assim, pegou-se o menor tempo de que aceleração de queda. As contas para esse registrador foram as seguintes:

Menor tempo de aceleração $\approx$ 30 ms.

Assim, parâmetro de queda = 30 (valor dos bits, em decimal)

FF_DUR = 0011110.

- Register 56 – Interrupt Enable (INT_ENABLE)

Tabela 11 - Mapa do registrador 56.

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
56	FF_EN	MOT_EN	ZMOT_EN	FIFOFLOW_EN	I2C_MST_INT_EN	-	-	DATA_RDY_EN

Fonte: INVENSENSE INC. (2010)

O registrador é utilizado para habilitar as interrupções do dispositivo. No caso do projeto, a única interrupção que foi utilizada foi a interrupção de queda, Free Fall Detection. Seu bit de configuração é o bit 7. Assim:

INT_ENABLE = 10000000

- Register 105 – Motion Detection Control (MOT_DETECT_CTRL)

Tabela 12 - Mapa do registrador 105.

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
105	-	-	ACCEL_ON_DELAY[1:0]		FF_COUNT[1:0]		MOT_COUNT[1:0]	

Fonte: INVENSENSE INC. (2010)

O registrador é responsável por decrementar o contador da interrupção para os casos em que a aceleração não respeita o parâmetro estabelecido.

Analisando os testes de queda, notou-se que as quedas são eventos de aceleração continua. Sendo assim, qualquer atividade que envolva o surgimento de uma aceleração, mas com intervalos em durante essas acelerações, não devem ser consideradas quedas. Portanto, o melhor decremento seria o “Reset”, em que, se a aceleração obtida pelo dispositivo for maior que a aceleração parametrizada no registrador 29, o contador do registrador 30 deve ser zerado diretamente.

A configuração desse registrador ficou da seguinte maneira:

MOT_DETECT_CTRL = 00000000

Tabela 13 - Configuração do decremento do contador da interrupção.

FF_COUNT	Counter Decrement
0	Reset
1	1
2	2
3	4

Fonte: INVENSENSE INC. (2010)

Estes registradores foram estudados, no entanto, a documentação disponível era superficial e não foi possível obter resultados satisfatórios, de tal modo que a detecção de queda através dessa interrupção programável não foi implementada. No entanto, apresentamos as informações a seguir como registro do estudo realizado.

4.2.3 APLICATIVO

O aplicativo de celular foi separado em três partes principais: (i) Leitura do SMS, (ii) Envio dos parâmetros contidos no SMS à Geolocation API para obtenção das coordenadas geográficas, e (iii) visualização do local da queda no mapa.

4.2.3.1 RECEBIMENTO E AVISO DE MENSAGENS

Para que o aplicativo possa receber mensagens é necessário cadastrar um *SMS Broadcast Receiver* para que o aplicativo saiba quando novos SMS são recebidos pelo dispositivo.

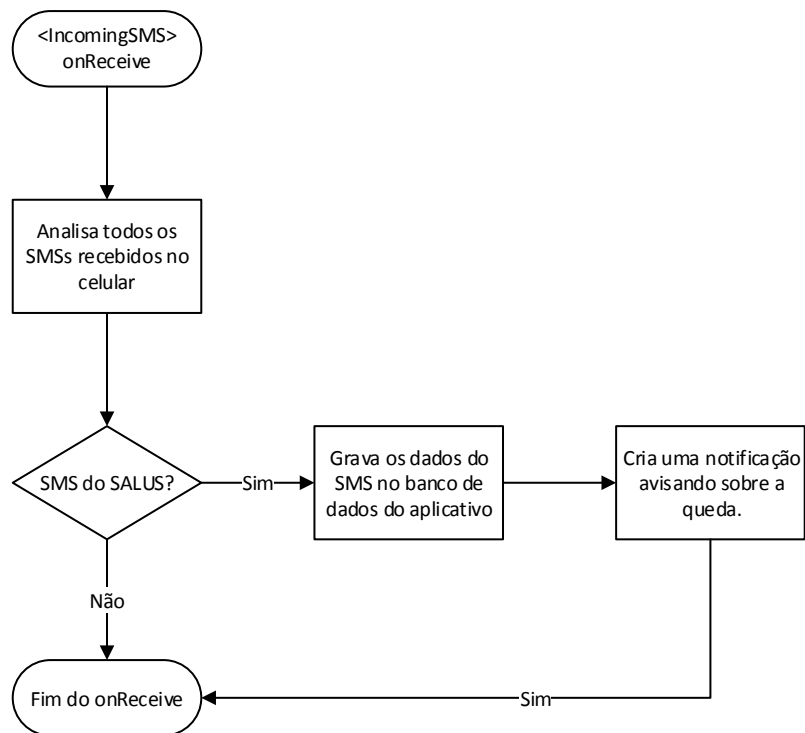
Como base para esta parte do aplicativo utilizou-se o código presente em ANDROID EXAMPLE (2013).

Em qualquer recebimento de SMS o aplicativo é avisado. Este, então, avalia se a mensagem é referente a um alerta de queda, o que é feito analisando o início da mensagem, que deve começar com: “SALUS Queda”. Caso o SMS seja um alerta de queda uma notificação é mostrada ao usuário.

Para a implementação desta parte estudou-se a própria documentação disponibilizada pelo Google (GOOGLE DEVELOPERS, 2014).

O fluxograma da implementação do método `onReceive`, do Broadcast Receiver criado é apresentado na Figura 24. A classe implementada para o Broadcast Receiver é chamada de `IncomingSMS`.

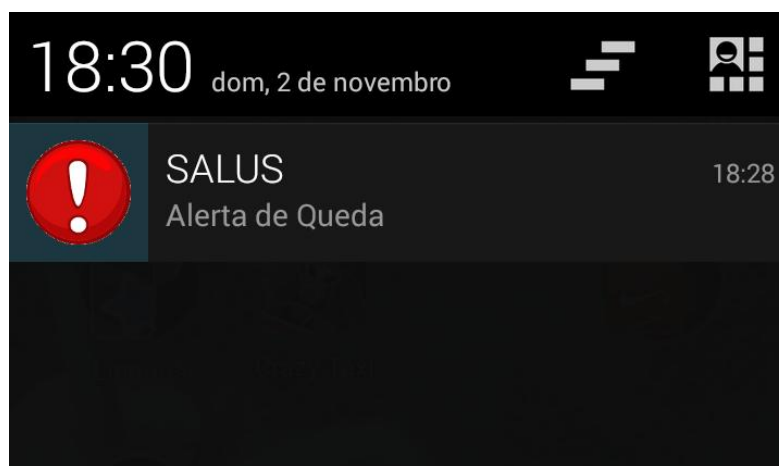
Figura 24 - Fluxograma do recebimento de SMS



Fonte: HOSHINO e WAI (2014)

A notificação tem como função evidenciar o alerta de queda, já que é mais visível que um simples recebimento de SMS.

Figura 25 - Notificação de queda



Fonte: HOSHINO e WAI (2014)

4.2.3.2 USO DA GEOLOCATION API

A mensagem de alerta não informa as coordenadas geográficas da queda, informa apenas alguns dados das antenas. Para obter as coordenadas é necessário enviar os dados das antenas à Geolocation API e interpretar a resposta obtida. Esta resposta deve indicar as coordenadas geográficas (latitude, longitude) e um raio precisão.

Utilizou-se a documentação da API (GOOGLE DEVELOPERS, 2014) que explica quais os dados necessários, qual o formato do arquivo a ser enviado (em JSON), qual o formato da resposta, como se obtém o acesso à API, entre outras coisas.

Inicialmente deve-se obter uma API_KEY que autoriza o uso da API nos aplicativos. A obtenção desta chave é descrita na própria documentação. Esta chave será utilizada no arquivo de configuração do aplicativo (AndroidManifest.xml) e no *link* com o qual será feito a *query* à API.

Para a realização da *query* é necessário organizar os dados provenientes do SMS e os passar para o formato JSON. Neste aplicativo utilizou-se as bibliotecas JSONArray, JSONObject e JSNTokener para lidar com os mensagens nesse formato.

Pelo próprio funcionamento da API é necessária uma conexão com a internet, seja via 3G, 4G ou Wi-Fi (GOOGLE DEVELOPERS, 2014).

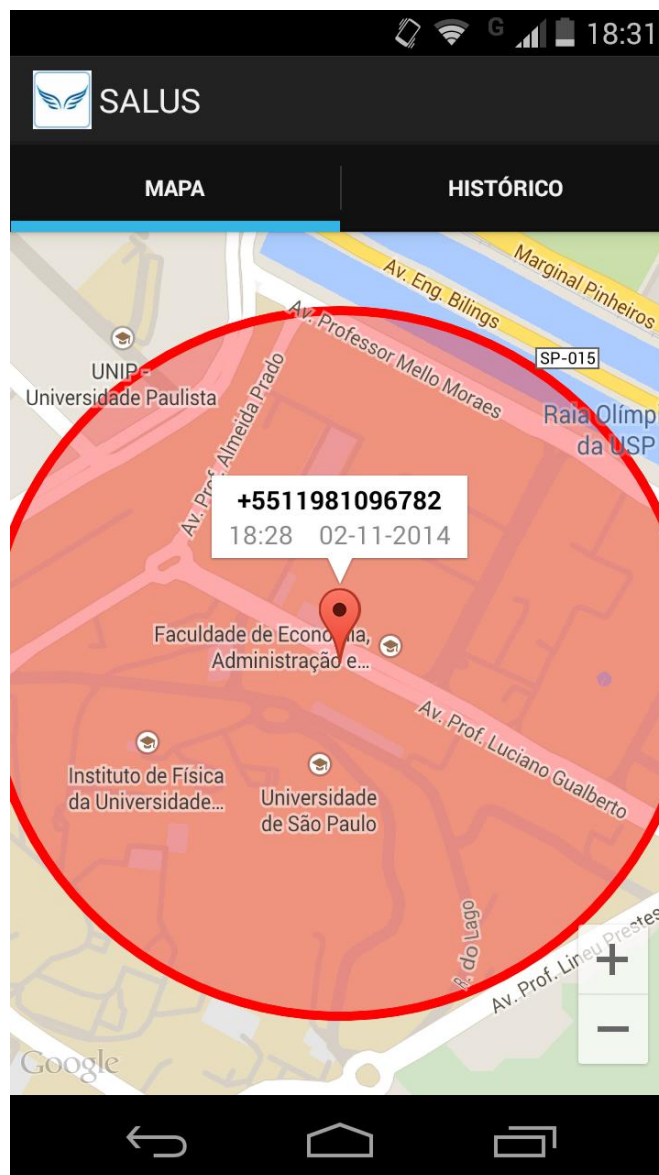
4.2.3.3 MAPA

O local de queda será marcado em um mapa. Além do marcador no local das coordenadas geográficas, será mostrado também um círculo com raio igual ao da precisão, ou seja, indicando a área na qual é possível ter havido a queda. No mapa será possível visualizar os últimos locais de queda registrados.

Para a implementação do código consultou-se diversos tutoriais, cada um ensinando coisas diferentes como (i) aplicativo de mapa que mostra marcadores, (ii) aplicativo utilizando MapsFragment (mais moderno) e (iii) código para deixa o marcador sempre visível (GOOGLE DEVELOPERS, 2014), (TSAGKLIS, 2013) e (STACKOVERFLOW, 2013).

O fragmento implementado possui a seguinte aparência:

Figura 26 - Visualização do marcador no mapa



Fonte: HOSHINO e WAI (2014)

4.2.3.4 BANCO DE DADOS

Para permitir que os dados sejam persistentes, ou seja, que não sejam perdidos mesmo que o aplicativo seja fechado ou que o smartphone seja desligado, optou-se pela utilização de um banco de dados. O próprio Android já disponibiliza bancos de dados baseados em SQLite.

Para a implementação desta parte do aplicativo baseou-se na documentação do próprio Android e em tutoriais disponíveis na internet (GOOGLE DEVELOPERS, 2014) e (VOGEL, 2014).

O banco de dados foi utilizado para armazenar os eventos de queda e as coordenadas geográficas e a precisão. Cada entrada no banco de dados possui os seguintes parâmetros:

- ID: Número sequencial atribuído pela própria biblioteca
- CELLPHONE: Número do chip do dispositivo que enviou o alerta
- DATE: Data e hora de recebimento do alerta
- LATITUDE: Coordenada do local
- LONGITUDE: Coordenada do local
- ACCURACY: Precisão das coordenadas
- MESSAGE: Texto do SMS

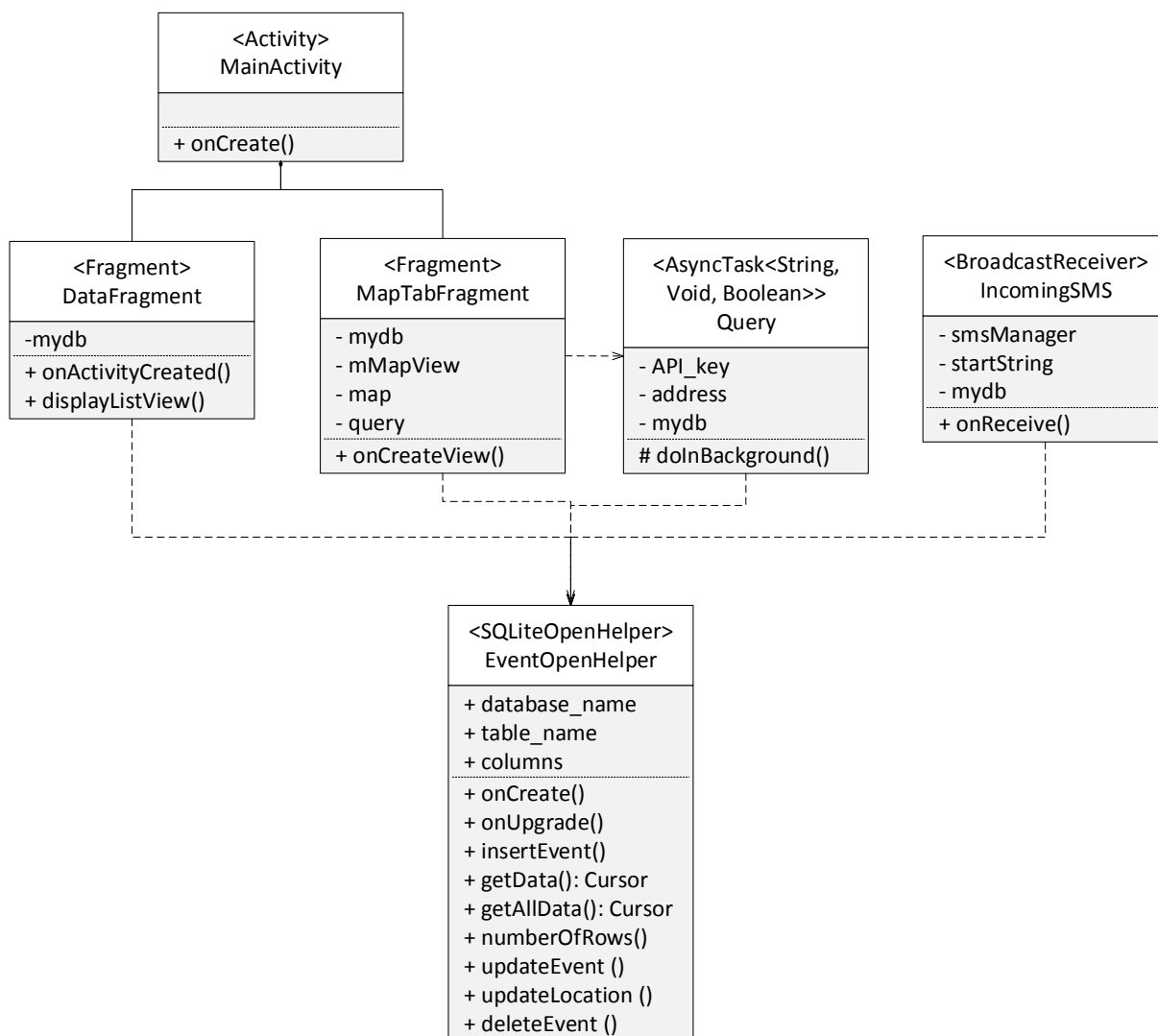
Quando uma nova mensagem de queda é recebida ela é cadastrada no banco de dados com os campos Cellphone e Date. Assim que o aplicativo é aberto, como após um clique na notificação ou um clique no ícone do aplicativo, este busca preencher os dados das coordenadas, enviando *queries* à API. Assim que se obtêm os dados, o evento é atualizado, e é possível visualizá-lo no mapa.

No aplicativo, também é possível visualizar todos os eventos que já ocorreram, em um histórico no qual é apresentado o Cellphone e a Date dos eventos.

4.2.3.5 CLASSES

Segue abaixo o diagrama de classes do aplicativo. Foram representadas somente as funções específicas à funcionalidade deste aplicativo.

Figura 27 - Diagrama de classes do aplicativo



Fonte: HOSHINO e WAI (2014)

As classes utilizadas no aplicativo serão descritas, em linhas gerais, a seguir:

a) MainActivity

Esta é a classe com a qual o aplicativo é iniciado. Ela contém os fragmentos **DataFragment** e **MapTabFragment**, sendo responsável por criá-los, colocá-los em abas, e mostrá-los.

b) DataFragment

Neste fragmento é mostrado o histórico de eventos já registrados. Para obter este histórico é realizada uma query ao banco de dados, trazendo todas as entradas deste.

Dentro desta classe cria-se um objeto do tipo EventOpenHelper, com o qual se obtém todas as entradas do banco de dados, através do método `getAllData()`. Esse método retorna um Cursor que é então utilizado para percorrer todos os eventos, inserindo-os, um a um, em uma lista que será mostrada na tela do celular.

c) MapTabFragment

Neste fragmento, que é mostrado assim que o aplicativo é aberto, exibe-se o mapa com os marcadores dos eventos. Este fragmento também acessa banco de dados através de um objeto do tipo EventOpenHelper, com o qual ele obtém os dados dos últimos n eventos. Há uma restrição quanto ao número de eventos para evitar que o mapa fique sobrecarregado de informações visuais. Para fins de teste e desenvolvimento, $n = 5$.

Com a obtenção das coordenadas geográficas a partir do banco de dados, cria-se um marcador, da classe Marker, para indicar o local do evento e também mostrar a data e o número do chip do dispositivo que sofreu a queda.

Após o posicionamento do marcador cria-se um círculo, da classe Circle, centralizado no marcador, de raio, em metros, igual à precisão salva no banco de dados.

d) IncomingSMS

Para poder receber os SMS o aplicativo deve cadastrar um SMS Broadcast Receiver. Quando um evento de SMS Broadcast ocorrer uma classe do aplicativo será chamada para lidar com o evento. Esta classe é a IncomingSMS.

Inicialmente, esta classe verifica se o SMS recebido corresponde a um SMS de alerta de queda. Caso negativo, este SMS é ignorado. Caso positivo, um evento é gravado no banco de dados, indicando o número do chip do dispositivo e a data de recebimento do alerta. Além disso, a mensagem é salva no campo “Message” e,

posteriormente, quando o aplicativo for aberto, a mensagem será tratada e será feita uma *query* à Geolocation API, de modo a obter as coordenadas do local.

Esta classe também é responsável por exibir uma notificação quando receber um SMS de alerta de queda, de modo chamar a atenção do cuidador para o ocorrido. Essa notificação é criada utilizando-se a classe NotificationManager.

e) Query

Esta é a classe responsável por realizar a *query* à Geolocation API. Nesta classe os dados da mensagem são tratados, ou seja, os valores em hexadecimais são passados para decimais. Além disso, todos os dados disponíveis são colocados em objetos JSON no formato especificado pela API.

Os dados são então enviados e a resposta informa quais as coordenadas e a precisão, em metros, do local do evento. O banco de dados é, então, atualizado com esses dados.

f) EventOpenHelper

Esta classe é a responsável por cuidar do banco de dados, estendendo a classe SQLiteOpenHelper. Ela deve criar o banco de dados e suas tabelas e criar, atualizar, deletar e acessar as entradas das tabelas.

No caso deste projeto foi implementada apenas uma tabela, chamada *events*. Os principais métodos implementados nesta classe, e que podem ser acessados pelas outras classes do aplicativo, são:

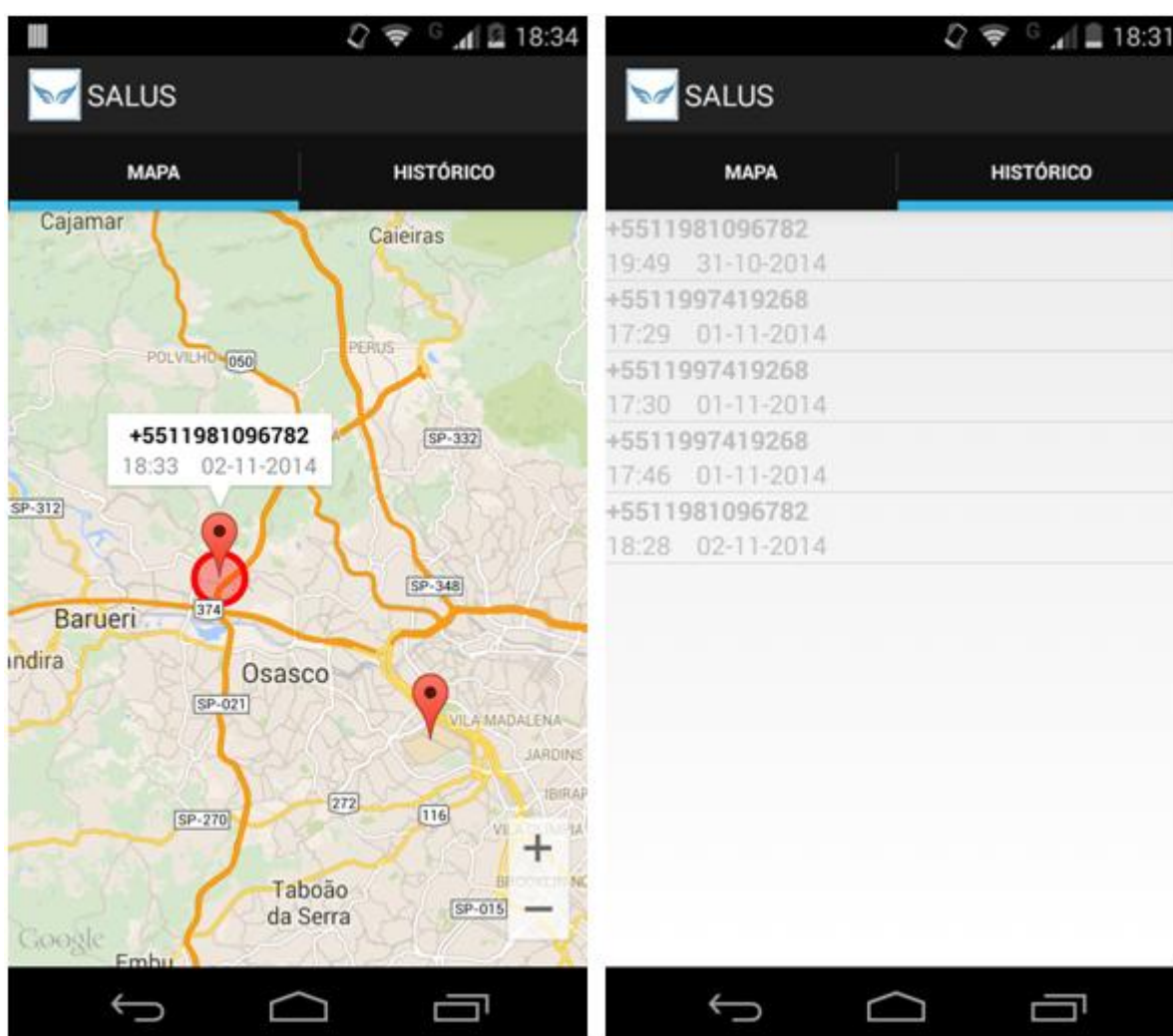
- `public boolean insertEvent (String cellphone, String date, String lat, String lng,String acc)`: permite a criação de uma nova entrada para a tabela.
- `public Cursor getData(int id)`: obtém a entrada referente a um Id específico.
- `public Cursor getAllData()`: obtém todas as entradas da tabela *events*.
- `public int numberOfRows()`: informa qual a quantidade de entradas.
- `public boolean updateEvent (Integer id, String cellphone, String date, String lat, String lng,String acc)`: Atualiza um evento.

4.2.3.6 INTERFACE

A interface do aplicativo é simples e direta, tanto para facilitar o uso deste como para simplificar o desenvolvimento, já que o aplicativo não é o foco principal deste projeto, sendo apenas uma ferramenta para o objetivo final.

O aplicativo é composto de duas abas, uma na qual é mostrado o mapa com os marcadores dos últimos eventos e outra na qual é mostrado o histórico dos eventos.

Figura 28 - Interface do aplicativo



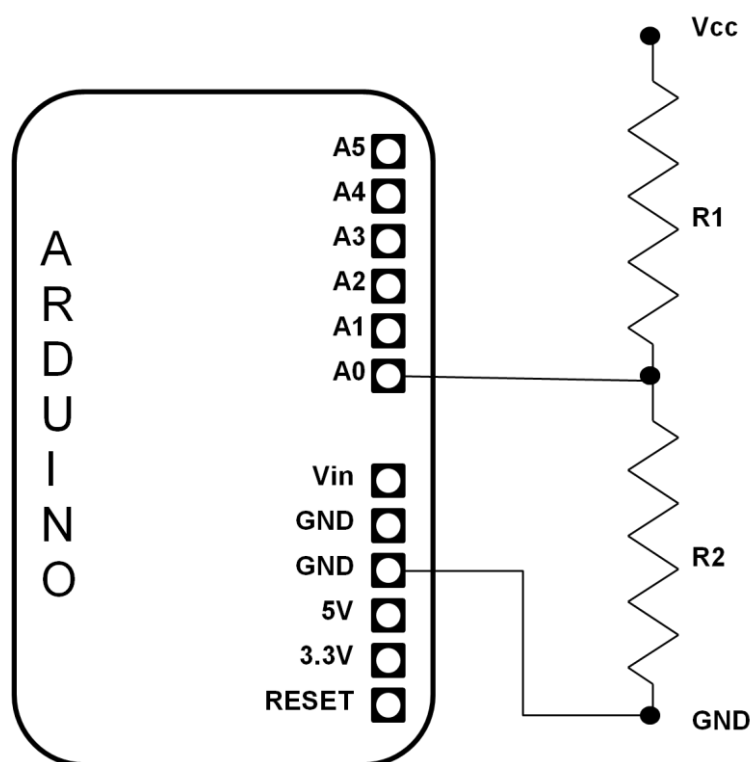
Fonte: HOSHINO e WAI (2014)

4.2.4 MEDIDOR DE BATERIA

Utilizando os pinos de sinais analógicos do Arduino, é possível fazer medições de voltagem para cargas conectadas no mesmo. Como a quantidade de carga que deve ser aplicada no Arduino para que este funcione é de 5V, optou-se por baterias com mais de 5V. No entanto, se uma carga maior que 5V for ligado em uma das entradas analógicas do Arduino, o mesmo corre o risco de ter o seu circuito queimado.

Assim, para contornar esse problema, utilizou-se um divisor de tensões (Figura 29) para que fosse possível submeter os pinos analógicos do Arduino a cargas abaixo dos 5V e ainda assim, fazer a medição correta da bateria. (MARGOLIS, 2011)

Figura 29 - Montagem do divisor de tensão da bateria.



Fonte: HOSHINO e WAI (2014)

Para se definir quais seriam as resistências dos resistores, consultou-se uma tabela fornecida por (MARGOLIS, 2011) (Tabela 14).

Tabela 14 - Tabela para a construção de divisor de tensão.

Máx. Voltagem (V)	R1 (Ω)	R2 (Ω)	R2/(R1 + R2)	Fator do Resistor (decimal)
5 (em cc)	-	-	-	1023
10	1K	1K	1(1 + 1)	511
15	2K	1K	1(2 + 1)	341
20	3K	1K	1(3 + 1)	255
30	4K (3.9K)	1K	1(4 + 1)	170

Fonte: MARGOLIS (2011)

Depois de definido os valores dos resistores, também foi preciso fazer o calculo da conversão dos valores captados pelo microprocessador e o seu valor real. Para fazer esse calculo, pegou-se o valor máximo, em decimal, que o pino analógico consegue ler, que é 1023, e fez-se a conta:

$$Fator\ do\ Resistor = FR = 1023 * \frac{R2}{(R1+R2)} * 5V$$

Voltagem da bateria

$$= \frac{valor\ lido\ pelo\ pino\ analógico\ (decimal) * Máx.Voltagem}{FR} * \frac{1023}{FR}$$

Com esse valor, e baseando-se na curva de consumo da bateria ao longo do uso, é possível relacionar o valor de voltagem lido pelo microprocessador e a quantidade, em porcentagem, restante da bateria. No entanto, não foi possível encontrar uma curva de descarga, de modo que apenas a funcionalidade de detecção de nível crítico de bateria foi testada, mas a leitura efetivada bateria não foi implementada no projeto.

4.2.4.1 SOFTWARE DO MEDIDOR DE CARGA DA BATERIA

Para fazer o medidor da carga da bateria, foi necessário configurar o pino 0 analógico como entrada, explicitar a carga referencia que deve ser utilizada no calculo da voltagem lida pelo microprocessador e fazer as contas do dado lido para a porcentagem restante de bateria, baseando-se na curva de consumo levantada experimentalmente com a bateria utilizada no projeto.

Para que o nível crítico da bateria fosse identificado e uma mensagem fosse enviada para o celular do cuidador, foram criadas duas flags: uma que indica o nível baixo de bateria, lowBat, e outra, SMSBat, que indica se a mensagem já foi enviada ou não, o que evita que seja enviada mais que de uma mensagem enquanto o nível crítico de bateria seja mantido. Para o projeto, foi definido que um valor próximo de 5V já pode ser considerado um nível crítico de bateria, visto que 5V é o mínimo de carga que o dispositivo deve ser alimentado para funcionar corretamente. A condição para que ambas as flags sejam limpas é a de que a bateria alcance pelo menos 80% de sua carga total, quantia suficiente para que ela possa durar por um tempo razoavelmente bom. Além disso, foi colocado também um led auxiliar que sinaliza visualmente quando o dispositivo encontra-se com pouca bateria. O led apenas se apaga quando a flag de nível crítico de bateria, lowBat, for limpa.

4.2.4.2 CONSUMO DE BATERIA

Apenas o Arduino exige uma corrente de aproximadamente 47,5 mA. Com esse consumo, uma bateria de 2000 mAh, facilmente encontrada em smartphones atuais, duraria pouco menos de 2 dias. Essa duração inviabiliza o uso do dispositivo em sua proposta inicial, mas, como discutido em 4.6 Análise dos resultados, é possível utilizar a estrutura e funcionamento de um Arduino com correntes da ordem de 1 mA, o que aumentaria muito a duração da bateria.

4.3 TESTES DOS MÓDULOS

Antes de fazer a integração completa de todos os módulos, primeiramente foram realizados testes individuais dos mesmos de modo a garantir o seu funcionamento e a sua comunicação com o Arduino e para também para poder refinar os códigos escritos neles eles, tanto em questões de funcionalidade, quanto de velocidade e memória. A seguir é mostrado os componentes e os testes feitos de acordo com sua funcionalidade.

4.3.1 MÓDULO GSM

Para o teste das funções implementadas, descritas anteriormente, criou-se um programa que é explicado no Apêndice A. O intuito desse programa era chamar diferentes funções a partir de certas teclas.

Com ele, foi possível identificar e corrigir alguns erros.

a) Buffer limitado a 64 bytes

Como mostrado anteriormente, o comando AT+CENG? retorna uma grande listagem de dados. Devido à pouca memória do Arduino, a própria biblioteca SoftwareSerial limita o buffer de transmissão da serial a 64 bytes. Desse modo, quando a resposta era lida, ela não cabia toda no buffer.

Este problema foi corrigido aumentando-se o tamanho do buffer para 256 bytes. Com 256 bytes ainda não é possível obter toda a resposta do comando AT+CENG? mas já é possível obter os dados de pelo menos 5 antenas (HOBBYTRONICS, 2014).

```
+CENG: 1,1
```

```
+CENG:0,"0130,49,99,724,10,09,35e1,15,05,2527,255"
```

```
+CENG:1,"0789,43,09,35e4,724,10,2527"
```

```
+CENG:2,"0129,32,255,ffff,000,00,0"
```

```
+CENG:3,"0132,33,44,35e0,724,10,2527"
```

```
+CENG:4,"0134,32,22,ffff,724,10,2527"
```

```
+CENG:5,"0790,28,255,ffff,000,00,0"
```

```
+CENG:6,"0788,27,255,ffff,000,00,0"
```

```
OK
```

b) Ajuste dos tempos de espera

Os testes foram importantes para se ajustar os tempos de espera entre comandos. Há casos em que, caso não haja uma espera, o comando não é bem-sucedido. Um

exemplo ocorre no envio de SMS. Após o primeiro comando, indicando que se quer enviar um SMS, deve-se esperar um tempo (por volta de 3s) para enviar o texto da mensagem. Caso não haja essa espera, erros foram gerados, já que módulo GSM não conseguia distinguir o que era comando do que era mensagem.

4.3.2 UNIDADE DE PROCESSAMENTO DE MOVIMENTO

O módulo MPU-6050 é um componente muito abrangente, o que permite que muitas funcionalidades sejam implementadas com ele. No entanto, como foi mencionado anteriormente, infelizmente o material que se tem disponível atualmente é escasso. Não foi possível descobrir o motivo, mas notou-se que a cada versão nova do datasheet que era disponibilizado, informações eram retiradas do mesmo, mais especificamente das interrupções do módulo. Para essas interrupções, só foi encontrado um único programa que foi utilizado como referência para o entendimento do funcionamento do módulo. Assim, além do estudo do programa, fez-se necessário a realização de testes do mesmo para de fato compreender e viabilizar o uso do módulo.

Porém, apesar de se conseguir gerar corretamente as interrupções de acordo com os parâmetros limiares estabelecidos, houve problemas em se compreender o registrador de decremento do contador, registrador 105, e o registrado 30 de contador.

4.3.2.1 LEITURA DE DADOS

Com base em um programa fornecido pela comunidade do Arduino, foi possível configurar a comunicação entre o módulo e o microprocessador de maneira adequada. Em seguida, com o uso de bibliotecas do próprio Arduino de comunicação serial, foi possível imprimir os dados lidos pelo MPU-6050 no terminal. Feito isso, tomou-se dois cuidados: verificar a velocidade máxima de comunicação que era possível estabelecer entre os dois componentes e validar se os dados obtidos de fato eram condizentes. Com relação à velocidade máxima de comunicação, conseguiu-se chegar a velocidade de 115200 bps. Já com relação aos

dados obtidos, como cada leitura era feita através de 16 bits, foram feitos testes com o módulo configurado para operar em um intervalo de $\pm 2g$ com uma frequência de 15 Hz, que foi o a frequência que se conseguiu alcançar devido ao tempo de processamento dos dados que o Arduino tomou. Os testes consistiram em soltar o dispositivo de uma altura de aproximadamente 1,5 metros de altura, tomando o cuidado de se manter o eixo x do módulo perpendicular ao chão, e gravar os valores obtidos em um arquivo no formato de csv para que posteriormente fossem feitas as validações. Abaixo segue uma amostra dos dados coletados.

Tabela 15 - Leitura de dados do MPU-6050 em queda.

Tempo(ms)	Acc_X	AccY	AccZ
398	16.128	- 4.000	1.296
585	15.716	- 4.216	960
654	16.308	- 4.180	1.056
723	15.728	- 4.100	1.032
795	15.484	- 5.208	872
864	15.684	- 4.956	1.304
934	15.060	- 3.584	2.272
1.005	16.500	- 4.048	2.048
1.075	16.468	- 2.296	736
1.144	18.300	- 2.208	3.248

Fonte: HOSHINO e WAI (2014)

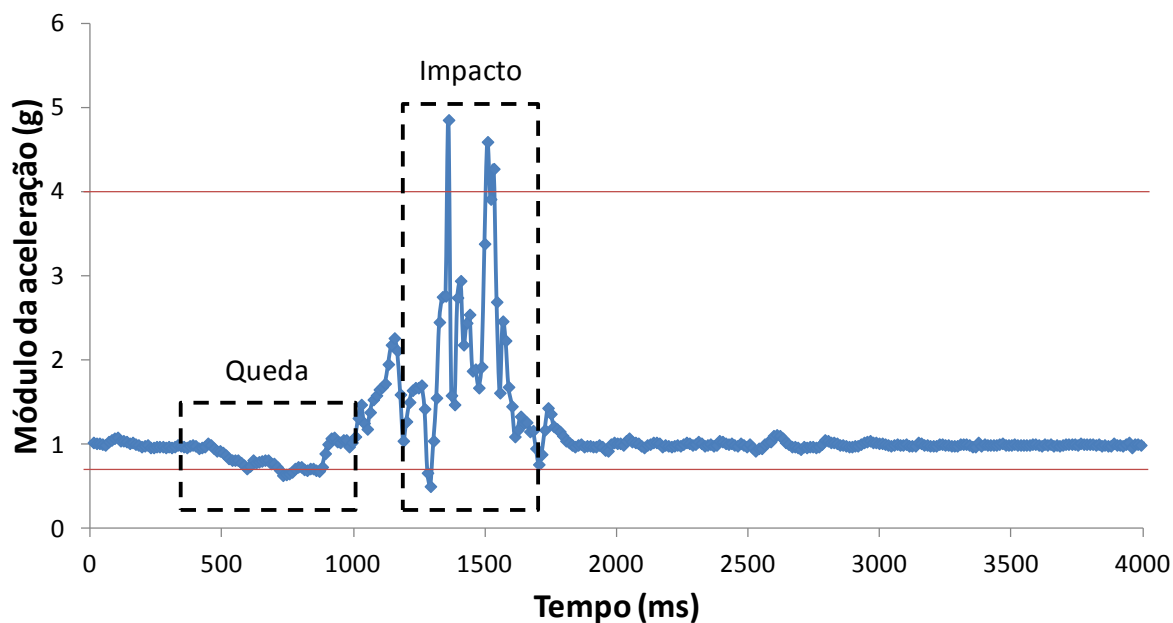
Dos dados obtidos, verificou-se que os dados, de fato, são condizentes com a realidade, visto que, como o módulo utiliza 16 bits para faz a leitura dos dados e que tal valor possui sinais positivos e negativos, tem-se que 16 bits variam entre ± 32.675 . Assim, quando o acelerômetro é submetido unicamente a ação da gravidade, a leitura no eixo x variou perto do valor de 16.383. Como o intervalo de operação foi de $\pm 2g$, então 2g é aproximadamente 32.675 e 1g é aproximadamente 16.383.

4.3.2.2 QUEDAS

Para elaborar o algoritmo de detecção de queda, foi necessário fazer uma outra rodada de testes de quedas com a aquisição dos módulos de velocidade. Com esses

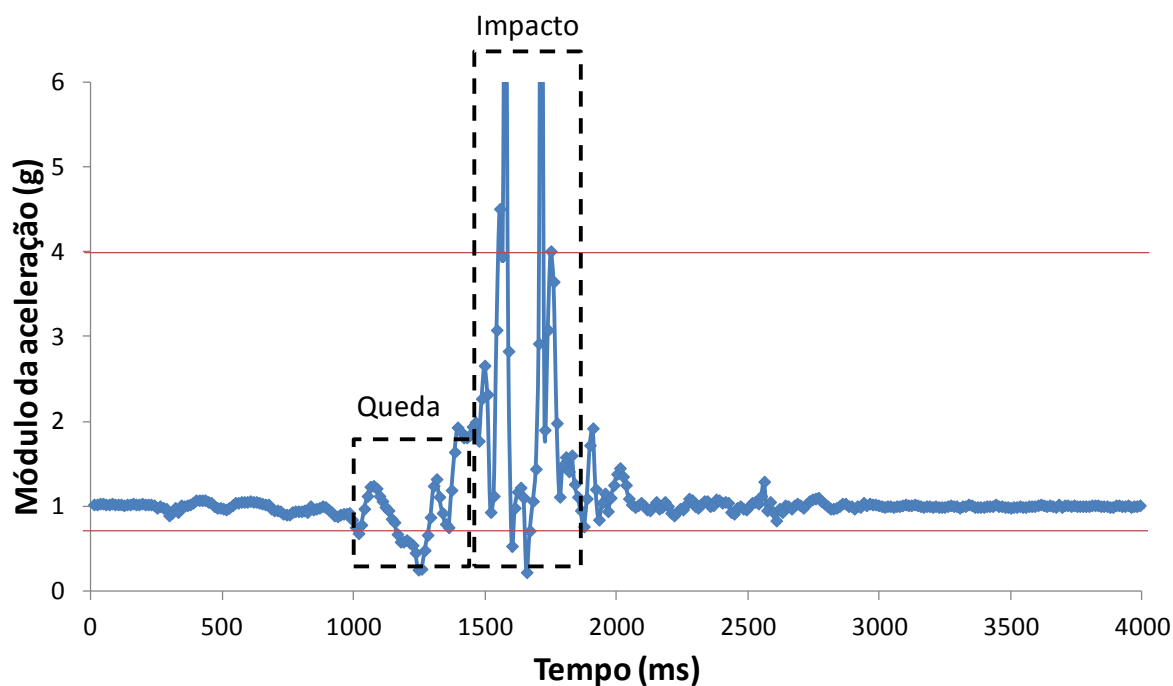
testes, foi possível identificar o comportamento das curvas de aceleração que caracterizam o acidente de queda. Para esses testes, foram três quedas repetidas para quatro diferentes tipos de queda: frontal, lateral, de costas e sentado. Os gráficos obtidos encontram-se a seguir.

Figura 30 - Queda de costas.



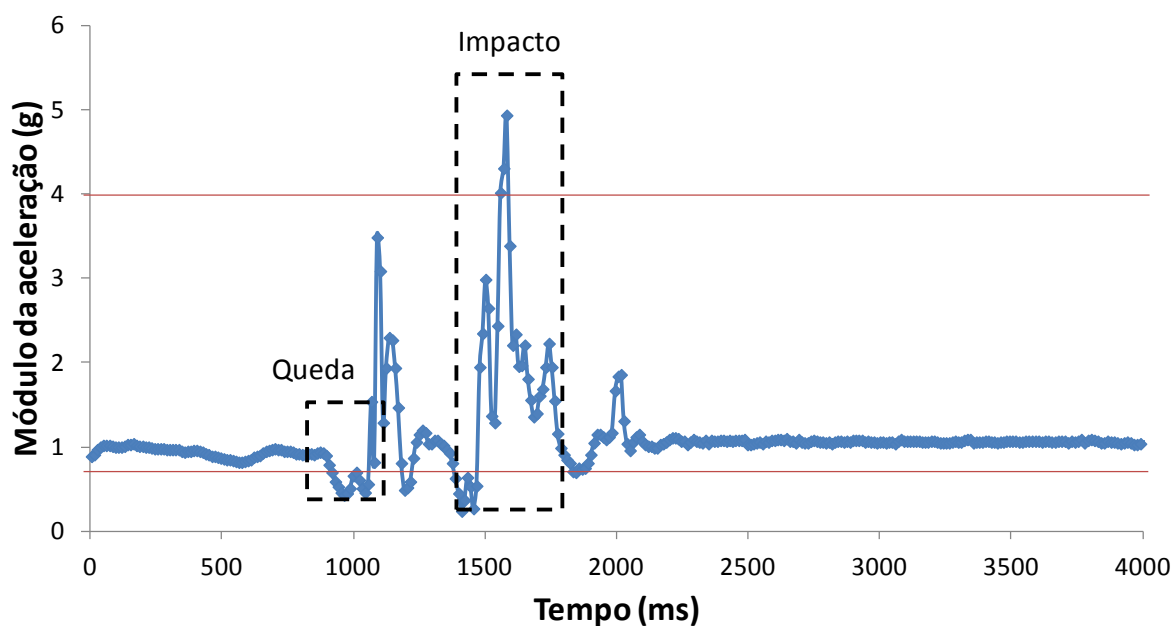
Fonte: HOSHINO e WAI (2014)

Figura 31 - Queda lateral.



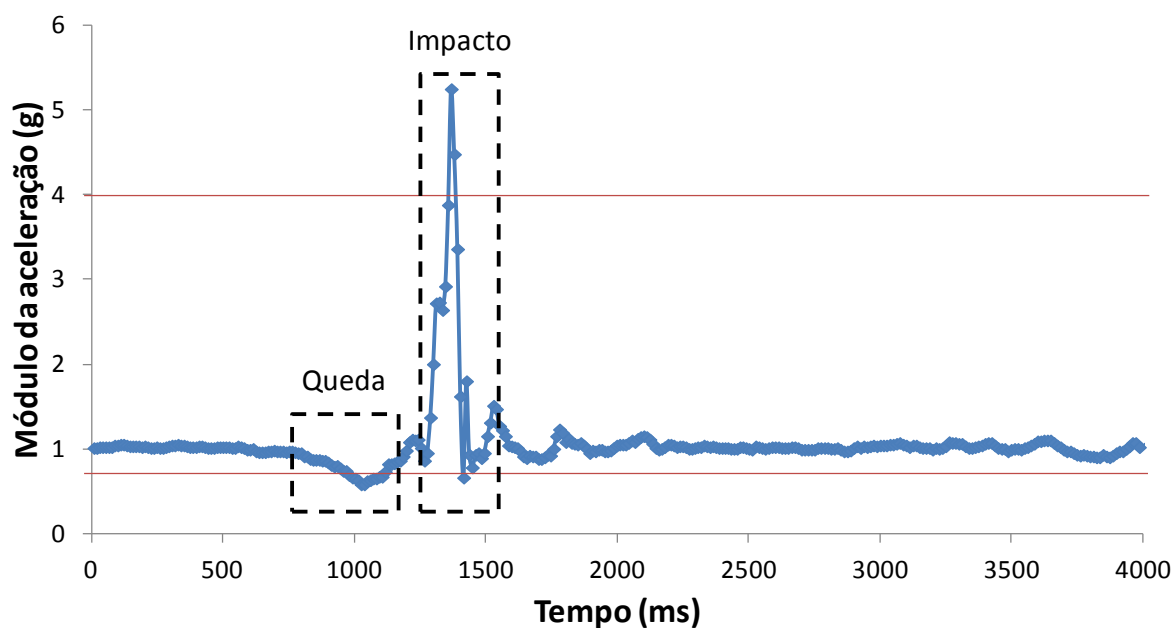
Fonte: HOSHINO e WAI (2014)

Figura 32 - Queda frontal.



Fonte: HOSHINO e WAI (2014)

Figura 33 - Queda sentado.



Fonte: HOSHINO e WAI (2014)

Analisando as curvas das quedas, pode-se notar que todas apresentam um perfil muito similar. Tal perfil pode ser caracterizado pela presença de um surgimento de um vale, seguido pelo aparecimento de um pico em um intervalo relativamente curto.

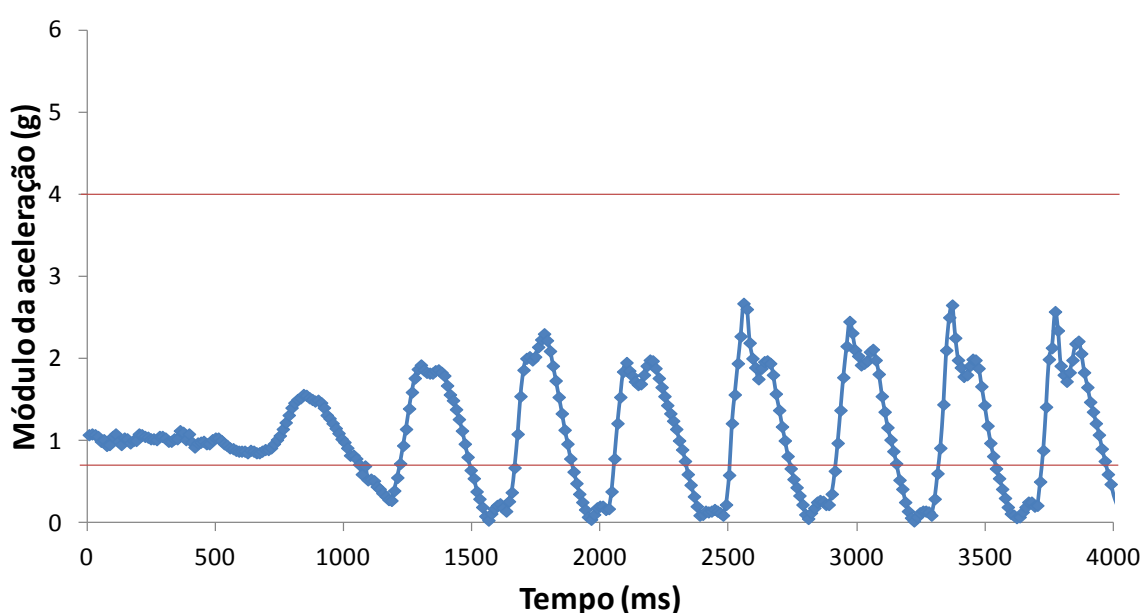
Assim, esses foram os parâmetros escolhidos para se detectar queda: presença de um vale, seguido pela presença de um pico, em um intervalo de tempo.

Escolhido os parâmetros a serem monitorados para se detectar uma queda, escolheu-se então, os parâmetros limiares. Para isso, analisando as 12 diretas quedas, pegou-se o maior vale obtido (valor de 0.64 g), o menor pico obtido (3.7 g) e o maior tempo entre um vale e um pico (801 ms) e definiu-se eles valores arredondados deles como sendo parâmetros limiares. Assim, bastou comparar com atividades do cotidiano para validar se esses parâmetros limiares eram suficientes para se identificar quedas exclusivamente.

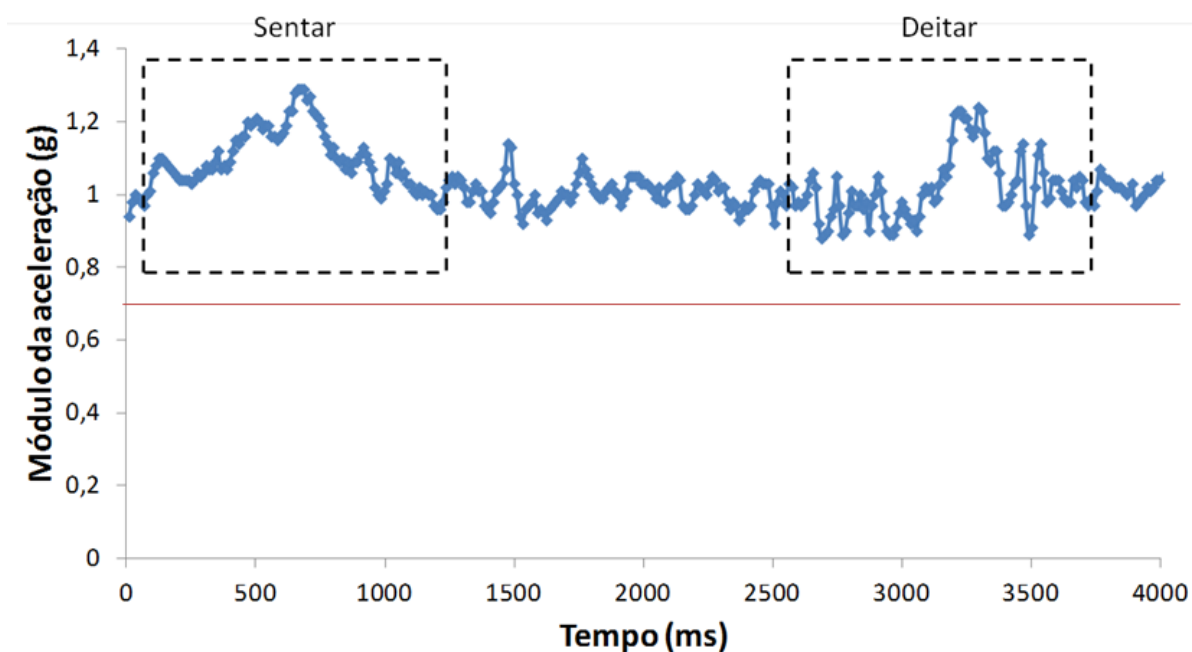
4.3.2.3 ATIVIDADES DO COTIDIANO

Foram realizados também ensaios de atividades do cotidiano e coletando as acelerações envolvidas. Isso porque acidentes de quedas não podem ser confundidos com atividades do cotidiano. Se confundidas, ou o dispositivo pode acabar enviando diversos alertas falso-positivos, comprometendo a confiabilidade do dispositivo ou, pior, pode ser incapaz de detectar uma queda verdadeira. Para as atividades simuladas para esse teste, considerou-se: andar, correr, sentar e deitar. A seguir, têm-se os gráficos obtidos.

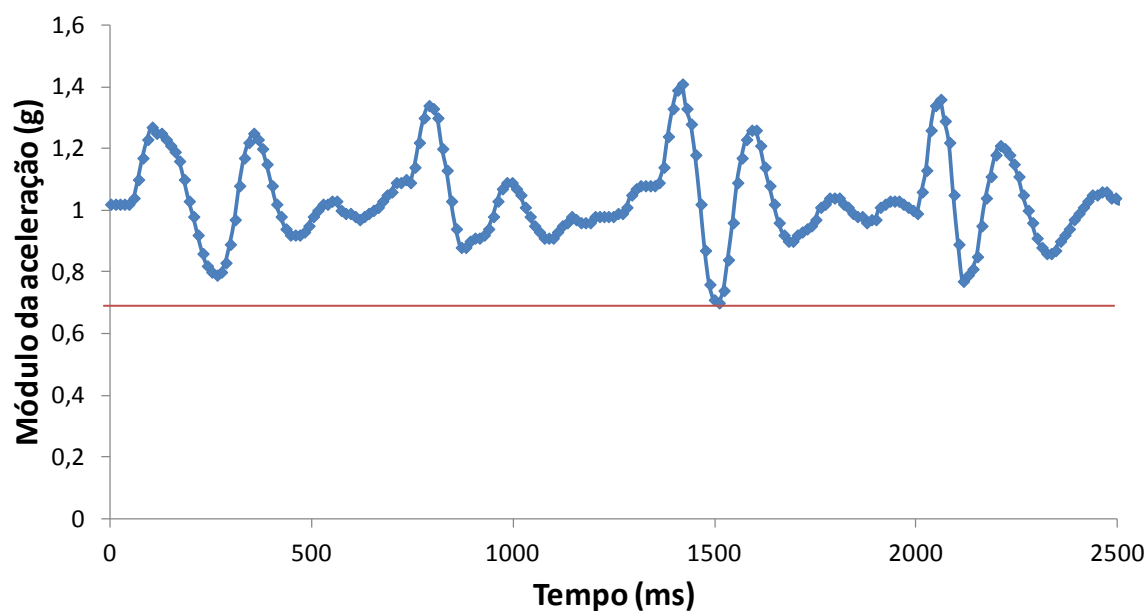
Figura 34 - Atividade de correr.



Fonte: HOSHINO e WAI (2014)

Figura 35 - Atividade de sentar e deitar

Fonte: HOSHINO e WAI (2014)

Figura 36 - Atividade de caminhar.

Fonte: HOSHINO e WAI (2014)

Analisando os gráficos obtidos, percebe-se que apenas a atividade de correr possui um perfil de comportamento parecido com o identificado com a queda, ou seja, a presença de vales, seguido por picos, em um curto intervalo de tempo. No entanto,

em nenhum dos casos, o valor da aceleração conseguiu atingir o parâmetro limiar estabelecido no código, 4.0 g (os parâmetros limiares para vale e pico estão representados pelas linhas vermelhas nos gráficos). Assim, pode-se garantir que nenhuma dessas atividades será confundindo com um acidente de queda.

4.3.3 APLICATIVO

Na implementação do aplicativo, este foi criado por partes que foram posteriormente integradas. Diversos testes foram realizados ao longo da implementação, sendo a maioria testes que verificam a funcionalidade primária do código. No entanto, alguns testes apresentaram resultados que levaram a alterações no aplicativo ou mesmo no projeto.

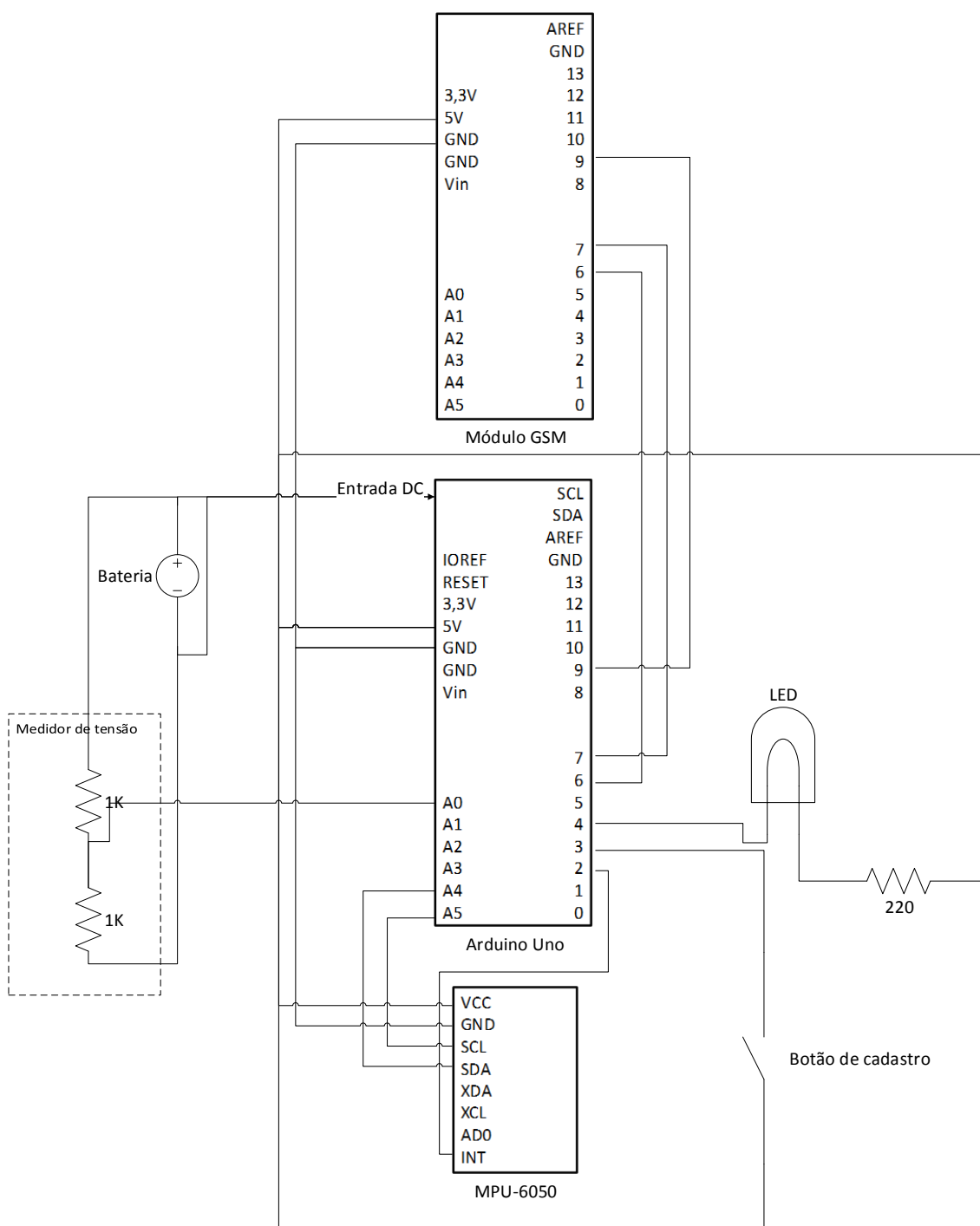
Inicialmente a *query* à Geolocation API era realizada unicamente assim que o SMS era recebido. No entanto, percebeu-se que este método não era o ideal, já que, caso o celular não possuísse conexão com a internet no momento do recebimento do SMS, a *query* falharia e não se tentaria novamente. Isso foi corrigido, e atualmente o aplicativo lida com os SMS e as *queries* conforme descrito anteriormente.

4.4 INTEGRAÇÃO

O protótipo final integrou o módulo GSM, o MPU-6050 e o Arduino Uno, além de contar com um LED para *feedback* e um botão para acionar o cadastramento do número de celular do cuidador.

A pinagem utilizada é mostrada a seguir.

Figura 37 - Pinagem do dispositivo



Fonte: HOSHINO e WAI (2014)

No entanto, até a obtenção do protótipo final, diversos protótipos foram realizados, cada um com objetivos predefinidos, tal como mostrado na Tabela 16.

Tabela 16 - Protótipos

Protótipo	Objetivos	Melhorias necessárias
1	✓ Envio de SMS ✓ Leitura do MPU-6050 ✓ Comunicação dos módulos com Arduino	✓ Aperfeiçoamento da detecção de queda
2	✓ Uso da interrupção do MPU-6050 ✓ Liga/desliga automático	✓ Otimização da rotina de interrupção
3	✓ Simplificação da interrupção ✓ Obtenção dos dados das antenas ✓ Localização pelo aplicativo	✓ Aprimorar localização ✓ Reconsiderar o uso das interrupções
4	✓ Cadastro de número de celular ✓ Detecção de queda através da leitura de dados	✓ Validação do cadastro
5	✓ Uso da bateria ✓ Detecção de nível crítico de bateria	✓ Calibração do monitoramento de bateria

Fonte: HOSHINO e WAI (2014)

4.4.1 PROTÓTIPO 1

No primeiro protótipo do projeto, foram testadas as comunicações dos módulos com o Arduino e as funcionalidades de leitura dos dados do acelerômetro presente no MPU-6050 e o envio de SMS através do módulo GSM. Os testes desse protótipo consistiram em um algoritmo que fazia a leitura das acelerações do MPU-6050, calculava o módulo da raiz da soma dos quadrados das acelerações de cada eixo e as comparava com um valor pré-definido de aceleração que caracterizava queda. Em casos que o módulo lido fosse maior que a aceleração de queda, era enviada um SMS para um número pré-cadastrado no sistema indicando uma mensagem de queda.

Como resultado, todas as comunicações funcionaram de modo adequado e sempre que se submetia o dispositivo a uma situação de queda, era enviado um SMS. No

entanto, simplificar a detecção de queda em apenas uma comparação entre os módulos de aceleração se mostrou ser um método muito simples, visto que outras atividades também eram confundidas com queda. Assim, foram feitos mais estudos para se utilizar as interrupções que o módulo MPU-6050 é capaz de emitir através da configuração de seus registradores.

4.4.2 PROTÓTIPO 2

No segundo protótipo, foram testados os comandos de liga e desliga através do Arduino (anteriormente, essas funções estavam sendo feitas através do botão do módulo GSM) e foi testada a detecção de quedas através de interrupções de queda livre presente no módulo MPU-6050.

Como resultado, o comando de desligar via Arduino funcionou, no entanto, enfrentaram-se problemas para o comando de ligar. Posteriormente, descobriu-se que era preciso fazer ajustes nas conexões entre o módulo GSM e o Arduino para que o comando de ligar funcionasse.

Com relação aos testes de interrupção de queda livre, encontraram-se duas grandes dificuldades: a compreensão do funcionamento das interrupções de queda livre do módulo MPU-6050, visto que os atuais datasheets do MPU-6050 pouco comentam ou sequer citam a existência de tal interrupção e problemas com a rotina de interrupção do Arduino. A primeira dificuldade foi contornada parcialmente através de testes da própria interrupção e da consulta de uma biblioteca disponível na comunidade do Arduino. Com isso, foi possível gerar interrupções de acordo com acelerações impostas ao dispositivo. Já a segunda dificuldade, o problema deveu-se às limitações que a rotina de interrupção do Arduino possui. Ela é incapaz de realizar funções muito complexas como ligar o módulo GSM, por exemplo. Para contornar esse problema, limitou-se a rotina de interrupção para o acionamento de flags que são utilizadas na rotina principal do programa.

4.4.3 PROTÓTIPO 3

No terceiro protótipo, testou-se a rotina de interrupção simplificada para a detecção de queda, a obtenção dos dados das antenas através do módulo GSM e a localização do dispositivo através desses dados com o uso do aplicativo desenvolvido para o celular, receptor final do alerta de queda via SMS.

Como resultado, foi possível detectar algumas quedas a partir dos testes da geração de interrupção do MPU-6050 e da rotina de interrupção otimizada no Arduino. No entanto, identificou-se um comportamento não uniforme do registrador de contagem, 30. Seu comportamento por vezes era linear e por vezes não-linear. Não apenas isso, mas o registrador 105 também apresentou comportamentos incompreensíveis em testes similares. Desse modo, reconsiderou-se o uso das interrupções para a detecção de queda.

Para a obtenção dos dados das antenas, foi possível obter dados de até 5 antenas próximas ao local do teste. No entanto, houve dificuldade em se obter os dados de cell ID de mais de uma antena, sendo possível obter esse dado apenas da antena à qual o módulo estava conectado. Felizmente, utilizando o próprio algoritmo de obtenção dos dados de antena, foi possível obter os Cell-IDs das outras antenas ao se inserir um parâmetro a mais na chamada da função.

Com relação à localização através do uso do aplicativo desenvolvido a partir da API do Google, o envio dos dados das antenas obtido pelo módulo GSM conseguiu ser enviado corretamente para um celular através do envio de SMS, o aplicativo conseguiu extrair corretamente os dados enviados e a localização conseguiu ser feita em uma representação gráfica disponibilizada pelo Google (Google maps) de modo adequado. Porém, a precisão alcançada nos testes foi muito baixa, cerca de 1 km. Tal precisão não condizia com a documentação oferecida pelo Google a respeito de sua API. Assim, foi preciso fazer testes para poder compreender o motivo de tal discrepância.

4.4.4 PROTÓTIPO 4

O quarto protótipo focou-se em testar a funcionalidade de cadastro de números de celular a partir do recebimento de SMS e o novo algoritmo de detecção de queda a

partir da análise de dados coletados pelo acelerômetro. Para a primeira funcionalidade, foi verificado se o dispositivo era capaz de receber SMS, extrair o número do celular do qual o SMS foi enviado e armazenar esse número em sua memória para que todos os SMS enviados posteriormente fossem direcionados para esse número cadastrado. Também para esse protótipo, foi inserido um botão responsável por ativar o modo de cadastro do dispositivo e um LED que indicasse o momento em que o dispositivo já estivesse preparado para receber o novo número cadastrado.

Para esse protótipo, identificou-se que o dispositivo não fazia nenhuma validação do SMS recebido durante o modo de cadastro. Com isso, o primeiro número que enviasse um SMS no momento em que o dispositivo estivesse esperando o recebimento de cadastro seria cadastrado. Assim, foi preciso implementar uma função de validação que só cadastrasse números que enviassem SMS contendo um código específico.

Além disso, devido a maior simplicidade do algoritmo de detecção de queda, o dispositivo conseguiu operar com uma velocidade próxima de 1 KHz, sendo assim, possível implementar um algoritmo de detecção de queda capaz de monitorar no mínimo 4 pontos seguidos de acelerações relevantes, diminuindo assim a quantidade de falso-positivos. O software implementado está descrito na seção 4.2.2.

4.4.5 PROTÓTIPO 5

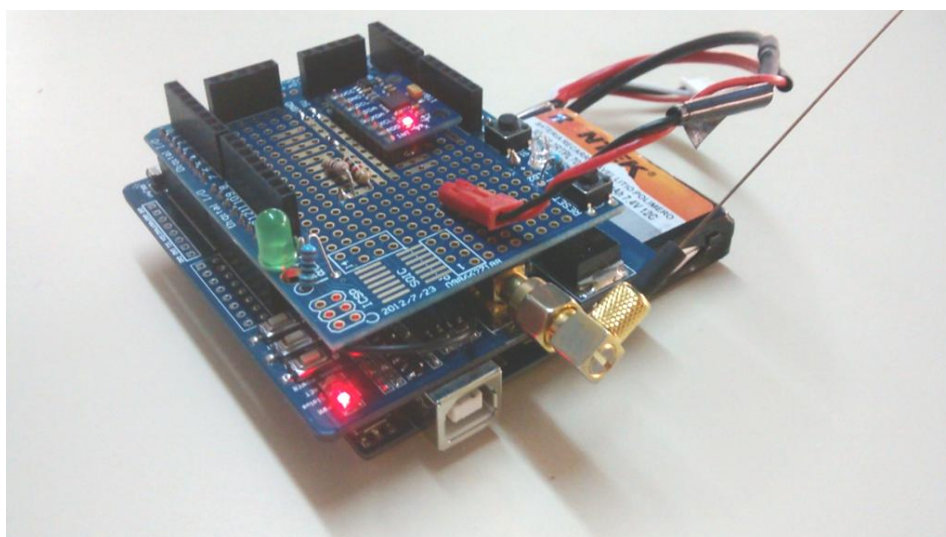
O quinto protótipo limitou-se a testar apenas as funcionalidades relacionadas a bateria. Assim, foi testado se o dispositivo conseguia operar a partir da alimentação de uma bateria de 7.4V, se o dispositivo era capaz de fazer medições do nível da bateria, se o dispositivo era capaz de enviar um SMS para um número cadastrado em casos que a bateria atingisse um nível crítico. Para esses testes, implementou-se um divisor de tensão para que o dispositivo conseguisse fazer medições do nível da bateria sem que corresse o risco de danificar o dispositivo.

Inicialmente, houve diversas falhas nos testes. Isso porque os valores lidos pelo dispositivo e imprimidos no terminal do computador variavam de maneira aleatória. Posteriormente, descobriu-se que o motivo para que as leituras do nível de bateria

estivessem erradas era devido ao fato do dispositivo estar conectado no computador. Por causa disso, a tensão de referencia utilizada dependia da tensão de saída do computador, que varia continuamente. Assim, foi necessário alterar os testes para que o dispositivo fosse alimentado por um gerador de tensão de corrente continua e um multímetro.

Como resultado, o dispositivo conseguiu indicar um nível critico de bateria quando a tensão fornecida pelo gerador alcançava valores baixos. A indicação de que a bateria alcançara um nível critico foi feita através do acendimento de um LED e o envio de SMS.

Figura 38 - Protótipo final



Fonte: HOSHINO e WAI (2014)

4.5 TESTES

Para o protótipo final, foram realizados diversos testes com o intuito de se avaliar e melhorar as diferentes funcionalidades que o dispositivo apresenta. Durante essa etapa do projeto, foram testadas a capacidade de detectar diferentes tipos de queda, a capacidade de diferenciar quedas de outras atividades, o tempo de duração do cadastro de um número de celular, o tempo entre a detecção de uma queda e o recebimento de um SMS, a precisão da localização da queda no aplicativo implementado e as condições extremas em que não se tem sinal de celular. Para todos os testes, foram montadas tabelas para que suas análises fossem mais fáceis de serem feitas.

4.5.1 TESTE 1 – QUEDAS

Para estes testes de detecção de queda, acoplou-se o protótipo final na altura do peito dos usuários com o uso de um elástico. Isso garantiu que o dispositivo não sofresse acelerações que não fossem referentes ao do corpo. Além disso, para evitar o envio de mensagens em todas as quedas, utilizou-se um LED em seu lugar. O LED simulou o alerta de queda e a cada vez que o dispositivo detectava uma queda, o LED acendia. Para apaga-lo, bastava apertar um botão presente no dispositivo. Além disso, as quedas foram realizadas sobre um colchão, de modo a manter a integridade física das pessoas. A seguir, encontra-se uma tabela contendo os protocolos de queda adotados para os testes.

Tabela 17 - Teste de detecção de queda

Tipo de queda	Dinâmica da queda
Frontal	➤ Queda de frente, ➤ Posição inicial: de pé ➤ Os primeiros membros a atingirem o chão são os joelhos. ➤ Posição final: Corpo deitado no chão, de bruços, mas apoiado com os braços.
De costas	➤ Queda para trás, ➤ Posição inicial: de pé ➤ O primeiro membro a atingir o chão é o quadril. ➤ Posição final: Corpo deitado no chão, com as costas encostando-se ao chão.
Lateral	➤ Queda para os lados direito e esquerdo (foi intercalado), ➤ Posição inicial: de pé ➤ A queda é direta, sem a presença de nenhum membro amortecendo a queda. ➤ Posição final: Corpo deitado no chão, de lado.
Sentado	➤ Queda de costas ➤ Posição inicial: de pé ➤ O primeiro membro a atingir o chão é o quadril e a queda se encerra. ➤ Posição final: Corpo sentado no chão.

Fonte: HOSHINO e WAI (2014)

Os resultados encontram-se na tabela a seguir.

Tabela 18 - Teste de detecção de queda

Tipo de queda	Número de testes	de	Número de detecção de queda	Taxa de acerto (%)
Frontal	5		5	100%
De costas	5		4	80%
Lateral	5		5	100%
Sentado	5		5	100%

Fonte: HOSHINO e WAI (2014)

4.5.2 TESTE 2 – ATIVIDADES DO COTIDIANO

Foram realizados testes para verificar se o dispositivo detectaria alguma queda em atividades do cotidiano. Essas atividades foram especificadas em 3.1.1. Para atividades contínuas como as de caminhar, correr e repouso, foram utilizados intervalos de 30 segundos de atividade contínua. Os resultados dos testes são apresentados a seguir:

Tabela 19 - Teste do dispositivo em atividades do cotidiano

Atividade	Número de testes	Número de Falso – positivo?	Taxa de erro (%)
Caminhar	5	0	0%
Correr	5	0	0%
Sentar	5	0	0%
Deitar	5	0	0%
Levantar	5	0	0%
Repouso	5	0	0%

Fonte: HOSHINO e WAI (2014)

4.5.3 TESTE 3 – TEMPO DE ENVIO DE SMS

Para os testes de envio de SMS pelo dispositivo, foram realizados 5 testes de envio de mensagem em que se mediu o tempo entre o Arduino ligar o módulo GSM e a mensagem chegar no celular. Todos os testes foram realizados no mesmo ambiente (laboratório de microprocessadores da Engenharia Elétrica). Os resultados estão na Tabela 20.

Tabela 20 - Teste de tempo de envio do SMS

Número do teste	Tempo de envio (s)
1	37
2	43
3	67
4	47
5	70
Média	53

Fonte: HOSHINO e WAI (2014)

4.5.4 TESTE 4 – TEMPO DE RECEBIMENTO DE SMS

Para os testes de recebimento de SMS, mediu-se o tempo que se levou entre o envio de SMS através de um celular e o desligamento do led de cadastro. Todos os testes foram realizados no mesmo ambiente (laboratório de microprocessadores da Engenharia Elétrica). Os resultados estão na Tabela 21.

Tabela 21 - Teste de tempo de recebimento do SMS

Número do teste	Tempo de recebimento (s)
1	7
2	12
3	8
4	15
5	17
Média	12

Fonte: HOSHINO e WAI (2014)

Uma ressalva com relação aos testes 3 e 4 é que existem duas condições para que seu funcionamento ocorra corretamente. A primeira é que é necessário que a conta referente ao cartão SIM utilizado esteja com créditos disponíveis e que não esteja bloqueada. Já a segunda condição é que o local deve possuir sinal, do contrário, o dispositivo vai ficar continuamente tentando procurar sinal até encontrá-lo.

4.5.5 TESTE 5 – LOCALIZAÇÃO

Para os testes da localização, focou-se na precisão da localização indicada pelo aplicativo desenvolvido no projeto. Para a obtenção das precisões, foram feitos testes da localização de quatro antenas distintas, próximas ao local dos testes. O motivo para tal metodologia de teste deveu-se ao fato explicado na seção 3.4.1. Os resultados estão na Tabela 22.

Tabela 22 - Testa da precisão da localização

Número do teste	Precisão da localização (m)
1	876
2	835
3	1119
4	1451
Média	1070

Fonte: HOSHINO e WAI (2014)

4.5.6 TESTE 6 – DISPOSITIVO SEM SINAL DE ANTENA

Caso o módulo GSM não possua sinal de celular quando for acionado, este se mantém ligado até conseguir uma conexão, e somente depois de conseguir sinal ele prossegue com sua operação normal, como enviar um SMS. Este comportamento foi verificado nos testes.

4.6 ANÁLISE DOS RESULTADOS

Analisando os dados obtidos dos testes realizados com o protótipo final, notou-se que o dispositivo não gerou falsos-positivos, conseguindo diferenciar bem as atividades cotidianas das quedas. No entanto, uma queda de costas não foi detectada como queda. Isso ocorreu devido ao tipo de queda que foi feito para o teste. Como citado anteriormente, existem dois tipos distintos de quedas, sendo um mais rápido que o outro devido à presença ou não da perna, que tenta interromper a

queda. Analisando a queda que não foi detectada, chegou-se a conclusão que durante ela, por se tratar a primeira queda teste de costas realizada pela usuária, a perna acabou por amortecer a queda, transformando a mesma, em uma queda lenta.

Com relação ao tempo de envio de SMS os resultados foram muito satisfatórios, na medida em que em todos os testes levou-se menos de um minuto para que a mensagem chegasse em seu destino. Valor muito abaixo do especificado que seria de 5 minutos. Porém, com relação a precisão da localização, os resultados foram pouco satisfatórios, visto que a precisão da localização foi em torno de 800 a 1000 metros, valor muito acima do especificado, de 100 metros. Esse seria um outro ponto a se melhorar.

Finalmente, com relação ao tempo de duração da bateria, é fato que ela é muito menor da estabelecida de uma semana. No entanto, esse é um problema mais simples de ser resolvido, uma vez que existem no mercado baterias com uma maior capacidade de armazenamento de energia (até de 6000 mAh, 7,5 vezes maior que a comprada para o projeto), mas também é possível de montar circuitos elétricos com o foco muito mais voltado às funcionalidades no projeto, deixando-os mais simples e com um consumo de energia muito inferior ao atual (SCHWARTZ, 2013)

5 CONCLUSÃO

A partir das análises dos resultados e com base na descrição completa do projeto, pode-se afirmar que o resultado final foi satisfatório. Isso porque o projeto conseguiu atender o escopo definido inicialmente. No entanto, apesar de o projeto ter conseguido cumprir a todas as funcionalidades principais, alguns dos requisitos estabelecidos não foram atendidos.

Retomando o escopo inicial, o projeto final conseguiu atender a todos os itens definidos. Ou seja, o projeto final foi capaz de detectar quedas, de enviar alertas em casos em que se detectavam quedas e também de informar a localização do local do acidente. Durante toda a execução do projeto, tomou-se o cuidado de garantir ao menos as funcionalidades principais.

Com relação aos requisitos não atingidos, não foi possível ter uma localização com uma precisão dentro do limite do aceitável estabelecido nos requisitos e também não foi possível fazer o dispositivo detectar todas as ocorrências de queda. O primeiro caso deveu-se em parte pelo fato da versão gratuita da API do Google não ser capaz utilizar os dados de mais de uma antena em seu programa e em parte pela dificuldade de se obter um mapeamento das antenas de sinal de celular da cidade de São Paulo. Já o segundo caso deve-se muito pelo perfil de aceleração da queda frontal, cuja aceleração é menor que das outras quedas. Para contornar esse problema, seria necessário uma análise de mais quedas e um aprimoramento do algoritmo de detecção.

5.1 PROJETO FINAL

O projeto final conseguiu cobrir todo o escopo estabelecido, mas para isso, foi preciso fazer algumas alterações ao longo do projeto. Inicialmente o projeto tinha objetivos mais amplos, tanto na parte de detecção de queda quanto na parte de localização. No entanto, devido ao alto grau de complexidade de alguns desses objetivos frente ao prazo de um ano, eles foram descartados do projeto final.

Mesmo assim, é válido notar que o foco principal do projeto que é a detecção de queda, o envio de alerta e a localização do acidente foi mantido e atendido.

Assim, o projeto final consiste em um dispositivo portátil que é capaz de detectar se o usuário que o está usando sofreu uma queda a partir da medição das acelerações que o mesmo está submetido e, como resultado da detecção, o dispositivo envia um SMS para números cadastrados em seu sistema indicando os dados das antenas que estão em sua proximidade para que o celular receptor do SMS possa fazer a localização do acidente a partir um aplicativo que foi implementado juntamente com o projeto.

5.2 TRABALHOS FUTUROS

Como próximos passos, pode-se citar: (i) a implementação da localização em ambientes internos através do uso dos sinais do Wi-Fi junto com um mapeamento do local em que se deseja fazer a localização, sendo assim possível localizar até o cômodo em que ocorreu a queda, (ii) o aprimoramento da localização a partir do cruzamento entre os dados das antenas e dados de pontos de Wi-Fi próximos do dispositivo e a utilização do serviço pago do Google para fazer o uso pleno da API que possui essa funcionalidade, (iii) fazer um banco de dados centralizado que armazene todos os locais das quedas e a data para que seja possível construir um “mapa de queda”, indicando locais aonde é necessário tomar medidas para se evitar as quedas, (iv) adicionar no dispositivo sensores adicionais capazes de melhorar o monitoramento do estado de saúde do acidentado, tais como sensores de batimento cardíaco, de pressão arterial, de temperatura corporal, de sudorese, entre outros, (v) envio de alerta para casos em que o dispositivo não está em uso, entre outras possíveis melhorias.

6 BIBLIOGRAFIA

ADAFRUIT. All about batteries, 2013. Disponível em: <<https://learn.adafruit.com/all-about-batteries/>>. Acesso em: 30 maio 2014.

ALECRIM, E. Tecnologia Bluetooth: o que é e como funciona? **InfoWester**, 2013. Disponível em: <<http://www.infowester.com/bluetooth.php>>. Acesso em: 01 jun. 2014.

ALENCAR, B. D. Pesquisa tenta entender a complicada relação entre idosos e tecnologia. **USP**, 2013. Disponível em: <<http://www5.usp.br/35129/pesquisa-tenta-entender-a-complicada-relacao-entre-idosos-e-tecnologia/>>. Acesso em: 01 jun. 2014.

ALLAN, A. **Make**, 2013. Disponível em: <<http://makezine.com/2013/04/15/arduino-uno-vs-beaglebone-vs-raspberry-pi/>>. Acesso em: 10 Outubro 2014.

ANATEL, 2011. Disponível em: <<http://legislacao.anatel.gov.br/resolucoes/2011/68-resolucao-575>>. Acesso em: 01 jun. 2014.

ANDROID EXAMPLE, 2013. Disponível em: <http://androidexample.com/Incoming_SMS_Broadcast_Receiver_-_Android_Example/index.php?view=article_discription&aid=62&aaid=87>. Acesso em: 15 Outubro 2014.

ARDUINO, 2014. Disponível em: <<http://arduino.cc/>>. Acesso em: 03 jun. 2014.

BAMBURIC, M. **Betanews**, 2014. Disponível em: <<http://betanews.com/2014/02/27/windows-phone-will-still-be-in-android-and-iphones-shadow-four-years-from-now/>>. Acesso em: 30 Outubro 2014.

BEAGLEBOARD FOUNDATION, 2014. Disponível em: <<http://beagleboard.org/>>. Acesso em: 03 jun. 2014.

BRAIN, M.; HARRIS, T. Medindo distâncias. **How Stuff Works**, 2014. Disponível em: <<http://tecnologia.hsw.uol.com.br/receptores-gps.htm>>. Acesso em: 04 Junho 2014.

BRENDLER, C. Como funciona o giroscópio?, 2002. Disponível em: <<http://inima.com.br/mat/idmat013132.htm>>. Acesso em: 20 maio 2014.

BYTELIGHT, 2014. Disponível em: <<http://www.bytelight.com/>>. Acesso em: 20 maio 2014.

CAMPO, E.; GRANGEREAU, E. **Wireless fall sensor with GPS location for monitoring the elderly**. Engineering in Medicine and Biology Society, 2008. EMBS 2008. 30th Annual International Conference of the IEEE. Vancouver, BC: IEEE Conference Publications. 2008. p. 498-501.

CARBONNEAU, E.; SMEESTERS, C. Gait & Posture. **Gait & Posture**, v. 39, p. 365-371, 2014.

CASTLE, A. How-to. **Tested**, 2013. Disponível em: <<http://www.tested.com/tech/robots/456466-know-your-arduino-guide-most-common-boards/>>. Acesso em: 02 Novembro 2014.

CIRIACO, D. Li-Ion e Li-Po: diferenças entre as baterias usadas no iPhone e no Galaxy. **TecMundo**, 2013. Disponível em: <<http://www.tecmundo.com.br/bateria/37981-li-ion-e-li-po-diferencas-entre-as-baterias-usadas-no-iphone-e-no-galaxy.htm>>. Acesso em: 30 maio 2014.

COOKING HACKS, 2014. Disponível em: <<http://www.cooking-hacks.com/documentation/tutorials/arduino-gprs-gsm-quadband-sim900>>. Acesso em: 11 Novembro 2014.

DINH, A.; WANG, T. **Bandage-size non-ECG heart rate monitor using ZigBee wireless link**. Bioinformatics and Biomedical Technology (ICBBT), 2010 International Conference on. Chengdu: IEEE Conference Publications. 2010. p. 160-163.

FRIZZO, M. A. USSD. **WirelessBr**, 2014. Disponível em: <<http://www.wirelessbrasil.org/wirelessbr/colaboradores/mfrizzo/ussd.html>>. Acesso em: 01 jun. 2014.

G1. **G1**, 2014. Disponível em: <<http://g1.globo.com/tecnologia/noticia/2014/07/smartphones-ja-sao-tres-cada-quatro-celulares-vendidos-no-brasil.html>>. Acesso em: 10 Outubro 2014.

GLOBO CIÊNCIA, 2011. Disponível em: <<http://redeglobo.globo.com/globociencia/noticia/2011/10/da-bussola-ao-gps-entenda-como-funciona-navegacao-na-era-digital.html>>. Acesso em: 03 Novembro 2014.

GOOGLE DEVELOPERS. **Google Developers**, 2014. Disponível em: <<https://developers.google.com>>. Acesso em: 20 maio 2014.

GOOGLE INC. Nosso Planeta Mobile: Brasil, 2012. Disponível em: <<http://www.slideshare.net/ivilabessa/relatrio-google-sobre-uso-de-smartphone-no-brasil-maio2012>>. Acesso em: 01 jun. 2014.

GOVERNMENT OF WESTERN AUSTRALIA DEPARTMENT OF HEALTH. **Post-Fall Management Guidelines in WA Healthcare Settings**. East Perth. 2013.

GPRS/GSM Quadband Module for Arduino. **Cooking Hacks**, 2014. Disponível em: <<http://www.cooking-hacks.com/documentation/tutorials/arduino-gprs-gsm-quadband-sim900>>. Acesso em: 22 Outubro 2014.

HEATHCOTE, C. 35 ways to find your location - O'Reilly Emerging Technology Conference, 9-12 Fevereiro 2004. Disponível em: <<http://www.slideshare.net/sidneydekoning/35-ways-to-find-your-location>>. Acesso em: 20 maio 2014.

HOBBYTRONICS, 2014. Disponível em: <<http://www.hobbytronics.co.uk/arduino-serial-buffer-size>>. Acesso em: 30 Setembro 2014.

HOSHINO, M.; WAI, R., 2014.

HOSPITAL ALBERT EINSTEIN. Como prevenir a queda de idosos?, 2012. Disponível em: <<http://www.einstein.br/einstein-saude/bem-estar-e-qualidade-de-vida/Paginas/como-prevenir-a-queda-de-idosos.aspx>>. Acesso em: 03 jun. 2014.

IBGE. Acesso à Internet e Posse de Telefone Móvel Celular para Uso Pessoal, 2013. Disponível em: <<http://www.ibge.gov.br/home/presidencia/noticias/imprensa/ppts/00000012962305122013234016242127.pdf>>. Acesso em: 01 jun. 2014.

INVENSENSE INC. **MPU-6000 and MPU-6050 Product Specification Revision 1.0**. [S.l.]. 2010.

JUNG, S.; CHANG, A.; GERLA, M. **Temporary Interconnection of ZigBee Personal Area Network (PAN)**. Mobile and Ubiquitous Systems: Networking & Services, 2007. MobiQuitous 2007. Fourth Annual International Conference on. Philadelphia, PA: IEEE Conference Publications. 2007. p. 1-5.

LEARN NOT TO FALL. Fear of Falling. **Learn Not To Fall**, 2012. Disponível em: <<http://www.learnnottofall.com/content/fall-facts/fear-of-falling.jsp>>. Acesso em: 27 Novembro 2014.

MALHI, K. et al. A Zigbee-Based Wearable Physiological Parameters Monitoring System. **Sensors Journal, IEEE**, v. 12, n. 3, p. 423-430, March 2012. ISSN 1530-437X.

MARGOLIS, M. **Arduino Cookbook**. 2nd edition . ed. [S.l.]: O'Reilly, 2011.

MAYNARD, C. Adopting Agile in Hardware and Software Combined Environments. **cPrime**, 01 ago. 2012. Disponível em: <<https://www.cprime.com/2012/08/challenges-of-adopting-agile-in-combined-hardware-and-software-environments/>>. Acesso em: 07 abr. 2014.

NARASIMHAN, R. **Skin-contact sensor for automatic fall detection**. Engineering in Medicine and Biology Society (EMBC), 2012 Annual International Conference of the IEEE. San Diego, CA: IEEE Conference Publications. 2012. p. 4038-4041.

NEWHARDWARE. **Mercado Livre**, 2014. Disponível em: <http://produto.mercadolivre.com.br/MLB-599114062-arduino-shield-modulo-gsm-gprs-sim900-antena-_JM>. Acesso em: Maio 2014.

OLHAR DIGITAL. Um em cada quatro domicílios possui wifi. **Olha Digital**, 2012. Disponível em: <<http://olhardigital.uol.com.br/noticia/um-em-cada-quatro-domicilios-possui-wi-fi/25373>>. Acesso em: 25 maio 2014.

OVERCASH, J. A.; JR., H. R. R.; SCHAICK, J. V. An Analysis of Falls Experienced by Older Adult Patients Diagnosed With Cancer. **Oncology Nursing Forum**, v. 37, n. 5, p. 573-580, Setembro 2010.

PARK, C.; KIM, J. **A Location and Emergency Monitoring System for Elder Care Using ZigBee**. Mobile Ad-hoc and Sensor Networks (MSN), 2011 Seventh International Conference on. Beijing: IEEE Conference Publications. 2011. p. 367-369.

POPESCU, M. et al. **An acoustic fall detector system that uses sound height information to reduce the false alarm rate**. Engineering in Medicine and Biology Society, 2008. EMBS 2008. 30th Annual International Conference of the IEEE. Vancouver, BC: IEEE Conference Publications. 2008. p. 4628-4631.

RAMBLINGS, Y. Understanding Android: Three Ways to Find Your Location, 2012. Disponível em: <<http://www.idyllramblings.com/2012/03/understanding-android-three-ways-to-find-your-location.html>>. Acesso em: 15 maio 2014.

RASPBERRY PI FOUNDATION, 2014. Disponível em: <<http://www.raspberrypi.org/>>. Acesso em: 03 jun. 2014.

SANDOVAL, A. Panorama do mercado móvel no Brasil e na América Latina para 2014. **Brasil Link**, 2014. Disponível em: <<http://brasillink.usmediaconsulting.com/2014/02/panorama-do-mercado-movel-no-brasil-e-na-america-latina-para-2014/>>. Acesso em: 01 jun. 2014.

SCHUTZBERG, A. Ten Things You Need to Know About Indoor Positioning. **Directions Magazine**, 2013. Disponível em: <<http://www.directionsmag.com/articles/10-things-you-need-to-know-about-indoor-positioning/324602>>. Acesso em: 22 maio 2014.

SCHWARTZ, M., 2013. Disponível em: <<http://www.openhomeautomation.net/arduino-battery/>>. Acesso em: 31 Outubro 2014.

SIM, S. Y. et al. **Fall detection algorithm for the elderly using acceleration sensors on the shoes**. Engineering in Medicine and Biology Society, EMBC, 2011 Annual International Conference of the IEEE. Boston, MA: IEEE Conference Publications. 2011. p. 4935-4938.

SIM900 AT Command Manual. **Webtronico**, 2010. Disponível em: <http://www.webtronico.com/documentos/SIM900_AT%20Command%20Manual_V1.03.pdf>. Acesso em: 22 Outubro 2014.

SIXSMITH, A.; JOHNSON, N. A smart sensor to detect the falls of the elderly. **Pervasive Computing, IEEE**, v. 3, n. 2, p. 42-47, 2004.

SMEESTERS, C.; HAYES, W. C.; MCMAHON, T. A. The threshold trip duration for which recovery is no longer possible is. **Journal of Biomechanics**, v. 34, p. 589–595, 2001.

SOAZ, C.; LEDERER, C.; DAUMER, M. **A new method to estimate the real upper limit of the false alarm rate in a 3 accelerometry-based fall detector for the elderly**. Engineering in Medicine and Biology Society (EMBC), 2012 Annual International Conference of the IEEE. San Diego, CA: IEEE Conference Publications. 2012. p. 244-247.

SSCI, 2014. Disponível em: <<http://www.ssci.co.th/modules.php?module=articles&do=article&artname=Mobile-Payment>>. Acesso em: 03 Novembro 2014.

STACKOVERFLOW, 2013. Disponível em: <<http://stackoverflow.com/questions/14579426/google-android-maps-api-v2-show-marker-tittle-always>>. Acesso em: 20 Outubro 2014.

SUH, K.-Y.; JEON, N. L.; PANG, C. **Wearable skin sensor using programmable interlocking of nanofibers**. Nano/Micro Engineered and Molecular Systems (NEMS), 2013 8th IEEE International Conference on. Suzhou: IEEE Conference Publications. 2013. p. 132-135.

TELECO, 2014. Disponível em: <<http://www.teleco.com.br/cobertura.asp>>. Acesso em: 22 maio 2014.

THAMMASAT, E.; CHAICHARN, J. **A simply fall-detection algorithm using accelerometers on a smartphone**. Biomedical Engineering International

Conference (BMEiCON), 2012. Ubon Ratchathani: IEEE Conference Publications. 2012. p. 1-4.

TSAGKLIS, I., 2013. Disponível em: <<http://www.javacodegeeks.com/2011/02/android-google-maps-tutorial.html>>. Acesso em: 18 Outubro 2014.

TYLERBRINKS. Learn How to Find GPS Location on Any SmartPhone, and Then Make it Relevant. **CodeProject**, 2014. Disponível em: <<http://www.codeproject.com/Articles/31965/Learn-How-to-Find-GPS-Location-on-Any-SmartPhone-a>>. Acesso em: 19 maio 2014.

TYTELL, E. The fall guy. **New Scientist**, Londres, v. 184, n. 2479/2480, p. 48-49, Dezembro 2004.

VAUGHAN-NICHOLS, S. J. How Google - and everyone else - gets Wi-Fi location data. **ZDNet**, 2011. Disponível em: <<http://www.zdnet.com/blog/networking/how-google-and-everyone-else-gets-wi-fi-location-data/1664>>. Acesso em: 18 maio 2014.

VOGEL, L., 2014. Disponível em: <<http://www.vogella.com/tutorials/AndroidSQLite/article.html>>. Acesso em: 22 Outubro 2014.

WEBTRONICO. **Webtronico**, 2014. Disponível em: <<http://www.webtronico.com/gsm-gprs-shield-sim900-arduino.html>>. Acesso em: 5 Novembro 2014.

WEN-TSAI SUNG, C.-C. H. Intelligent environment monitoring system based on Innovative Integration Technology via Programmable System On Chip platform and ZigBee network. **IET Communications**, 17 Julho 2013. 1789-1801.

WEXLER, J. All about Wi-Fi location tracking. **TechWorld**, 2006. Disponível em: <<http://features.techworld.com/mobile-wireless/2374/all-about-wi-fi-location-tracking/>>. Acesso em: 20 maio 2014.

WIRELESSBR. Métodos de Localização Baseados no Cell-ID. **WirelessBR**, 2014. Disponível em: <http://www.wirelessbrasil.org/wirelessbr/colaboradores/thienne_johnson/cell_id/cell_id.html>. Acesso em: 04 jun. 2014.

WORLD HEALTH ORGANIZATION. **WHO Global report on falls Prevention in older Age**. Genebra. 2007.

YI, W.; TONG, W.; PAI, L. **Smart home system based on ZigBee and ARM.** Electronic Measurement & Instruments (ICEMI), 2013 IEEE 11th International Conference on. Harbin: IEEE Conference Publications. 2013. p. 754-759.

ZHENG, J.; ZHANG, G.; WU, T. **Design of Automatic Fall Detector for Elderly Based on Triaxial Accelerometer.** Bioinformatics and Biomedical Engineering , 2009. ICBBE 2009. 3rd International Conference on. Beijing: IEEE Conference Publications. 2009. p. 1-4.

APÊNDICE A PROGRAMA PARA TESTE DO MÓDULO GSM

Desenvolveu-se um programa para testar as funções criadas para o uso do módulo GSM. Cada função é chamada através do envio do código correspondente (ex: 1,2,3...). As funções implementadas e seus códigos são:

1. Ligar
2. Desligar
3. Comando AT
4. Comando AT+COPS?
5. Comando AT+CENG=1,1
6. Comando AT+CENG? -> pegar parâmetros
7. Enviar mensagem
8. Cadastra celular

Para o uso dessas funções foi necessária a criação de algumas rotinas que foram posteriormente utilizadas no software do dispositivo. Essas rotinas foram, por exemplo, a de ler a resposta do módulo GSM (readSIM), a de ligar automaticamente o módulo, a de enviar SMS e a de receber SMS.

1. Ligar

Esta função alterna o estado do módulo GSM entre ligado e desligado (*standby*) através do envio de sinais específicos no pino 9.

2. Desligar

Esta função realiza o desligamento do módulo GSM através do comando AT:

AT+CPOWD=1

No entanto, optou-se pela utilização da função que alterna o estado do módulo para o dispositivo final, de modo a simplificar o programa e minimizar o uso de memória.

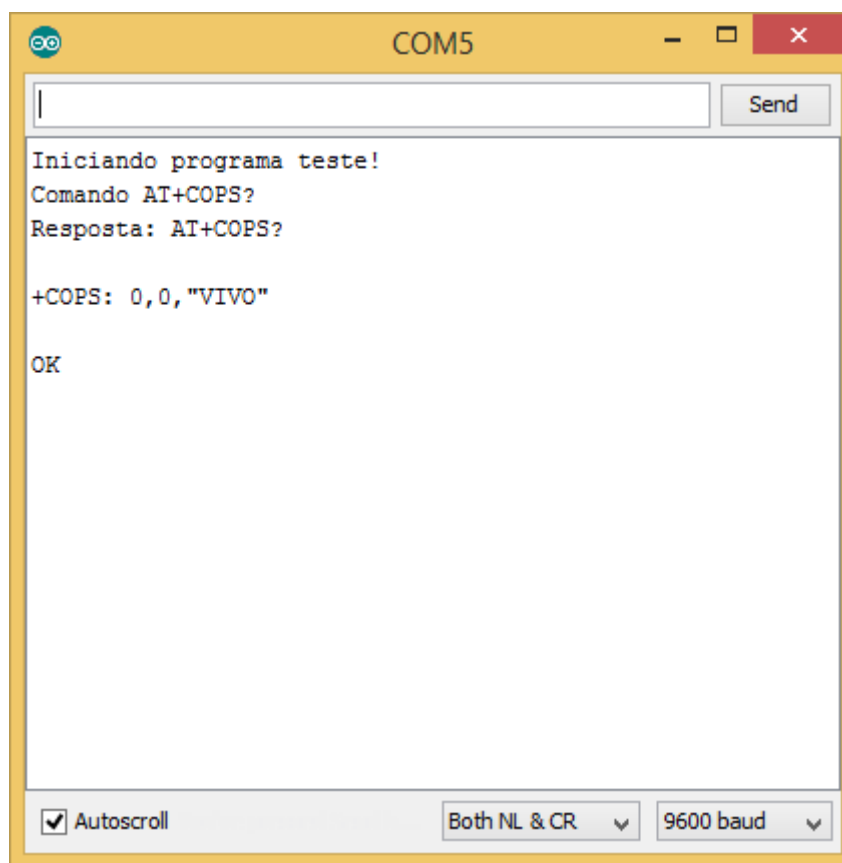
3. Comando AT

Este comando indica somente se a comunicação com o módulo GSM está funcionando.

4. Comando AT+COPS?

Este comando indica se o módulo GSM está conectado à uma rede de celular. Assim, este comando foi utilizado na verificação de rede antes de se realizar um envio de SMS, por exemplo. Na Figura 39, por exemplo, a resposta ao comando indica que o módulo GSM está conectado à rede VIVO.

Figura 39 - Comando de verificação de rede



Fonte: HOSHINO e WAI (2014)

5. Comando AT+CENG=1,1

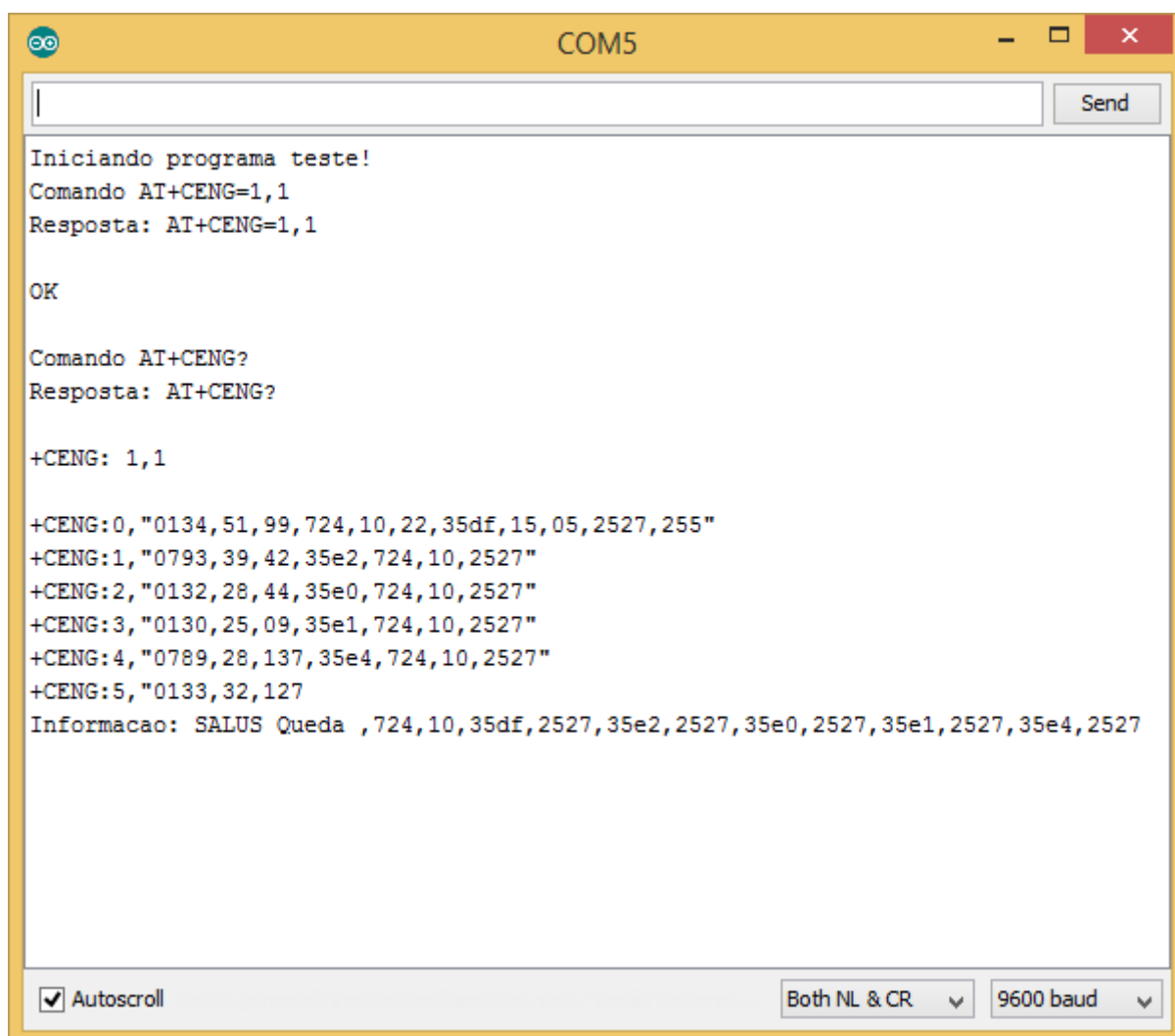
Este comando ativa o modo engenheiro do módulo GSM, necessário para a aquisição dos dados das antenas. O primeiro parâmetro é responsável pela ativação do modo engenheiro, já o segundo parâmetro indica que se deseja obter os CellIDs das antenas vizinhas. Caso este segundo parâmetro não seja

especificado, apenas o CellID da antena à qual o modelo está conectado será mostrado quando for enviado o comando “AT+CENG?”.

6. Comando AT+CENG? -> pegar parâmetros

Este comando é o responsável pela obtenção dos dados das antenas. Para este projeto é necessário apenas o MCC, MNC, LAC e CellID, no entanto, dados como o Timing Advance, Signal Strength, a frequência utilizada, entre outros, também podem ser obtidos.

Figura 40 - Habilitação do GSM para a visualização e obtenção dos dados das antenas

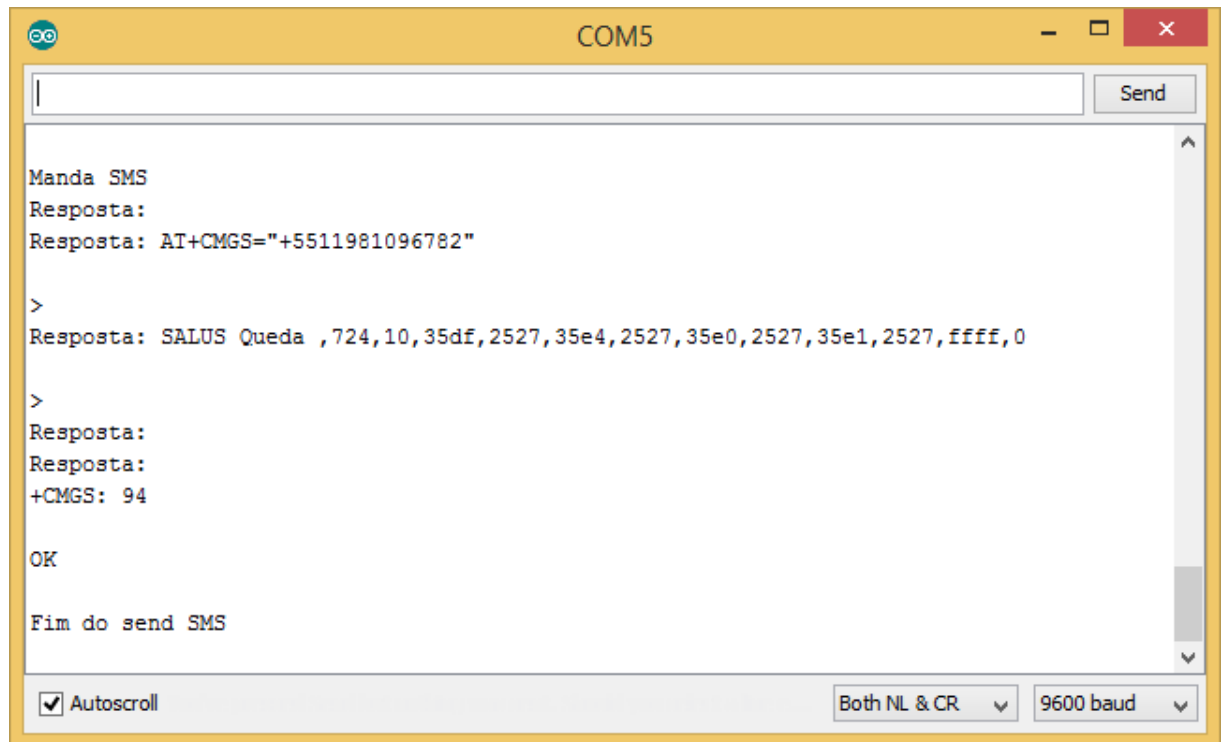


Fonte: HOSHINO e WAI (2014)

7. Enviar mensagem

Esta função é responsável pelo envio de um SMS. Ela funciona tal como explicado em 4.2.1.5.

Figura 41 - Comando de envio de SMS

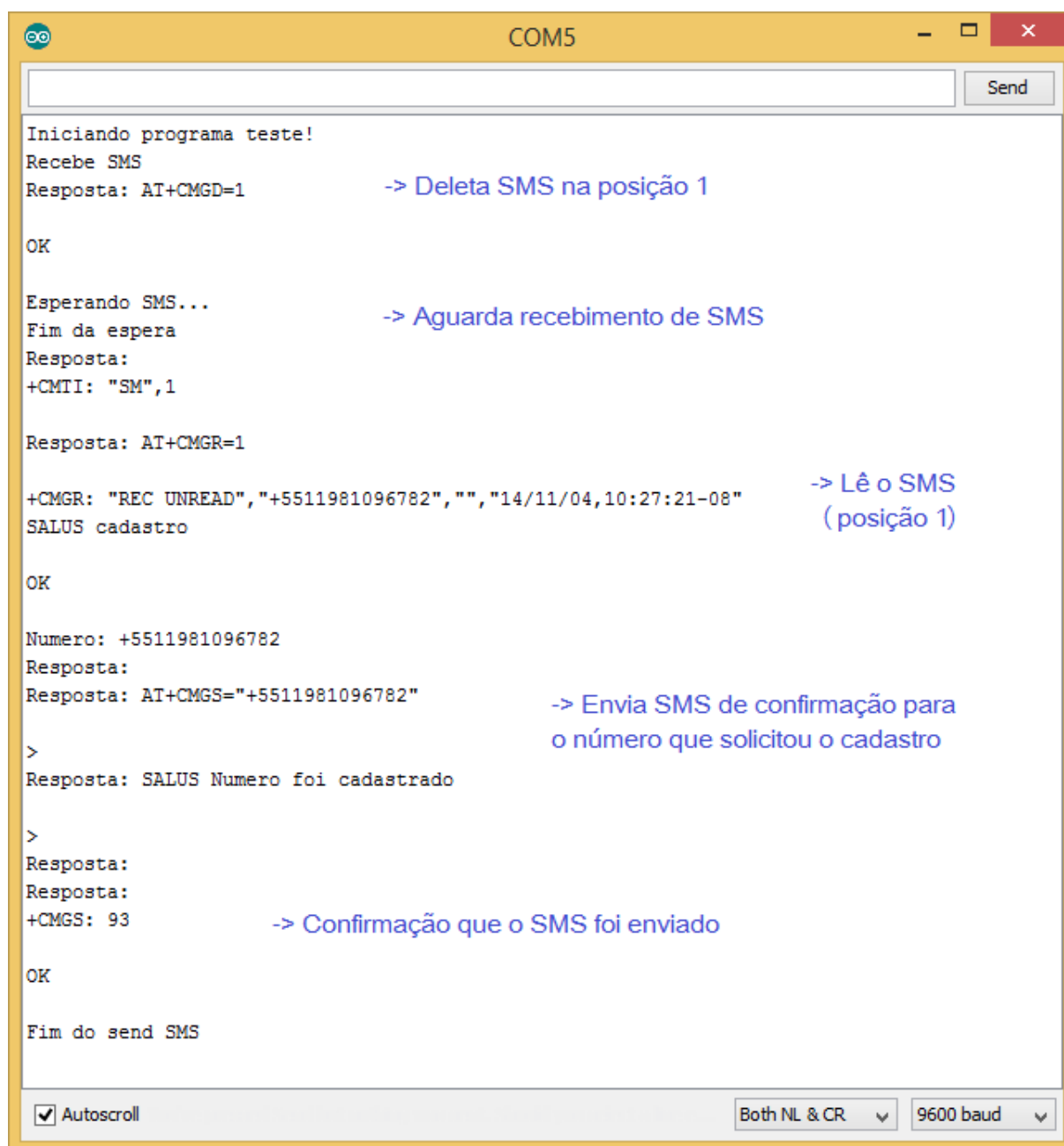


Fonte: HOSHINO e WAI (2014)

8. Cadastra celular

Esta função é responsável pelo recebimento de um SMS. Ela funciona tal como explicado em Figura 42

Figura 42 - Procedimento de cadastro de celular



Fonte: HOSHINO e WAI (2014)

APÊNDICE B MANUAL DE INSTRUÇÕES

Este manual explica o uso do dispositivo e do aplicativo envolvidos neste projeto.

1. Antes de começar

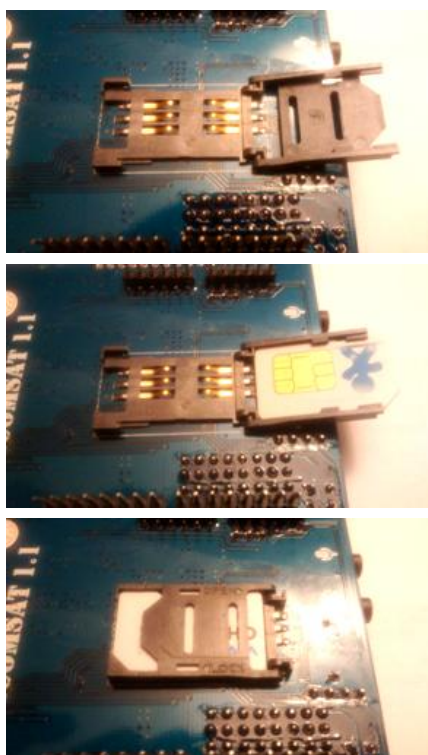
Antes de iniciar o uso do dispositivo é necessário prepará-lo. A preparação inclui a obtenção de um chip de celular, a instalação do aplicativo e carregar a bateria do dispositivo.

a) Chip de celular

É necessário obter um chip de celular válido para uso no dispositivo. Este chip pode ser pré-pago ou pós-pago, já que a única exigência é que possa receber e enviar SMS. Caso o chip seja pré-pago, deve-se verificar a suficiência de créditos para o uso do dispositivo. O chip deve ser inserido no slot que está na parte inferior do módulo GSM.

A inserção do chip deve ser feita conforme as figuras a seguir.

Figura 43 - Sequência para inserção do chip



Fonte: HOSHINO e WAI (2014)

b) Instalação do aplicativo (Android)

Para instalar o aplicativo, deve-se fazer o *download* da .apk, no smartphone, através do link:

<https://sites.google.com/site/tcc2014salus/aplicativo>

Após o *download*, seguir o tutorial disponível no mesmo link.

c) Carregar bateria

Antes do uso do dispositivo deve-se verificar se a bateria do mesmo encontra-se carregada. Para isso, pode-se tentar ligar o dispositivo. Se algum LED do Arduino e do módulo GSM se acender, o dispositivo então possui carga. Caso o LED encontrado na parte superior do dispositivo não se acenda a bateria possui um nível aceitável de carga e não precisa ser imediatamente carregada.

Caso nenhum LED do Arduino ou do módulo GSM se acenda, deve-se carregar a bateria. Também é necessário carregar a bateria caso o LED da parte superior permaneça aceso, o que indica bateria baixa.

Para carregar a bateria, conecte-a ao carregador apropriado para ela.

2. Cadastro de telefone

Para iniciar o uso do dispositivo deve-se cadastrar o número de celular que receberá as mensagens de alerta. O procedimento para cadastro é:

- i. Apertar o botão na parte superior do dispositivo
- ii. Aguardar até o LED começar a piscar
- iii. Enviar um SMS para o número do chip do dispositivo com a mensagem:

SALUS Cadastro

- iv. Quando o LED parar de piscar é porque o dispositivo recebeu a mensagem de cadastro
- v. Aguardar a seguinte mensagem para confirmar o cadastro:

SALUS Numero foi cadastrado

Após o cadastro de um número de telefone o dispositivo pode ser usado.

3. Alerta de queda

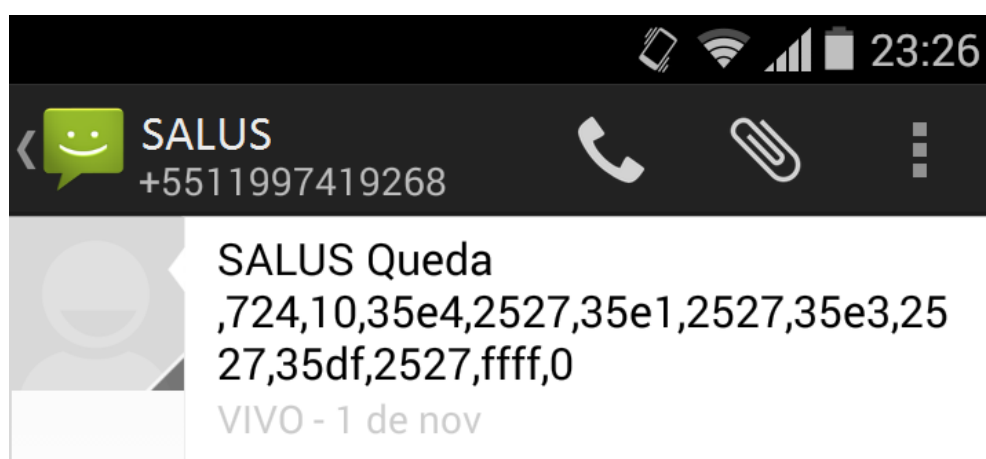
Para que o dispositivo consiga detectar quedas é necessário que ele esteja ligado e que esteja sendo utilizado pelo usuário, preferencialmente na região peitoral.

Ao detectar uma queda, o dispositivo envia a seguinte mensagem ao celular cadastrado:

SALUS Queda ,<série de dados das antenas>

Um exemplo real dessa mensagem é o seguinte:

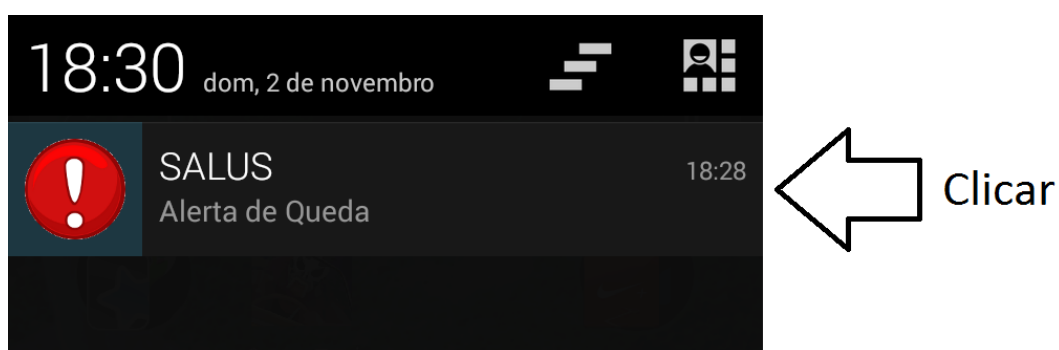
Figura 44 - Exemplo de mensagem de queda



Fonte: HOSHINO e WAI (2014)

Logo após que a mensagem é recebida pelo celular, uma notificação será mostrada Área de Notificações. Para visualizar o local da queda basta clicar sobre a notificação:

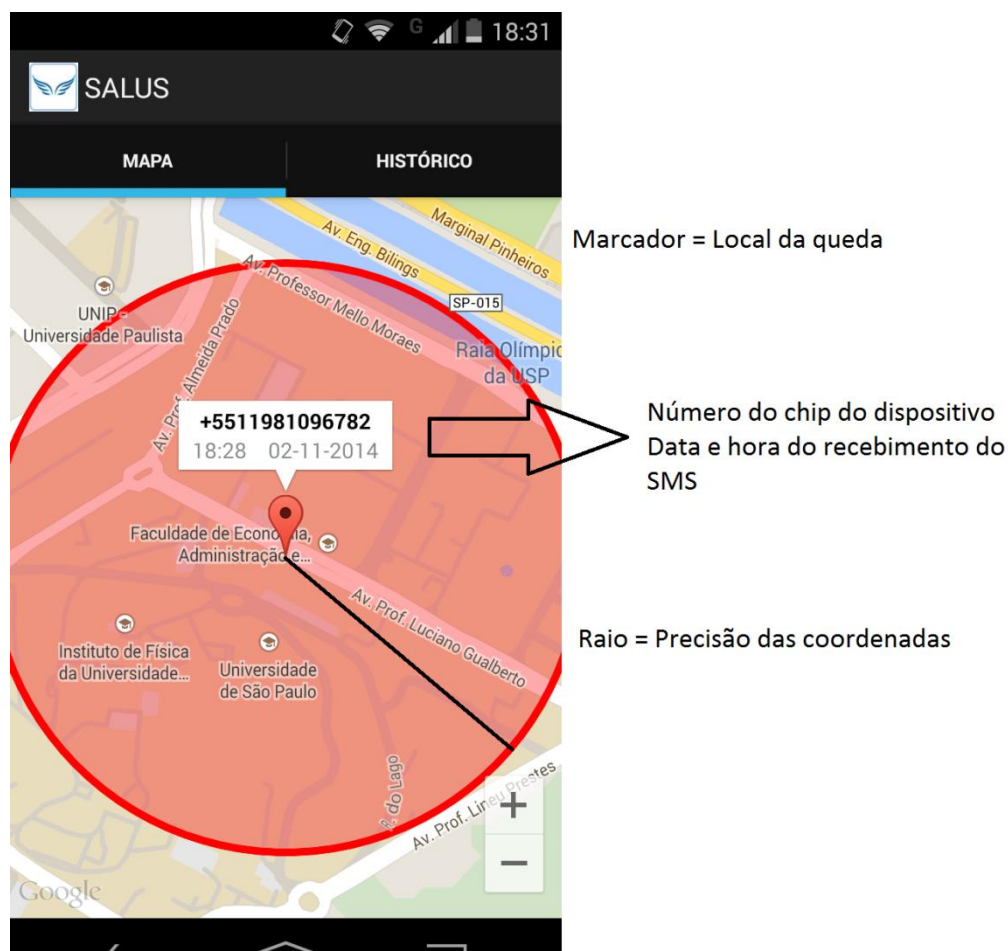
Figura 45 – Procedimento na notificação de queda



Fonte: HOSHINO e WAI (2014)

Após o clique, o aplicativo será iniciado. Assim que o aplicativo é aberto este busca conectar-se à internet para interpretar os dados do SMS. Por isso, o celular deve estar conectado a uma rede 3G/4G ou a uma rede Wi-Fi.

Figura 46 - Visualização do local da queda



Fonte: HOSHINO e WAI (2014)

4. Alerta de bateria baixa

Caso o dispositivo esteja com um nível baixo de bateria ele realizará o envio da seguinte mensagem:

SALUS Bateria baixa

Além disso, o LED superior se manterá aceso, indicando o baixo nível da bateria, até que esta seja carregada.

5. Recomendações

Segue abaixo uma lista de recomendações quanto ao uso do dispositivo:

- Procure usar o dispositivo no peito, ou próximo ao centro de massa do usuário para uma melhor precisão na detecção de quedas.
- Caso haja a detecção de queda procure contatar o idoso, caso não seja possível, providencie um pronto atendimento para certificar-se do estado físico dele.
- Caso o chip utilizado no dispositivo seja pré-pago, mantenha-o sempre carregado com créditos.
- Mantenha o dispositivo sempre carregado.

