

**UNIVERSIDADE REGIONAL DE BLUMENAU**  
**CENTRO DE CIÊNCIAS EXATAS E NATURAIS**  
**CURSO DE CIÊNCIA DA COMPUTAÇÃO – BACHARELADO**

**USO DO ACELERÔMETRO PARA DETECÇÃO DE QUEDA  
DE IDOSOS**

**JEAN CARLO GASPAR FILHO**

**BLUMENAU**  
**2017**

**JEAN CARLO GASPAR FILHO**

**USO DO ACELERÔMETRO PARA DETECÇÃO DE QUEDA  
DE IDOSO**

Trabalho de Conclusão de Curso apresentado ao curso de graduação em Ciência da Computação do Centro de Ciências Exatas e Naturais da Universidade Regional de Blumenau como requisito parcial para a obtenção do grau de Bacharel em Ciência da Computação.

Prof. Mauro Mattos, Doutor - Orientador

**BLUMENAU  
2017**

# **USO DO ACELERÔMETRO PARA DETECÇÃO DE QUEDA DE IDOSO**

Por

**JEAN CARLO GASPAR FILHO**

Trabalho de Conclusão de Curso aprovado  
para obtenção dos créditos na disciplina de  
Trabalho de Conclusão de Curso II pela banca  
examinadora formada por:

Presidente: \_\_\_\_\_  
Prof. Mauro Marcelo Mattos, Doutor – Orientador, FURB

Membro: \_\_\_\_\_  
Prof. Miguel Alexandre Wisintainer, Mestre – FURB

Membro: \_\_\_\_\_  
Profa. Gabriele Jennrich Bambineti, Especialista – FURB

Blumenau, 12 de dezembro de 2017

Dedico este trabalho a todos aqueles que amo:  
minha família e amigos.

## **AGRADECIMENTOS**

À minha família que sempre esteve ao meu lado, independente das circunstâncias.

À minha amiga Brenda, a qual me apoiou para me manter firme e motivado com o objetivo de concluir este trabalho.

Aos meus caros amigos que amo, pois pude contar com eles sempre que precisei.

Ao meu orientador Mauro por todo o conhecimento compartilhado e todo esforço por este trabalho.

Ao professor Miguel por todo o apoio oferecido aos problemas de eletrônica que tive.

Success is not final, failure is not fatal: it is the courage to continue that counts.

Winston Churchill

## RESUMO

Este trabalho apresenta o desenvolvimento de uma solução para monitoramento da queda de idosos, a fim de notificar alguém para que o idoso seja socorrido. Para isto foi usado um microcontrolador Espruino Pico com bateria, conexão wifi e um acelerômetro, que obtém e transfere os dados de posicionamento para um servidor via TCP (Transfer Control Protocol). O servidor é o responsável por monitorar através dos dados do acelerômetro, identificar e notificar uma queda. Devido a ruídos do acelerômetro é utilizado um filtro de passa baixa evitando falsos positivos, e em seguida é aplicado um cálculo de norma de vetor. Os impactos da queda geram picos nos valores, os quais são usados para definir a queda, onde caso o este pico exceda um valor limite, este será considerada uma queda. Apesar de os testes de queda não terem sido realizados com o equipamento em uma pessoa, foram conduzidos testes para validar falsos positivos. De forma geral o trabalho atendeu os objetivos propostos, detectando quedas e sabendo diferenciar atividades diárias como caminhar, subir e descer escadas, entretanto, ainda assim, existem situações de falso positivo como choques do equipamento com algum obstáculo não originadas de uma queda do usuário.

Palavras-chave: Acelerômetro. Espruino. Idoso. Detecção de queda.

## **ABSTRACT**

This monograph presents the development of a solution to watch the fall of elderly, in order to notify someone to help them. For that it was used a microcontroller Espruino Pico with a battery, wifi connection and an accelerometer, to get and send the data of positioning to a server trough TCP (Transfer Control Protocol). The server is the responsible to monitor trough the data from the accelerometer, identify and notify a fall. Due to noise from accelerometer its used a low pass filter to avoid false positive, then it's taken the euclidean length. The impacts of the falls generate peaks in the values, which are used to determine what is a fall, where if a value threshold is triggered, then it'll be considered a fall. Despite the fall tests not being made with the device on a person, tests for false positive were made. In general the work meet the objectives proposed, detecting falls and differentiating daily activities like walking, climb or down stairs, yet still it has situations of false positive like collisions of the device non originated from a fall of the user.

**Key-words:** Accelerometer. Espruino. Elderly. Fall Dectection.



## LISTA DE FIGURAS

Figura 1 - Espruino original.....	18
Figura 2 – Espruino Pico, versão sem pinos.....	18
Figura 3 – IDE Web Espruino .....	20
Figura 4 – Esquema elétrico da Espruino Pico.....	21
Figura 5 – ESP-01 ESP8266.....	22
Figura 6 – Fórmula de norma do vetor .....	23
Figura 7 – Fórmula de filtro de passa baixa baseado em média móvel.....	23
Figura 8 – Desenho da interface gráfica da aplicação .....	24
Figura 9 - Monitoramento da aceleração do corpo em queda, utilizando cabo elástico com 8g de massa.....	26
Figura 10 - Gráfico de variação da aceleração tangencial ocorrida com o acelerômetro quando acoplado aos pontos de valor (r) igual a 1/3L, 2/3L ou L.....	26
Figura 11 – Resultados obtidos através dos testes.....	28
Figura 12 – Diagrama de caso de uso.....	31
Figura 13 – Dispositivo e os LEDs quando o dispositivo funciona normalmente .....	32
Figura 14 – Dispositivo e os LEDs quando o dispositivo não consegue conectar .....	32
Figura 15 – Diagrama de atividades do funcionamento do dispositivo .....	33
Figura 16 – Diagrama de rede do protótipo.....	34
Figura 17 – Diagrama de atividades da aplicação no servidor .....	34
Figura 18 – Esquema elétrico do dispositivo.....	35
Figura 19 – Dispositivo após solda de todos os componentes .....	36
Figura 20 – Tela da aplicação servidor.....	38
Figura 21 – Código para transformar dados recebidos do <i>stream</i> e aplicar o filtro .....	40
Figura 22 – Código do filtro de passa baixa.....	40
Figura 23 – Exemplo de e-mail gerado pela aplicação servidor.....	42
Figura 24 - Gráfico de queda do dispositivo .....	43
Figura 25 – Gráfico de queda de uma vassoura.....	44
Figura 26 – Vassoura acoplada do dispositivo e de peso extra para testes .....	45
Figura 27 – Gráfico de queda da vassoura na almofada com peso extra.....	46
Figura 28 – Dispositivo preso na altura do peito.....	46
Figura 29 – Gráfico de queda simulada.....	47

Figura 30 – Gráfico de simulação de falsos positivos batendo diretamente no dispositivo .....	47
Figura 31 – Gráfico de dados obtidos durante andar, subir e descer uma escada .....	48

## LISTA DE QUADROS

Quadro 1 – Código exemplo de conexão Espruino à ESP8266, conectar a wi-fi e fazer uma requisição HTTP.....	19
Quadro 2 – Requisitos funcionais.....	30
Quadro 3 – Requisitos não funcionais.....	30
Quadro 4 - Código para conectar ao servidor TCP e leitura de acelerômetro.....	37
Quadro 5 – Código que recebe a conexão TCP e lê os dados .....	39
Quadro 6 – Código do cálculo de norma de vetor e verificação de queda .....	41
Quadro 7 – Código para envio de e-mail via SMTP .....	41
Quadro 8 – Comparativo do trabalho presente e os correlatos.....	49

## **LISTA DE ABREVIATURAS E SIGLAS**

LED – Light Emiting Diode

RF – Requisito funcional

RNF – Requisito não funcional

SMTP – Simple Mail Transfer Protocol

TCP – Tranmission Control Protocol

UC – Use de case

UML – Unified Modeling Language

# SUMÁRIO

<b>1 INTRODUÇÃO.....</b>	<b>13</b>
1.1 OBJETIVOS.....	14
1.2 ESTRUTURA.....	14
<b>2 FUNDAMENTAÇÃO TEÓRICA .....</b>	<b>15</b>
2.1 DIMINUIÇÃO DA CAPACIDADE FUNCIONAL EM IDOSOS .....	15
2.1.1 Fatores associados.....	16
2.2 A PLATAFORMA ESPRUIÑO PICO .....	17
2.3 ALGORITMO PARA DETECÇÃO DA QUEDA .....	22
2.4 TRABALHOS CORRELATOS .....	23
2.4.1 fAlert : Um sistema android para monitoramento de quedas em pessoas com cuidados especiais .....	24
2.4.2 Acelerômetro eletrônico e a placa Arduino para ensino de Física em tempo real.....	25
2.4.3 Evaluation of a threshold-based tri-axial accelerometer fall detection algorithm .....	27
<b>3 DESENVOLVIMENTO DO PROTÓTIPO .....</b>	<b>30</b>
3.1 REQUISITOS.....	30
3.2 ESPECIFICAÇÃO .....	30
3.3 IMPLEMENTAÇÃO .....	35
3.3.1 Construção do dispositivo .....	35
3.3.2 Técnicas e ferramentas utilizadas.....	36
3.4 ANÁLISE DOS RESULTADOS .....	41
<b>4 CONCLUSÕES .....</b>	<b>51</b>
4.1 EXTENSÕES .....	51
<b>REFERÊNCIAS.....</b>	<b>53</b>

## 1 INTRODUÇÃO

O idoso está na fase da sua vida na qual se alcançou muitos objetivos, mas que também houveram perdas, onde a saúde é um dos aspectos mais afetados. (MENDES, 2012 apud MENDES 2000). As pessoas idosas ou com algum tipo de deficiência preferem ficar em casa mesmo quando suas situações de saúde estejam piorando. (LEE; CHUAH; CHIENG, 2013). Isso significa que caso estejam sozinhos, não existe como alguém socorrer o idoso caso necessário fosse após sofrer uma queda. Um dos motivos disto ser um problema segundo Lee, Chuah e Chieng (2013) é que as fraturas são uma das principais consequências para os idosos ao sofrerem uma queda, sendo que mais da metade das quedas ocorrem dentro de suas casas. Corroborando esta informação, dados apontam que 28 a 35% de pessoas acima de 65 anos de idade caem pelo menos uma vez durante o ano no mundo. A proporção aumenta para até 42% quando os idosos têm mais de 70 anos. (SIQUEIRA et al., 2007)

Com isto, os sistemas de assistência hospitalar estão sendo procurados para que esta parte da população possa continuar vivendo dentro de suas casas. Uma parcela destes idosos mora sozinha, aumentando a necessidade do apoio sem intervenção humana. Já que os problemas de envelhecimento e incapacitação estão convergindo, sistemas inteligentes de assistência hospitalar no domicílio se tornou uma área-chave de pesquisa para a computação (LEE; CHUAH; CHIENG, 2013).

O envelhecimento da população aumenta a demanda por cuidados de saúde e o progresso nesta área tem sido ampliado com as novas tecnologias de cuidado. A população brasileira tem incorporado equipamentos de monitoramento à saúde de uso doméstico e estes podem auxiliar o sujeito se este for adequadamente instrumentalizado para o uso. (SANTANA et al., 2014, p.383).

Os equipamentos de monitoramento também tem sido interesse do setor privado. Em agosto de 2015 a empresa Pfizer lançou um desafio para valorizar jovens empresas de tecnologia com o foco em produtos e soluções digitais que facilitassem a prática médica diária. Esta iniciativa nasceu em parceria com a empresa Berrini Ventures, (primeira aceleradora nacional dedicada exclusivamente para o setor de saúde). Na fase final, o Desafio Pfizer selecionou dez startups para participar do evento Hospital Innovation Show (HIS), que aconteceu em São Paulo em setembro de 2015. Foram selecionadas as empresas: Clever Care, Cybergia, Doctor ID, Epistemic, iCare, MinsulinTech, Patient Box, Ravek, Senfio e TNH Digitall Health (SAUDE BUSINESS, 2015).

Visto o aqui exposto, foi construído um protótipo afim de monitorar pessoas idosas, permitindo que um servidor, ao receber os dados de monitoramento através da internet, possa notificar alguém sobre as quedas que o idoso venha a sofrer permitindo uma ação de auxiliar em seu socorro.

## 1.1 OBJETIVOS

O objetivo deste trabalho é construir uma solução envolvendo hardware e software que permita o monitoramento de quedas de usuário.

Os objetivos específicos são:

- a) permitir monitoramento em tempo real do usuário;
- b) transmitir os dados de monitoramento para um servidor de retaguarda;
- c) detectar indicativos de quedas e enviar notificação para usuário qualificado.

## 1.2 ESTRUTURA

A presente monografia encontra-se dividida em quatro capítulos: introdução, fundamentação teórica, desenvolvimento e conclusões. O capítulo 2 mostra a importância para que sejam tomados cuidados com a queda de um idoso, além de também fornecer um embasamento teórico a respeito de como é detectada uma queda e que meios são utilizados para se realizar isto. Consta também no mesmo capítulo uma análise de trabalhos correlatos. Já o capítulo 3 apresenta os requisitos, a especificação da implementação, e o pontos mais importantes no desenvolvimento do protótipo, contendo também uma análise dos resultados obtidos na sua construção e testes e um comparativo com trabalhos correlatos. Por último, o capítulo 4 possui um encerramento apresentando as conclusões sobre o trabalho e possíveis extensões a partir deste.

## 2 FUNDAMENTAÇÃO TEÓRICA

Este capítulo tem como objetivo explorar os principais assuntos necessários para a realização do trabalho. Diante disto, o capítulo foi subdividido em quatro seções, onde a seção 2.1 analisa a importância, as consequências, e os fatores que levam às quedas nos idosos. Já a seção 2.2 apresenta o desenvolvimento com a microcontroladora Espruino Pico e sua definição. A seção 2.3 apresenta como é feita a detecção de quedas e que cálculos podem ser envolvidos ao se realizar esta atividade através de um acelerômetro. Por fim, na seção 2.4 são descritos três trabalhos correlatos.

### 2.1 DIMINUIÇÃO DA CAPACIDADE FUNCIONAL EM IDOSOS

Conforme Gasparotto, Falsorella e Coimbra (2014), as quedas têm sido consideradas como um importante fator de risco para a redução da capacidade funcional em indivíduos idosos e o grande desafio dos profissionais da área da saúde é o desenvolvimento de propostas que visem à manutenção da saúde desta população. Capacidade funcional é um conceito utilizado na área de Gerontologia que diz respeito a um dos parâmetros da qualidade de vida nesta população e é expressa por meio da independência funcional (habilidade para executar tarefas do cotidiano) e autonomia funcional (capacidade para decidir e delegar funções) sem que nas duas opções seja necessário algum pedido de ajuda. (D'ORSI; XAVIER; RAMOS, 2011)

A medida em que a idade avança, o processo de envelhecimento promove modificações no corpo e no caso dos idosos, é comum identificar parâmetros reduzidos da massa muscular o que contribui para a redução da força física. Além disso, há evidências de redução de densidade óssea que contribuem para o enfraquecimento da estrutura esquelética do indivíduo tornando-os mais frágeis. (VRIES et al., 2013)

Apesar de não haver uma definição padronizada, do ponto de vista científico, considera-se uma queda é definida como um evento não intencional que tem como resultado a mudança da posição inicial do indivíduo para um mesmo nível ou nível mais baixo (GASPAROTTO; FALSORELLA; COIMBRA, 2014). Alguns autores consideram também alguns tipos de tropeços e escorregões. (DÁMIAN et al., 2013)

Segundo Gasparotto, Falsorella e Coimbra (2014), estudos envolvendo a população idosa no Brasil aponta que as quedas atingem de 30 a 40% dos casos. Uma amostra verificou que 10,8% dos idosos relataram duas ou mais quedas com prevalência de fratura óssea de 5,2%. Outra análise apontou que 12,1% tiveram fratura como consequência. As projeções apontam para um indicativo de que cerca de 10% da população acima de 75 anos venha a



perder a independência em uma ou mais atividades da vida diária. (D'ORSI; XAVIER; RAMOS, 2011)

Conforme Gasparotto, Falsorella e Coimbra (2014),

A queda, entre os diagnósticos de admissão em hospitais, é um dos motivos que mantêm por mais tempo o indivíduo internado, fato este que gera mais gasto para a saúde pública. Entre alguns países do norte da América e Europa, verificou-se que o tempo de permanência no hospital costuma variar de quatro a 15 dias. (Gasparotto; Falsorella; Coimbra, 2014, p. 203)

Conforme Lopes et al., aproximadamente 29% a 92% dos idosos pesquisados que sofreram quedas têm medo de cair de novo, mostrando a possibilidade de o indivíduo restringir algumas tarefas por conta desta sensação e, por isso, ficarem suscetíveis a novas quedas. (LOPES et al, 2009)

As quedas em idosos ocorrem por vários motivos, e mais da metade das quedas ocorrem dentro de suas próprias casas (FREITAS; SCHEICHER, 2008). Isto é um reflexo da pesquisa que mostra que os idosos e pessoas com necessidades especiais preferem estar em casa mesmo com as condições de saúde não sendo boas (LEE; CHUAH; CHIENG, 2013). Além de quedas por atividades rotineiras, segundo Coutinho e Silva (2002) até mesmo a medicação consumida pelo idoso pode influenciar nas chances de o idoso vir a sofrer uma queda.

Dentre as várias consequências que a queda de um idoso pode causar, a mais recorrente delas são as fraturas, principalmente do fêmur. Existem outras consequências mais graves, como danos que acarretam na morte por consequência direta de uma queda. (RIBEIRO et al., 2006)

Segundo Ribeiro et al. (2006), cerca de 20% dos idosos moram sozinhos. Isto significa que caso não esteja ninguém a cuidados do idoso, caso ele sofra alguma queda com algum tipo de lesão como consequência, o socorro seria dificultado. Isto é um problema, visto que segundo Lee, Chuah e Chieng (2013) as chances de falecimento do idoso podem ser reduzidas se ele for atendido até uma hora após o ocorrido.

### 2.1.1 Fatores associados

Muitos são fatores que podem induzir o idoso a uma queda, e vários estudos vem levantando pontos a respeito disto:

A prevalência de quedas tem sido associada com idade avançada, sedentarismo, autopercepção de saúde como ruim e maior consumo de medicações variadas de uso contínuo. Os estudos sugerem que tal procedimento seja repensado em casos de idosos com queixas desta enfermidade. Entre as quedas recorrentes, verificou-se a predominância maior em mulheres, idosos viúvos, solteiros e desquitados, idosos com história prévia de fratura, com algum grau de comprometimento nas atividades de vida diária e entre os que não referem leitura como atividade de lazer.

(GASPARETTO; FALSORELLA; COIMBRA, 2014, p. 204)

Conforme Pereira, Ceolin, Neri (2013), um estudo com mais de 600 idosos mostrou uma prevalência de sintomas de insônia em quase metade dos participantes (49%) que sofreram quedas e entre aqueles que relataram cochilar durante o dia, a prevalência foi de 62%. Outra pesquisa relacionada a utilização de fármacos como benzodiazepínicos, neurolépticos e antidepressivos, em pacientes com osteoartrite aumenta o risco de quedas (ROLITA et al., 2013).

Gasparotto, Falsorella e Coimbra (2014), indicam que a diabetes mellitus apresenta consequências como decréscimo da função sensoriomotora, déficits neuromusculares e musculoesqueléticos e complicações farmacológicas. Em decorrência disso, a doença tem sido referenciada como de risco para quedas.

O acometimento do segmento estrutural do corpo é um fator que apresenta importante relação com a queda, por reduzir a estabilização postural e das sobrecargas impostas. A osteoporose é considerada o grau avançado de perda da densidade óssea e deixa tais estruturas menos compactas e mais porosas. A fase da osteopenia, primeiro estágio deste processo, já sugere o desequilíbrio funcional entre osteoblastos e osteoclastos, células que trabalham na deposição da absorção óssea e que, quando comprometidas, aumentam o risco de a queda gerar fraturas. É importante lembrar que cada sexo tem sua peculiaridade quanto ao tempo e fatores de acometimento. A mulher, por exemplo, deve ficar mais atenta na menopausa, em que os níveis reduzidos de estrógeno contribuem para acelerar esse processo. (GASPAROTTO; FALSORELLA; COIMBRA, 2014)

Demências como Alzheimer, Parkinson ou desordens que dificultam a marcha e reduzem a acuidade visual também estão inclusas como fatores neurológicos que predispõem a quedas pois alteram a função vestibular, de propriocepção e cognição. (GASPAROTTO; FALSORELLA; COIMBRA, 2014).

## 2.2 A PLATAFORMA ESPRUINO PICO

A Espruino Pico é uma placa de hardware livre, que foi um projeto financiado pelo Kickstarter anos após a primeira placa, chamada Espruino (KICKSTARTER, 2015). Ela é embarcada com suporte a vários componentes de terceiros, como acelerômetros, bluetooth, bateria recarregável externa, pinos para ser usado em protoboard, etc. (KICKSTARTER, 2015).

A Espruino original foi uma placa maior do que a Espruino Pico, com um processador ARM Cortex M3. Ela mede 5.4cm por 4.1cm. (Espruino, 2015). A Figura 1 ilustra o formato da placa.

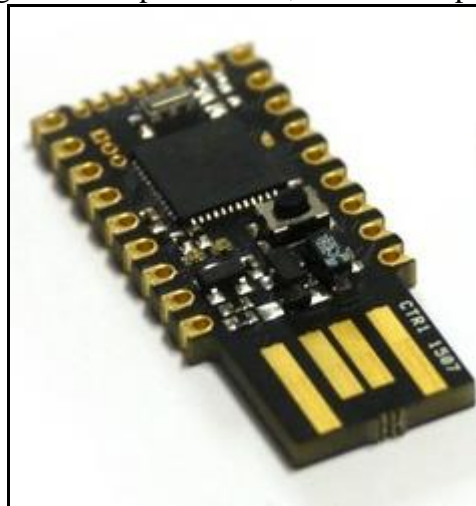
Figura 1 - Espruino original



Fonte: Espruino (2015).

Com o desenvolvimento da Espruino Pico, maior parte do suporte a módulos externos foi mantido. Entretanto, o processador é mais recente sendo um ARM Cortex M4 e outra diferença é o tamanho da placa, medindo 3.3cm por 1.5cm como ilustra a figura 2 (KICKSTARTER, 2015).

Figura 2 – Espruino Pico, versão sem pinos



Fonte: Espruino (2015).

A placa vem com um interpretador de Javascript embutido, sendo este o único meio de construir instruções e um programa para a placa. (Espruino, 2015). Com isto, ela já tem suporte a várias estruturas de dados. Os componentes suportados são possíveis de serem

usados graças a bibliotecas previamente construídas, onde ambos os módulos de acelerômetro e Wi-Fi são suportados pela placa. Para usar as bibliotecas basta utilizar o comando *require* e o nome da biblioteca a ser carregada, onde a referência do nome de cada biblioteca disponível é encontrada no site oficial do Espruino. O Quadro 1 mostra um exemplo de uma aplicação realizando conexão HTTP através do componente de Wi-Fi.

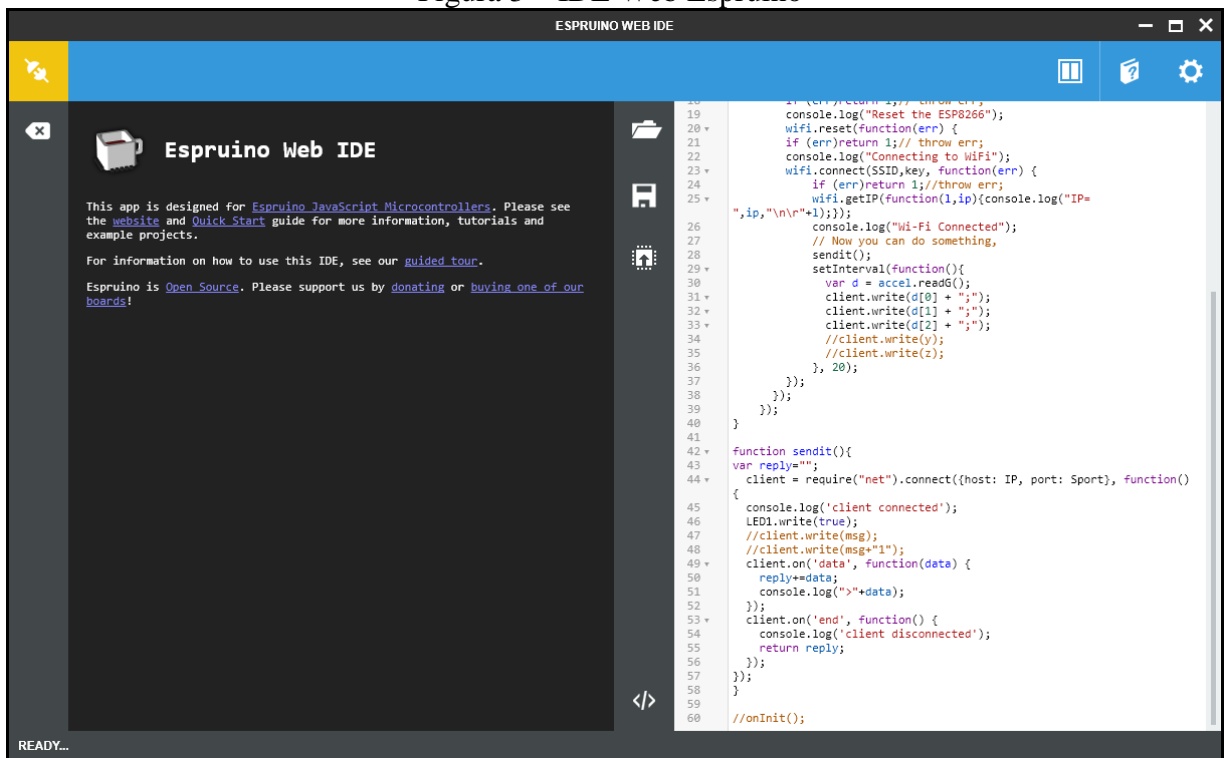
Quadro 1 – Código exemplo de conexão Espruino à ESP8266, conectar a wi-fi e fazer uma requisição HTTP

```
digitalWrite(B9,1); // habilitar uso através do Pico Shim V2
Serial2.setup(115200, { rx: A3, tx : A2 });
var wifi = require("ESP8266WiFi_0v25").connect(Serial2,
function(err) {
  if (err) throw err;
  wifi.reset(function(err) {
    if (err) throw err;
    console.log("Conectando WiFi");
    wifi.connect("WiFi_Name","WPA2_Key", function(err) {
      if (err) throw err;
      console.log("Conectado");
      require("http").get("http://www.google.com", function(res) {
        console.log("Resposta: ",res);
        res.on('data', function(d) {
          console.log("--->" +d);
        });
      });
    });
  });
});
});
```

Fonte: Do autor.

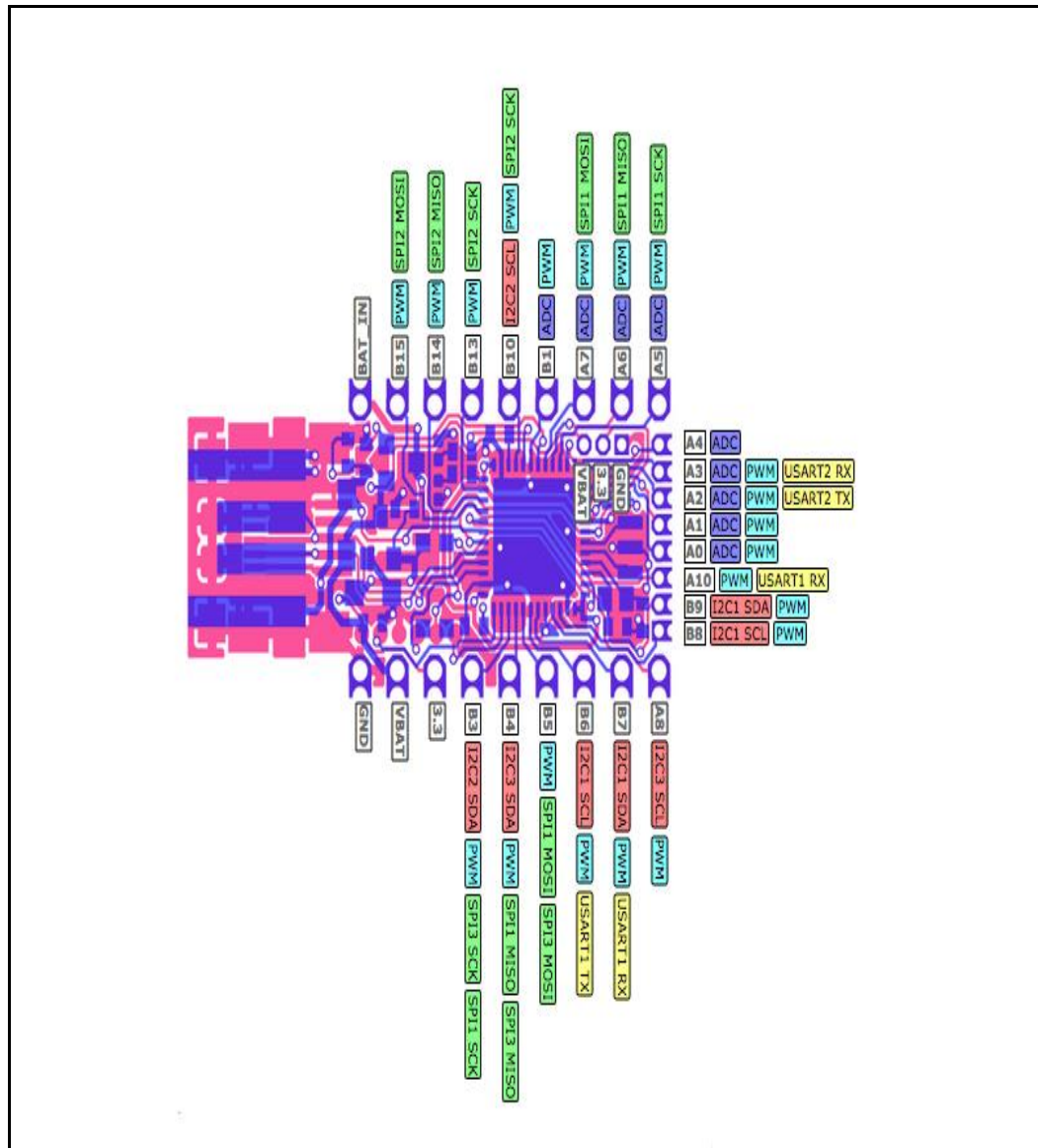
Estas estruturas podem ser construídas através da IDE disponível como plugin do navegador Google Chrome. A IDE, como pode ser visto na Figura 3, possui um terminal que permite enviar comandos ao Espruino assim como receber saída de console. Possui também um editor de código com *highlight* para o JavaScript. Segundo Williams (2017), não existe uma definição exata da compatibilidade quanto ao padrão ES5 ou ES6 do JavaScript, assim o Espruino possui quase todas as características do ES5 e alguns aspectos do ES6. Quando é necessário utilizar as bibliotecas, todas que se fazem necessárias são carregadas na hora da compilação, através da internet (PUR3, 2017).

Figura 3 – IDE Web Espruino



Os componentes podem ser soldados nas várias entradas e saídas disponíveis na Espruino Pico, conforme mostra a Figura 4. Além das entradas e saídas disponíveis nas laterais da placa, existem também 3 entradas e saídas, menores que as demais, disponíveis no meio. Todas as entradas marcadas de ADC, em azul, são *Analog to Digital Converters*, ou seja, recebem sinais analógicos e então são convertidos para digitais. GND, do inglês *ground* é o terra (0V), enquanto o 3.3V é a saída de 3.3V através do regulador de tensão. VBAT é para enviar qual a tensão de saída da bateria. PWM, em azul claro, é a sigla para Pulse Width Modulation, ou Modulação por Largura de Pulso. USART, em amarelo, é para o recebimento de dados em serial, identificado qual é o RX (receber) e qual é o TX (transmitir), utilizado para conectar a ESP8266 através do *shim*. (Espruino, 2015)

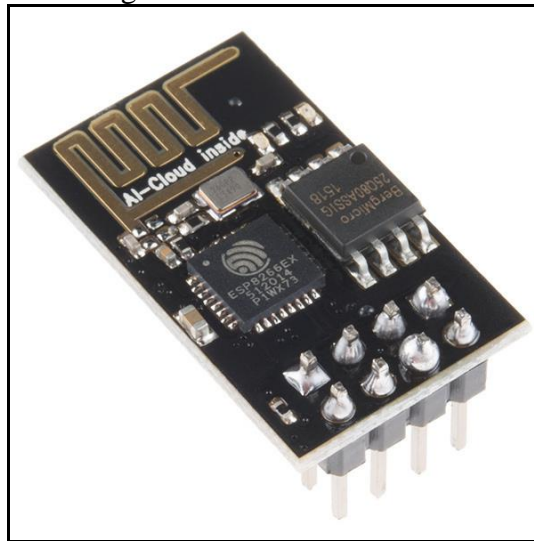
Figura 4 – Esquema elétrico da Espruino Pico



Fonte: Espruino (2015).

O módulo relevante para este trabalho para realizar a conexão à internet via wifi é a ESP8266. A arquitetura é chamada de ESP8266EX, produzida pela Expressif, permitindo vários meios de conexão e vários meios de programá-la. (EXPRESSIF, 2017) O modelo utilizado neste trabalho é a ESP-01 ESP8266, como mostra a Figura 5. Esta placa permite a utilização de comandos AT, que segundo a IEEE (2007), é o conjunto de instruções inicialmente desenvolvido pela empresa Hayes para permitir a comunicação entre computadores e modems.

Figura 5 – ESP-01 ESP8266



Fonte: Sparkfun Electronics (2017).

### 2.3 ALGORITMO PARA DETECÇÃO DA QUEDA

Apesar de uma queda ser facilmente observada e identificada, segundo Noury. et al. (2007) desde 1991 existe o desafio de construir algoritmos e mecanismos para detectar uma queda, seja através do uso de acelerômetros ou processamento de imagem.

Segundo Noury et al. (2007), uma queda frequentemente atinge o solo ou parede, em que ocorre um choque da colisão, causando uma intensa inversão do vetor de aceleração na direção da trajetória da queda. Um vetor segundo Silva Neto (2014) é a representação mais simples para representar informações algébricas. A partir deste comportamento, é possível monitorar os valores de aceleração a fim de detectar o pico causado pelo choque da colisão.

Entretanto, para trabalhar com a detecção de queda é necessário ter uma representação que seja mais simples. Isto é possível através da norma do vetor, que segundo Silva Neto (2014, p.7) “é o comprimento comum a todos os representantes do vetor”. A norma do vetor possui algumas propriedades. A norma do vetor é uma distância e por isso nunca será um número negativo, assim como a única possibilidade da norma ser zero é caso o vetor seja nulo, ou seja, os valores de cada representante do vetor sejam iguais. Caso a norma do vetor seja calculada a partir de qualquer vetor nulo até um ponto  $(x,y,z)$ , sua norma será dada pela raiz da soma dos quadrados dos representantes  $(x,y,z)$ .

Através da matemática de vetores, é possível obter um único valor para representar a aceleração dos três eixos conforme a Figura 6, uma propriedade da norma do vetor, onde  $n$  seja o momento obtido os valores do acelerômetro. Este único valor tem o propósito de facilitar a comparação de valores, visto que a queda pode se dar em diferentes eixos dependendo da direção da queda.

Figura 6 – Fórmula de norma do vetor

$$A_n = \sqrt{x_n^2 + y_n^2 + z_n^2}$$

Fonte: Silva Neto (2014).

Segundo Minervino (2013, p.19), “Filtros são dispositivos usados para selecionar a transmissão de sinais de determinadas frequências. Assim, em função de suas finalidades, eles podem ser classificados como: passa-baixa, passa-alta, passa-faixa e rejeita-faixa”. Os filtros de passa-baixa permitem apenas a transmissão da frequência abaixo da frequência de corte do filtro, impedindo a transmissão de sinais de frequência igual ou superior. Enquanto que os filtros de passa-alta são o oposto dos de passa-baixa, permitindo apenas a transmissão da frequência acima da frequência de corte do filtro, impedindo a transmissão de sinais de frequência inferiores. Os filtros de passa-faixa podem ser considerados uma união de passa-baixa e passa-alta, onde apenas uma faixa da frequência de transmissão é permitida, em que valores menores ou maiores de sinal de frequência são descartados. Por último, os filtros de rejeita-faixa podem ser associados como a inversão de um filtro de passa-faixa, onde apenas os valores que estejam acima ou abaixo da frequência de corte do filtro não são descartados. (MINERVINO, 2013)

Contudo, valores obtidos do acelerômetro podem vir com ruído, gerando valores que podem vir a atrapalhar e até mesmo acusar um falso positivo. Para isto, pode ser utilizado um filtro de passa baixa. Um filtro de passa baixa que pode ser usado é o de média móvel, que segundo Smith (2011), é o filtro de passa baixa mais utilizado, por ser o mais simples com o propósito de remover ruídos aleatórios, mas ainda mantendo as curvas acentuadas. Para este trabalho, é utilizado uma variação conforme Figura 7, baseando-se em considerar apenas uma parte da média para o valor final, definido por  $\alpha$ , onde  $1 > \alpha > 0$ .

Figura 7 – Fórmula de filtro de passa baixa baseado em média móvel

$$y_n = (\alpha y_{n-1}) + (x - (\alpha x_n))$$

Fonte: Smith (2011).

## 2.4 TRABALHOS CORRELATOS

Para o trabalho proposto foram selecionados três trabalhos correlatos, onde dois deles são com o objetivo de detecção de quedas de pessoas e outro sendo um dispositivo para o aprendizado de Física. Primeiramente é apresentado o fAlert, um aplicativo para Android para monitoramento de pessoas com cuidados especiais (PIVA et al., 2014). Em seguida, é apresentado o trabalho de uso de acelerômetro eletrônico e a placa Arduino para ensino de Física em tempo real (ROCHA; MARRANGHELO, 2014), e por último “Evaluation of a threshold-based tri-axial accelerometer fall detection” (BOURKE; O’BRIEN; LYONS, 2007).

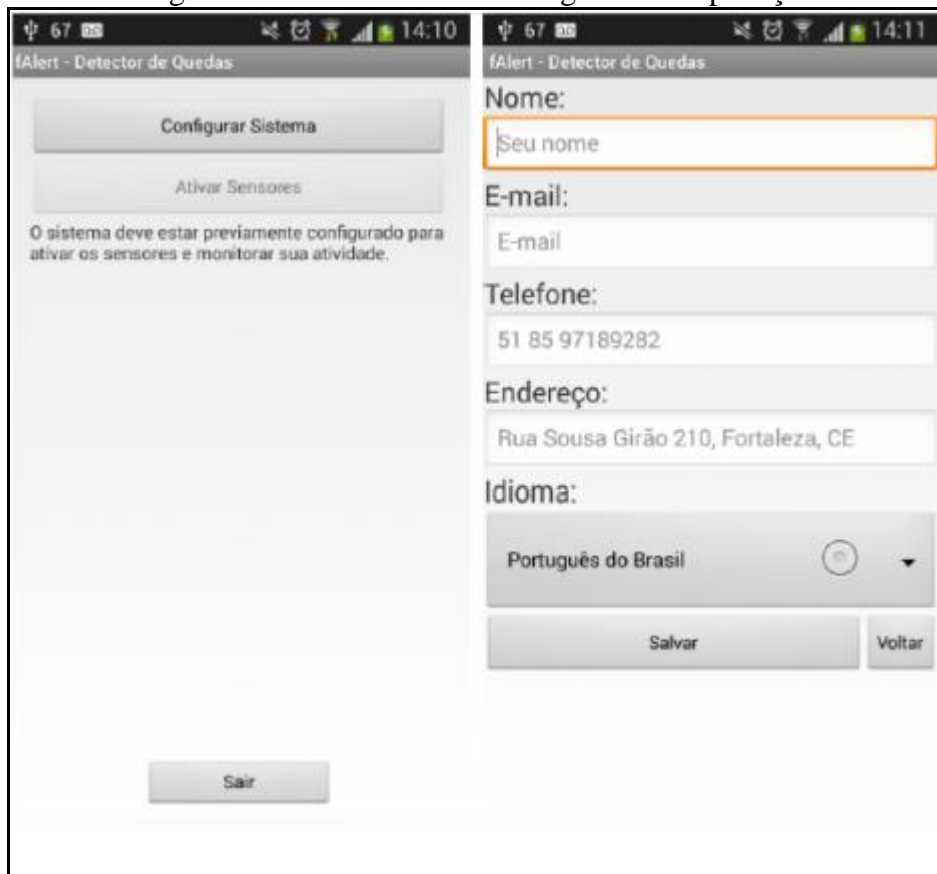


2.4.1 fAlert : Um sistema android para monitoramento de quedas em pessoas com cuidados especiais

Sistema desenvolvido em Android na Universidade Federal do Paraná. Foi desenvolvido em Java, utilizando máquina virtual Dalvik e banco de dados SQLite. O dispositivo usado para testar a aplicação desenvolvida foi o Samsung Galaxy S3 I19300.

A interface gráfica foi reduzida a somente 2 botões para realizar a configuração do aplicativo. Um dos botões era para a escolha de linguagem, com três opções diferentes. A aplicação lia os dados do acelerômetro mais de 40 vezes por segundo utilizando uma configuração disponível nativamente do Android. Uma tela de exemplo pode ser vista na Figura 8.

Figura 8 – Desenho da interface gráfica da aplicação



Fonte: Piva et al. (2014)

O sistema realiza uma verificação de queda com os dados do acelerômetro dos últimos 1,5 segundos. Para realizar esta verificação, é calculado o vetor de aceleração linear e então calculado o valor mínimo e máximo obtido na faixa de tempo. Se o máximo e mínimo estão além dos parâmetros definidos e o ângulo do dispositivo está em até 45° em relação ao chão, o sistema notifica como queda.

Segundo Piva et al. (2014, p. 113), “foi identificado que alterando a sensibilidade pode-se chegar a 100% de eficiência do algoritmo para detecção de quedas, porém também aumenta a porcentagem de falsos positivos na ação de deitar”.

O trabalho testou vários parâmetros para chegar aos valores otimizados e conseguiu detectar 93% das quedas durante os testes. Ao mesmo tempo, existe 33% de falso positivos causados quando o usuário deita.

As recomendações deste trabalho são para que seja feita uma calibração automática dos limites que definem a detecção da queda, visto a grande heterogeneidade de dispositivos que utilizam o sistema operacional Android. Outra frente se dá para o aprendizado de máquina para conseguir diferenciar com maior eficiência quedas de ações como “deitar” e “cair”.

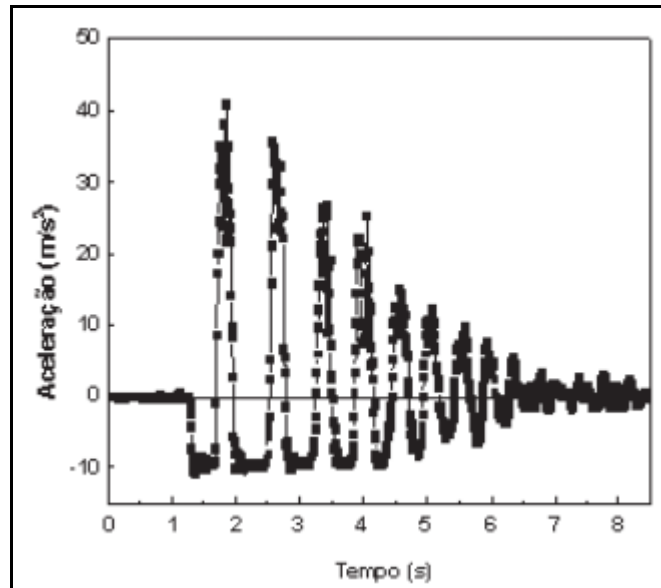
#### 2.4.2 Acelerômetro eletrônico e a placa Arduino para ensino de Física em tempo real

Conforme Rocha e Marranghelo (2014), o intuito deste trabalho era prover um equipamento para a realização do ensino ativo de física nas escolas. Com o uso do equipamento seria possível levantar discussões entre os alunos através dos resultados obtidos nos experimentos.

Foi utilizada uma placa Arduino ligada via USB em um computador, sendo que o Arduino servia apenas de coletor de dados e todo o processamento era realizado pelo computador para se tornar possível a experiência de física em tempo real. A placa Arduino estava ligada a uma placa auxiliar onde é conectado o acelerômetro para realizar a coleta de dados.

Um dos testes realizados com o equipamento foi o *Bungee Jump* a fim de mostrar a variação de aceleração durante a queda do dispositivo em um objeto como um pêndulo em uma mola. Os resultados dos testes podem ser observados na Figura 9. Foi observado um comportamento em que o objeto e a mola dissipam energia a cada oscilação apresentada no gráfico. (ROCHA; MARRANGHELO, 2014).

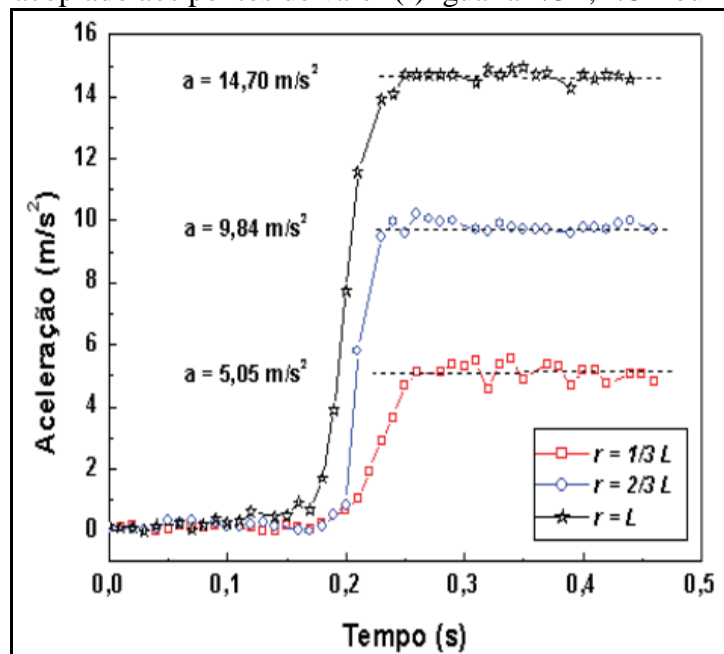
Figura 9 - Monitoramento da aceleração do corpo em queda, utilizando cabo elástico com 8g de massa.



Fonte: Rocha; Marranghelo (2014).

Outro teste realizado foi colocar a placa junto a uma haste rodando 90°, ida e vinda. O equipamento é ligado em 3 pontos diferentes, onde que a distâncias destes pontos é idêntica entre eles, afim de demonstrar as variações de aceleração obtidas no experimento. Os resultados obtidos estão apresentados no gráfico da Figura 10. Foi observado que os valores estão com maior desvio em 1/3L em relação ao valor esperado, que é definido pela linha tracejada. (ROCHA; MARRANGHELO, 2014)

Figura 10 - Gráfico de variação da aceleração tangencial ocorrida com o acelerômetro quando acoplado aos pontos de valor (r) igual a 1/3L, 2/3L ou L



Fonte: Rocha; Marranghelo (2014).

A conclusão segundo Rocha e Marranghelo (2014) foi,

As práticas mostradas podem ser desenvolvidas mesmo em laboratórios modestos de ensino de Física. As características do acelerômetro podem ser aproveitadas tanto com a finalidade de inovar e incrementar práticas de laboratório como quanto para incentivar estudantes a trabalhar em projetos interdisciplinares. (ROCHA, MARRANGHELO, 2014, p. 118)

#### 2.4.3 Evaluation of a threshold-based tri-axial accelerometer fall detection algorithm

Conforme descrito por Bourke, O'Brien, Lyons (2007) foi construído um equipamento a partir de dois acelerômetros bixiais, modelo ADXL210, sendo que um dos dois foi colocado ortogonalmente ao outro. Com isto, apenas um eixo se repete, enquanto o outro acaba por representar o terceiro eixo não disponível no outro acelerômetro.

Até a data da publicação deste trabalho, os sistemas de detecção de queda tinham usado apenas jovens para testar os falsos positivos de atividades do cotidiano. Entretanto, os idosos frequentemente se movem diferente dos jovens já que eles tem menos controle da velocidade dos movimentos do corpo decorrente da perda da força muscular. Por isto foi considerado apropriado que os testes de atividades do cotidiano fossem feitas com participantes idosos para aumentar a confiabilidade da metodologia dos testes. (BOURKE; O'BRIEN; LYONS, 2007)

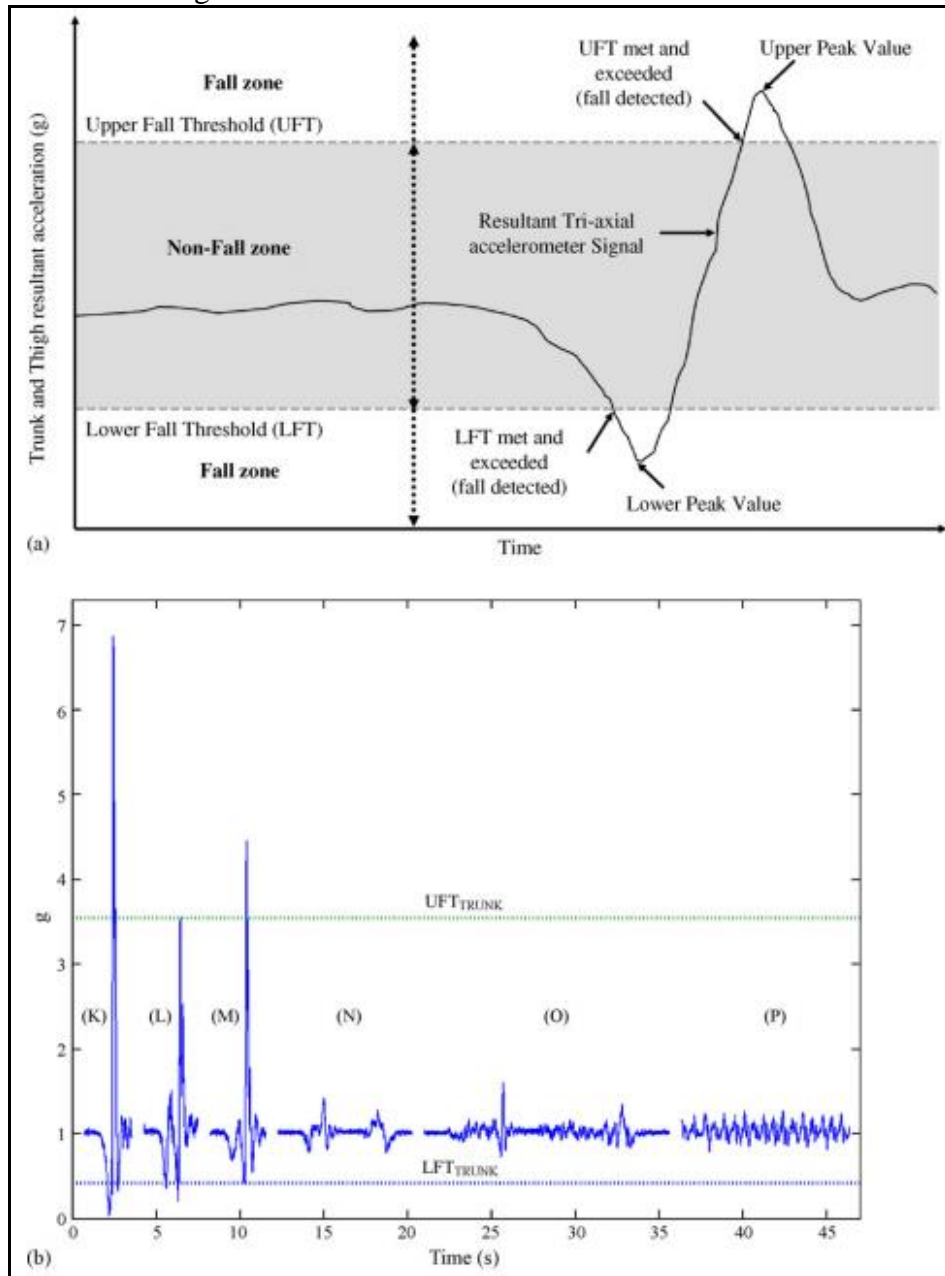
Para os testes de queda foram envolvidos 10 jovens saudáveis executando quedas simuladas em colchonetes de ginástica. Eles caíam de uma plataforma construída especificamente para os testes sob a supervisão de um profissional de educação física. Eles realizaram oito tipos diferentes de cada sendo cada um deles realizado três vezes. (BOURKE; O'BRIEN; LYONS, 2007)

O trabalho teve 10 participantes idosos para realizar avaliações do equipamento em suas atividades diárias. O equipamento era disposto com um acelerômetro sendo colocado no tronco, por cima do esterno, enquanto o outro era colocado perto da cintura, logo no topo do fêmur. O equipamento era amarrado aos usuários utilizando velcro e tiras elásticas. Cada um deles realizou oito atividades diferentes, três vezes cada, que possam vir a gerar um impacto e possivelmente um falso positivo. (BOURKE, O'BRIEN, LYONS; 2007).

Com os dados coletados foi construídos os gráficos da Figura 11. O primeiro gráfico mostra um exemplo de queda, explicando a detecção e o padrão reconhecido. Inicialmente os dados estão na área cinza, que não é considerada uma queda. Quando o valor ultrapassa o valor mínimo uma queda é detectada, pois isto significa que uma queda está ocorrendo. Caso o valor ultrapasse o valor máximo uma queda também é detectada, pois isto significa que a queda já ocorreu, sendo o valor um pico gerado pelo impacto (BOURKE, O'BRIEN, LYONS; 2007).

O segundo gráfico da Figura 11 representa vários valores obtidos para comparação. Os valores demonstrados em K é de uma queda típica, os em L são da queda com o menor valor obtido pelo dispositivo na cintura, os em M são da queda com o menor valor obtido pelo dispositivo no torso, os em N são da atividade de sentar em uma cadeira, os em O são de sair e entrar do banco de um carro e por últimos os valores em P são de caminhada.

Figura 11 – Resultados obtidos através dos testes



Fonte: Bourke; O'Brien, Lyons (2007).

O método utilizado para avaliação de queda se assemelha ao do trabalho apresentado na seção 2.4.1, visto que são determinados limites fixos para mínimo e máximo. Primeiramente são aplicados alguns filtros nos valores, sendo um deles o filtro Butterworth. São avaliados

ambos os acelerômetros para realizar a detecção de queda e caso o cálculo de aceleração vetorial supere os limites, é considerado como queda (BOURKE, O'BRIEN, LYONS; 2007).

Com os experimentos realizados, foi concluído que o melhor local para o uso do acelerômetro é no tronco do usuário, conseguindo detectar todas as quedas. Atingiu-se também uma margem de erro considerável para a distinção de atividades rotineiras de uma queda com os limites que foram obtidos pelo trabalho, onde a atividade que atingiu um maior valor foi “sentar na cadeira da cozinha”, mas ainda assim mantendo uma margem de 14.3% até que esta atividade fosse detectada como uma queda (BOURKE, O'BRIEN, LYONS; 2007).

### 3 DESENVOLVIMENTO DO PROTÓTIPO

Este capítulo visa apresentar o desenvolvimento do protótipo para detecção de queda. Primeiramente na seção 3.1, são apresentados os requisitos funcionais e não funcionais levantados para realizar o desenvolvimento, tendo sua especificação logo em seguida na seção 3.2, apresentando a arquitetura básica definida para a construção do dispositivo e da aplicação servidor. Na seção 3.3 são apresentadas as ferramentas e técnicas utilizadas pela implementação. Por último, a seção 3.4 possui a análise dos resultados obtidos com presente trabalho.

#### 3.1 REQUISITOS

O projeto proposto deve atender aos Requisitos Funcionais (RF) e Requisitos Não Funcionais (RNF) apresentados no Quadro 2 e Quadro 3, respectivamente. Os requisitos funcionais estão relacionados aos casos de uso (UC) apresentados na Figura 12.

Quadro 2 – Requisitos funcionais

Requisitos funcionais (RF)	Casos de Uso (UC)
RF01: monitorar através de um acelerômetro o usuário	UC01
RF02: conectar ao servidor para realizar a transmissão de dados	UC01
RF03: transmitir os dados do acelerômetro para o servidor assim que disponível	UC01
RF04: realizar os cálculos e tratativas sobre os dados do acelerômetro para identificar uma possível queda	UC02
RF05: alertar na aplicação servidor caso uma queda ocorra	UC03

Fonte: Do autor.

Quadro 3 – Requisitos não funcionais

Requisitos não funcionais (RNF)	Casos de Uso (UC)
RNF01: monitorar os dados do acelerômetro o mais próximo do tempo real possível	UC01
RNF02: notificar as quedas em até 1 minuto	UC03

Fonte: Do autor.

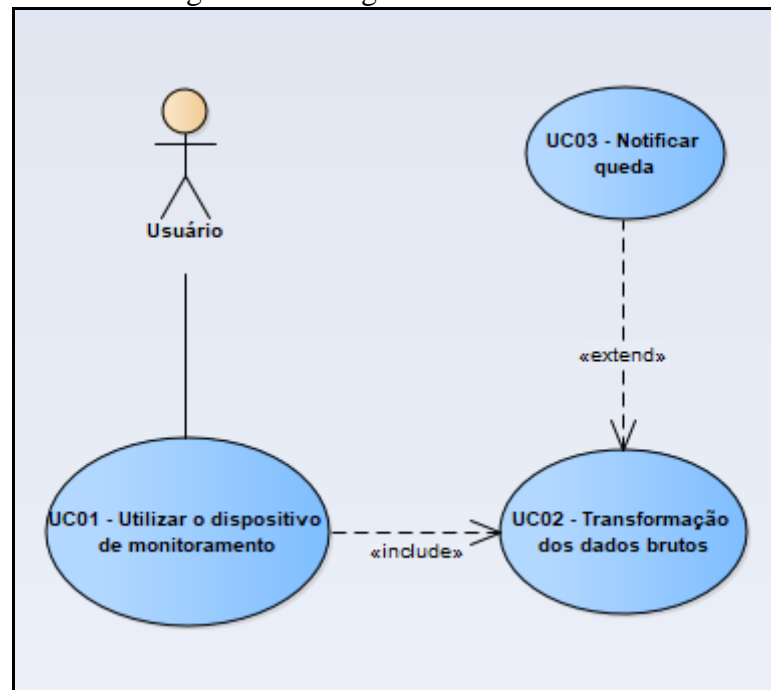
#### 3.2 ESPECIFICAÇÃO

Para o presente trabalho foram elaborados três casos de uso. A Figura 12 apresenta o diagrama de casos de uso, construído a partir da UML (Unified Modeling Language). Também foram elaborados dois diagramas de atividades seguindo o UML, para demonstrar o funcionamento do dispositivo, conforme Figura 15, e para demonstrar o funcionamento da aplicação do servidor, conforma Figura 17.

No diagrama da Figura 12, o caso de uso *Utilizar o dispositivo de monitoramento* é a única ação disponível para o usuário, em que envolve certificar-se que o aparelho esteja funcional e preso ao corpo, e caso esteja, então ligar o dispositivo. O

dispositivo ficará responsável por coletar os dados enquanto o usuário o estiver usando. Ao mesmo tempo, estes dados devem ser transmitidos ao servidor, para que possa ser executado o caso de uso *Transformação dos dados brutos*. Este caso de uso é responsável por fazer a limpeza dos dados, e os devidos cálculos para que seja obtido o valor que deve ser usado para verificar a queda. Caso o sistema reconheça como uma queda, o caso de uso *Notificar queda* é acionado, em que ele proporciona um meio de que haja uma notificação do ocorrido.

Figura 12 – Diagrama de caso de uso

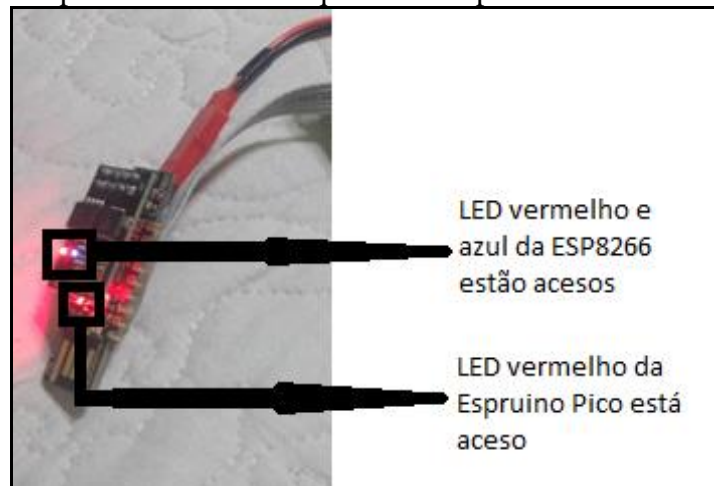


Fonte: Do autor.

O projeto tem por objetivo ter a menor dependência possível do usuário para seu funcionamento correto, logo, as únicas responsabilidades passadas para o usuário final são recarregar a bateria e manter o dispositivo preso ao corpo sempre que a bateria não esteja sendo recarregada. Assim que o dispositivo é ligado, a Espruino Pico iniciará os processos de inicialização dos dois componentes, a ESP8266 e o ADXL335. Deve existir um feedback para o usuário da situação do dispositivo, como mostra a Figura 13 e Figura 14, onde os LEDs ajudam ao usuário para saber se o dispositivo está funcionando devidamente. O ESP8266 terá o LED azul ligado caso tenha obtido conexão a wi-fi, caso contrário o LED vermelho estará ligado. A Espruino Pico acenderá o LED vermelho sempre que todas as inicializações forem finalizadas. O dispositivo procura conectar-se assim que a inicialização é finalizada, sem depender de nenhuma entrada do usuário.



Figura 13 – Dispositivo e os LEDs quando o dispositivo funciona normalmente



Fonte: Do autor

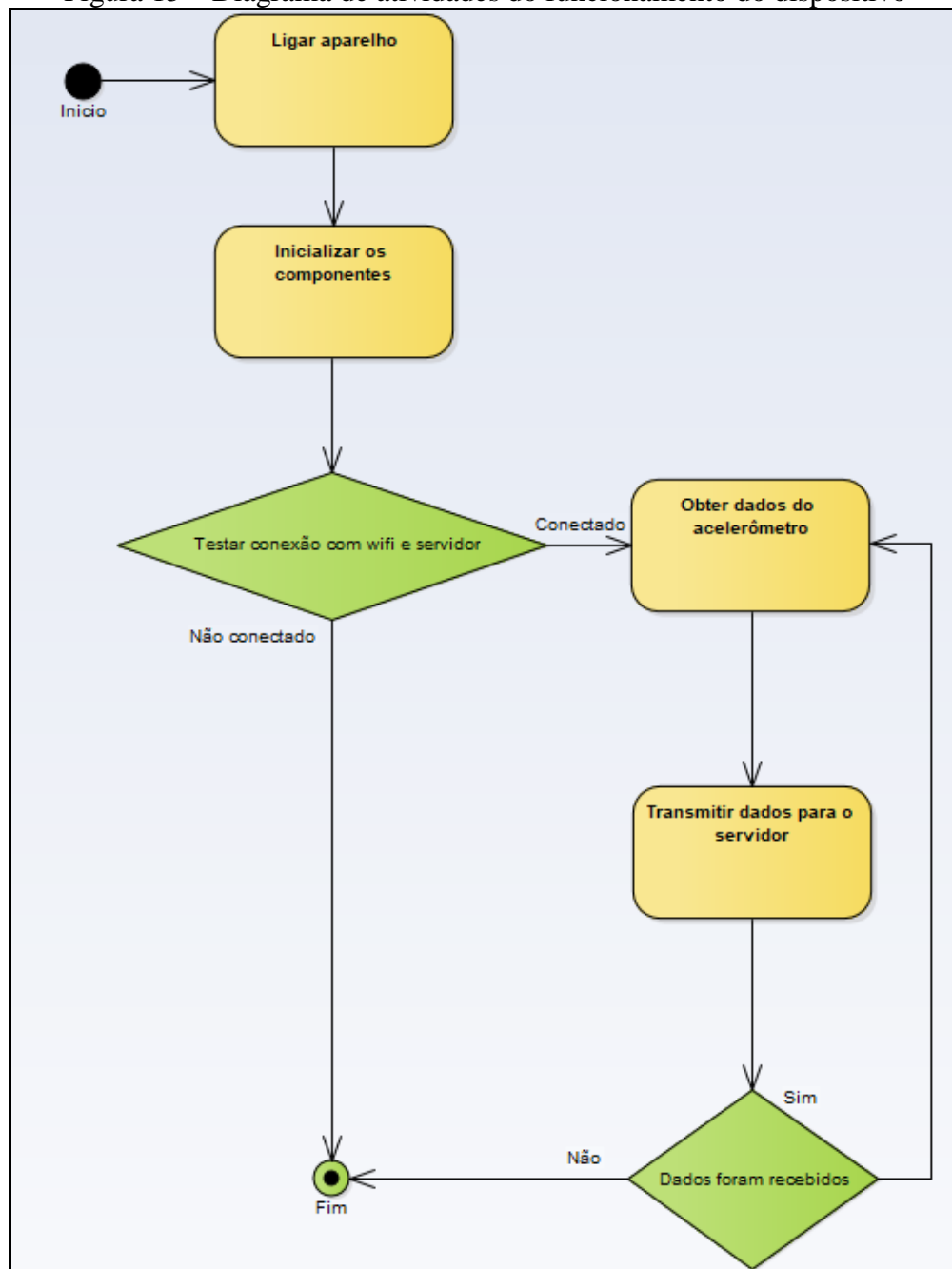
Figura 14 – Dispositivo e os LEDs quando o dispositivo não consegue conectar



Fonte: Do autor.

O diagrama de atividades mostra o funcionamento básico interno do dispositivo. Assim que o dispositivo é ligado pelo usuário, é automaticamente inicializado os componentes, a placa wifi e o acelerômetro, para então testar a conexão com a wifi e o servidor TCP. Caso isto não seja possível, o fluxo é encerrado, caso contrário, o fluxo segue para que os dados do acelerômetro sejam obtidos e então transmitidos. Transmitido os dados, se a conexão ainda estiver disponível e os dados foram enviados, é reiniciado o ciclo a partir da coleta dos dados.

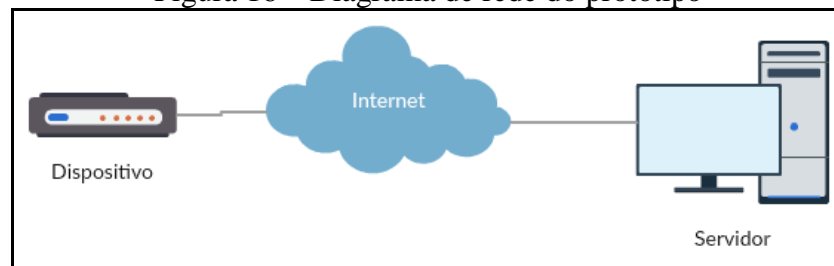
Figura 15 – Diagrama de atividades do funcionamento do dispositivo



Fonte: Do autor.

A comunicação do dispositivo é realizada via o protocolo TCP ao servidor para enviar os dados, através da internet, mantendo-o atualizado para fazer os cálculos de detecção de queda, conforme mostra a Figura 16. O servidor ficará aguardando conexões, e assim que o dispositivo o aciona, ele tentará a conexão com o servidor para iniciar o processo de monitoramento do usuário. A aplicação mantém a conexão aberta o tempo todo para que o monitoramento seja feito em tempo real. Os dados do acelerômetro são enviados ao socket TCP assim que são lidos.

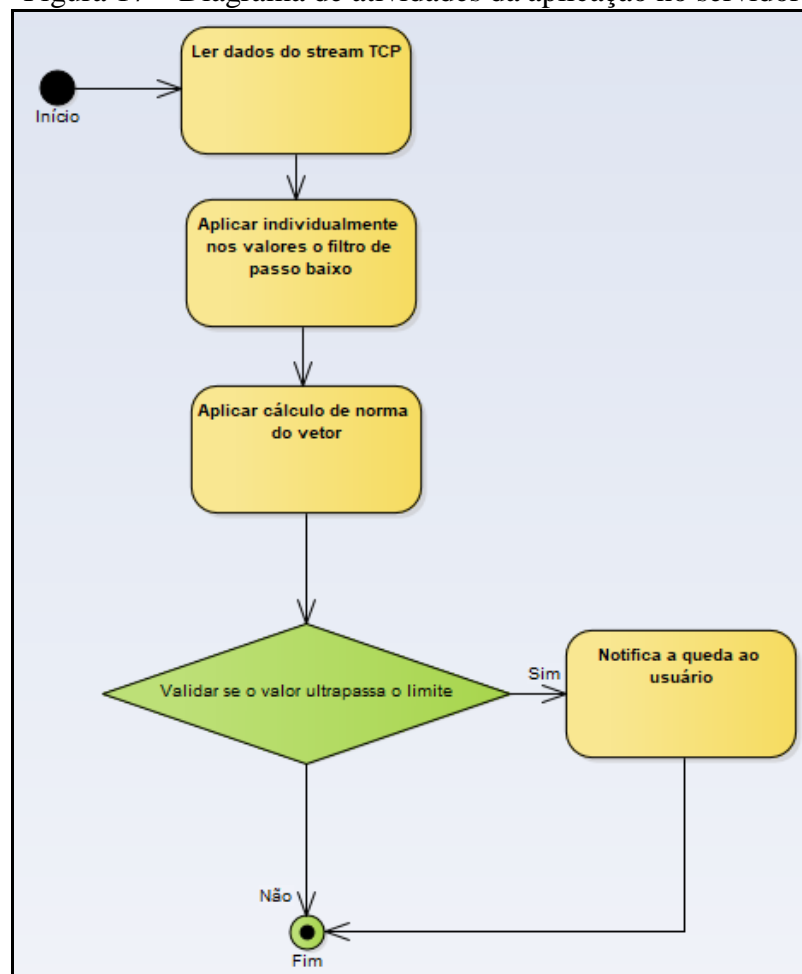
Figura 16 – Diagrama de rede do protótipo



Fonte: Do autor.

A Figura 17 mostra o funcionamento da aplicação no servidor de retaguarda a cada ciclo. Os bytes vão sendo lidos através do stream do TCP, e sempre que o dado completo é recebido através da desserialização do valor, é aplicado o filtro de passa baixa para atenuar os valores, mas sem perder os picos, garantindo assim que ruídos não sejam interpretados como queda. Após isto, os três valores, um de cada eixo, são transformados e um único valor seguindo a fórmula de norma do vetor. Com os valores calculados é verificado se o valor passa do limite estabelecido de 4m/s. Todo este processo é feito a cada ciclo de execução da coleta de dados que acontece no dispositivo.

Figura 17 – Diagrama de atividades da aplicação no servidor



Fonte: Do autor.

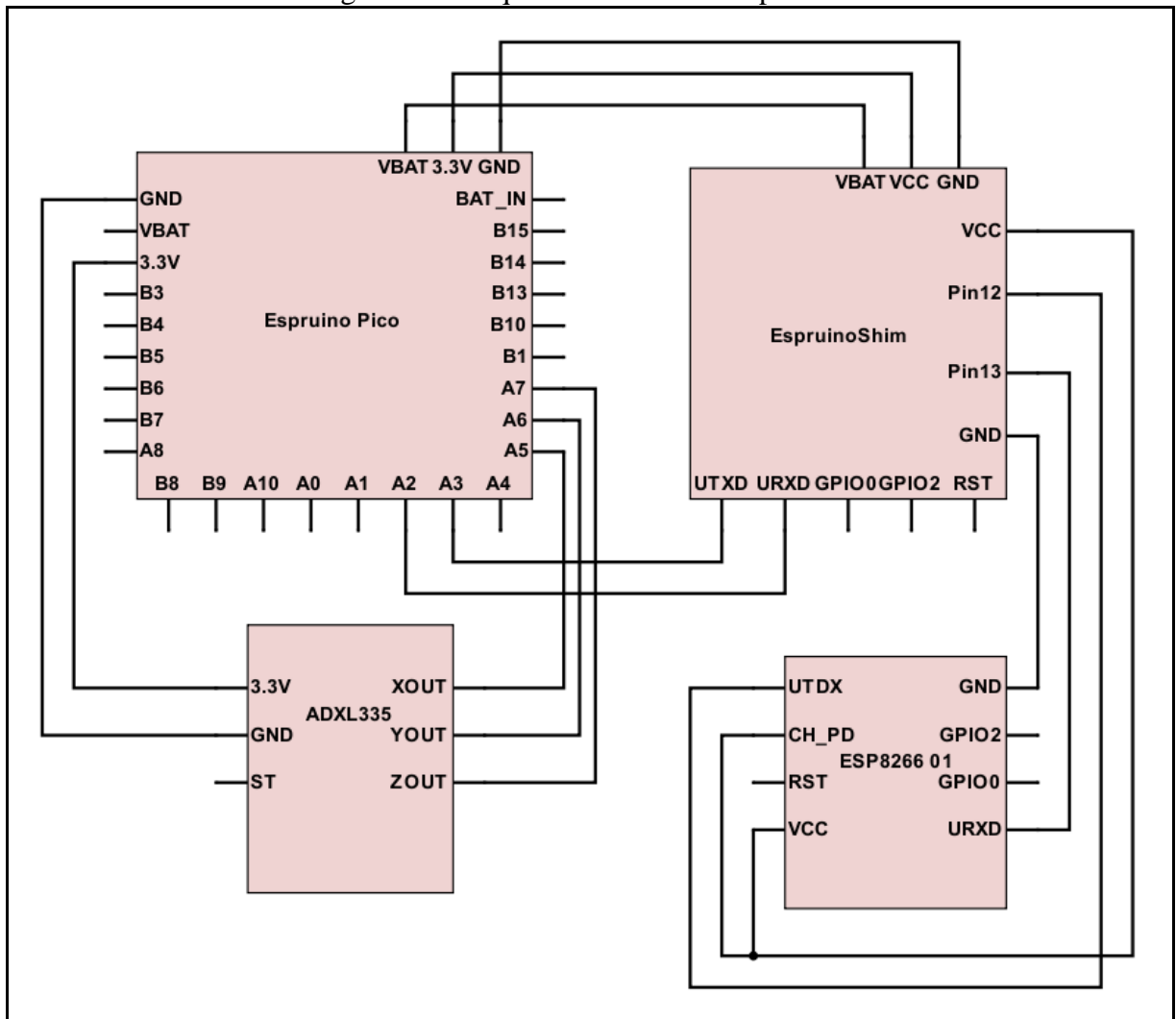
### 3.3 IMPLEMENTAÇÃO

A seguir são mostradas a construção do dispositivo e as técnicas e ferramentas utilizadas.

#### 3.3.1 Construção do dispositivo

Visto o tamanho menor da Espruino Pico ante a Espruino original e o processador mais recente, conforme foi apresentado na seção 2.2, este trabalho pretende utilizar esta placa para o desenvolvimento em toda a fase de implementação. Além da Espruino Pico serão utilizados também uma bateria de polímero de lítio, um módulo ESP8266 para Wi-Fi e o módulo ADXL335 de acelerômetro de três eixos. O esquema elétrico final do dispositivo é exibido na Figura 18.

Figura 18 – Esquema elétrico do dispositivo



Fonte: Do autor.

Foi definido o módulo ESP-01 ESP8266 para realizar a conexão do dispositivo a rede wi-fi, visto que a Espruino Pico já possui suporte nativo. Foi utilizado um *shim* que permite

realizar a solda da ESP8266 nos três pinos que se encontram no meio da Espruino Pico e os dois pinos de RX e TX para transmissão serial. Isto permite que comandos AT sejam enviados para a ESP8266. Apesar disto, nenhum conhecimento prévio de comandos AT foi necessário, visto que a biblioteca pronta de abstração já contemplou todas as necessidades.

Para a coleta dos dados de aceleração, o acelerômetro de três eixos ADXL335 foi escolhido. Ele supre as necessidades visto que o trabalho não exige de cálculos que envolvam acelerações elevadas, e também pois já existe suporte nativo pela Espruino Pico.

A alimentação fica por baixo da placa, onde uma bateria de lítio de 3.7V e 1100mah foi utilizada e colada. A Espruino Pico fica responsável por alimentar os outros componentes, o que torna mais fácil a fiação e a solda visto que só é necessário a bateria conectada no microcontrolador.

Após todos estes procedimentos, foi aplicado silicone em todas as soldas. Isto foi feito para garantir que as soldas não se rompessem tão facilmente durante os testes das quedas. O resultado final da construção do dispositivo pode ser visto na Figura 19.

Figura 19 – Dispositivo após solda de todos os componentes



Fonte: Do autor.

### 3.3.2 Técnicas e ferramentas utilizadas

Primeiramente foi construído o código para o dispositivo, que é programado em Javascript e possui uma IDE própria. O código foi desenvolvido sendo apenas um script, contendo os métodos necessários para implementar as funcionalidades requeridas. O primeiro código a ser executado são todas as inicializações necessárias para que os componentes funcionem corretamente. Todo este procedimento foi retirado da documentação da Espruino Pico.

Para ativar as bibliotecas nativas é utilizado o comando `require`, passando parâmetro o nome da biblioteca, que pode ser obtido nas referências e documentação da Espruino. No caso, foram utilizadas as bibliotecas `ESP8266WiFi_0v25` e `ADXL335` que tratam de cada um dos componentes.

Por último, resta apenas a conexão com o servidor TCP e a transmissão dos dados. Isto foi realizado conforme mostra o Quadro 4. A primeira linha invoca a biblioteca nativa da Espruino Pico, e tenta realizar uma conexão ao IP e porta passados como primeiro parâmetro, e no segundo parâmetro é uma função de callback onde sempre que uma informação é enviada, recebida ou a conexão é interrompida, esta função é executada, em que no caso não foi utilizado. O método `setInterval` faz com que a função passada como primeiro parâmetro seja executada sempre após a quantidade de milissegundos que é definido no segundo parâmetro. Dentro deste código que irá se repetir indefinidamente, é buscado os valores do acelerômetro, que é retornado um vetor de três posições contendo um valor de ponto flutuante, e então são enviados para o stream do TCP, separados por ponto e vírgula. O motivo para os valores serem enviados separados por ponto e vírgula é discutido na implementação do servidor.

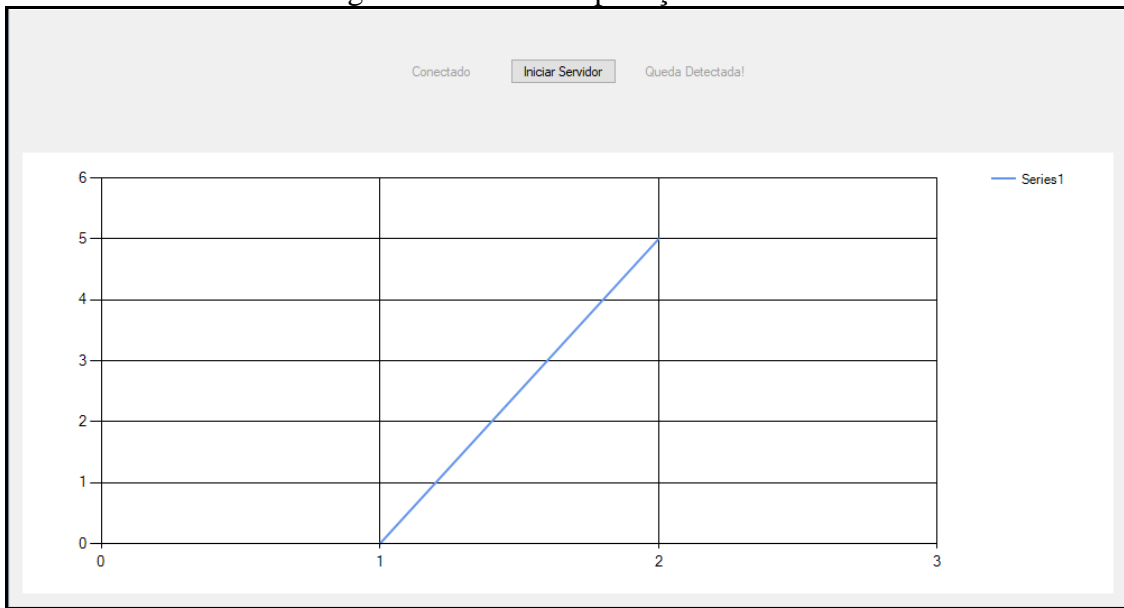
Quadro 4 - Código para conectar ao servidor TCP e leitura de acelerômetro

```
client = require("net").connect({host: IP, port: Sport},
function(){ ... }
setInterval(function(){
  var d = accel.readG();
  client.write(d[0] + ";");
  client.write(d[1] + ";");
  client.write(d[2] + ";");
}, 20);
```

Fonte: Do autor.

Para a aplicação servidor foi utilizado a linguagem C# e a IDE Visual Studio 2015. A aplicação servidor foi construída conforme a Figura 20, com uma tela dispendo de textos notificando se o dispositivo está conectado e se foi detectada uma queda, e um botão para iniciar o servidor TCP e aguardar a conexão do dispositivo. Além disto, também possui um gráfico, que mostra em tempo real os dados recebidos de aproximadamente os últimos 4 segundos. Toda implementação foi feita na classe `FallChartServer`.

Figura 20 – Tela da aplicação servidor



Fonte: Do autor.

Utilizando das classes já disponíveis no C# .Net nos *namespaces* `System.Net` e `System.Net.Sockets`, é aberto o servidor TCP com a classe `TcpListener` para receber os dados uma vez que a conexão é feita, como mostra o Quadro 5. As primeiras linhas mostram como é feita a inicialização do servidor TCP utilizando a classe `TcpListener`, sendo necessários passar os parâmetros de IP utilizando a classe `IPAddress`. Feito isto é criada uma instância da classe `TcpClient` através do método `AcceptTcpClient`, o qual impede o código de prosseguir enquanto não é obtida nenhuma conexão. Uma vez que o dispositivo se conecta ao servidor, é criada uma instância da classe `NetworkStream`, a qual permite que sejam lidos os bytes enviados pelo cliente. O `stream.Read()` permite que seja feita uma repetição, para que seja executado os códigos de tratamento de recebimentos de dados apenas se foi recebido algo.

Quadro 5 – Código que recebe a conexão TCP e lê os dados

```
IPAddress localAddr = IPAddress.Parse("IP");
server = new TcpListener(localAddr, port);
server.Start();

TcpClient client = server.AcceptTcpClient();
Console.WriteLine("Conectado!");
NetworkStream stream = client.GetStream();

int i = 0;

while ((i = stream.Read(bytes, 0, bytes.Length)) != 0)
{
    ...
}
```

Fonte: Do autor.

Uma vez que o stream é aberto, é realizado uma iteração sobre os bytes recebidos do stream, guardando os bytes e então transformando de volta em valores de ponto flutuante, como mostra a Figura 21. Para isto é primeiramente lidos os bytes em forma de `string`, depois é trocado os pontos por vírgulas da string, para que sejam lidos como números reais ao serem convertidos para `float`. Sempre que é encontrado um ponto-e-vírgula, isso significa que a variável `bytesRead` recebeu bytes suficientes para ser reestruturado em `float`. Isto é feito lendo todos os bytes até o caractere anterior a ponto-e-vírgula. A variável `ind` indica qual eixo este valor pertence, isto é feita dessa maneira visto que os dados são sempre enviados pelo cliente na mesma ordem: eixo X, Y e Z. Após isto é só feito um tratamento para limpar a parte da string que foi lida, mas caso o último caractere seja o ponto-e-vírgula, para evitar uma exceção, a string assumi uma string vazia.



Figura 21 – Código para transformar dados recebidos do *stream* e aplicar o filtro

```

while (bytesRead.IndexOf(";") > 0)
{
    switch (ind)
    {
        case 0:
            vector[ind] = lowPassX(Convert.ToDouble(bytesRead.Substring(0, bytesRead.IndexOf(";") - 1)));
            break;
        case 1:
            vector[ind] = lowPassY(Convert.ToDouble(bytesRead.Substring(0, bytesRead.IndexOf(";") - 1)));
            break;
        case 2:
            vector[ind] = lowPassZ(Convert.ToDouble(bytesRead.Substring(0, bytesRead.IndexOf(";") - 1)));
            break;
    }
    //vector[ind] = (Convert.ToDouble(bytesRead.Substring(0, bytesRead.IndexOf(";") - 1)));
    if (bytesRead.IndexOf(";") != bytesRead.Length - 1)
        bytesRead = bytesRead.Substring(bytesRead.IndexOf(";") + 1);
    else
        bytesRead = "";
    ind++;
}

```

Fonte: Do autor.

Os valores passam pelo filtro de passa baixa, como é possível ver os métodos de `lowPass` sendo invocados. A implementação do filtro é mostrada na Figura 22. Cada método possui sua própria variável para guardar o valor que será reutilizado, ou seja, o filtro é aplicado individualmente nos valores dos três eixos da aceleração.

Figura 22 – Código do filtro de passa baixa

```

public const double alpha = 0.1;
private double filterX = 0;
private double filterY = 0;
private double filterZ = 0;
private double lowPassX(double actualValue)
{
    filterX = (alpha * filterX) + (actualValue - (actualValue * alpha));
    return filterX;
}

private double lowPassY(double actualValue)
{
    filterY = (alpha * filterY) + (actualValue - (actualValue * alpha));
    return filterY;
}

private double lowPassZ(double actualValue)
{
    filterZ = (alpha * filterZ) + (actualValue - (actualValue * alpha));
    return filterZ;
}

```

Fonte: Do autor.

Realizado este procedimento, os dados são utilizados para realizar o cálculo de norma do vetor, como mostra o Quadro 6. Sempre que a variável `ind` atinge o valor 3 significa que o vetor leu os valores dos três eixos. Com estes dados, é aplicado o cálculo de norma de vetor

que foi discutido anteriormente. Caso o valor supere 4, então o texto de queda detectada é atualizada na tela, na cor vermelha, e logo em seguida um e-mail é enviado.

Quadro 6 – Código do cálculo de norma de vetor e verificação de queda

```

if (ind == 3)
{
    ind = 0;
    var calc = Math.Sqrt((vector[0] * vector[0]) + (vector[1] *
vector[1]) + (vector[2] * vector[2]));
    if(calc > 4.0)
    {
        label1.ForeColor = Color.Red;
    }
}

```

Fonte: Do autor.

Para demonstrar a funcionalidade de notificação, foi adotado a solução de envio de email. O e-mail é enviado através do protocolo SMTP (Simple Mail Transfer Protocol), conectando-se ao servidor de SMTP do Gmail através da classe `SmtpClient` do *namespace* `System.Net.Mail`. Após definido os parâmetros, é construído o objeto `Message`, para definir remetente e destinatário. Com este objeto, é definido também o título e conteúdo, e então, utilizado no método `Send` para enviar o e-mail. Todo este procedimento é exibido no Quadro 7.

Quadro 7 – Código para envio de e-mail via SMTP

```

SmtpClient email = new SmtpClient();
email.Host = "smtp.gmail.com";email.Port = 587;
email.EnableSsl = true;
email.DeliveryMethod = SmtpDeliveryMethod.Network;
email.UseDefaultCredentials = false;
email.Credentials = new NetworkCredential("usuario", "senha");
MailMessage mensg = new MailMessage("remetente", "destinatário");
mensg.Subject = "Uma queda foi detectada!";
mensg.Body = "Você está recebendo este e-mail porque o usuário do
dispositivo sofreu uma queda.";
email.Send(mensg);

```

Fonte: Do Autor.

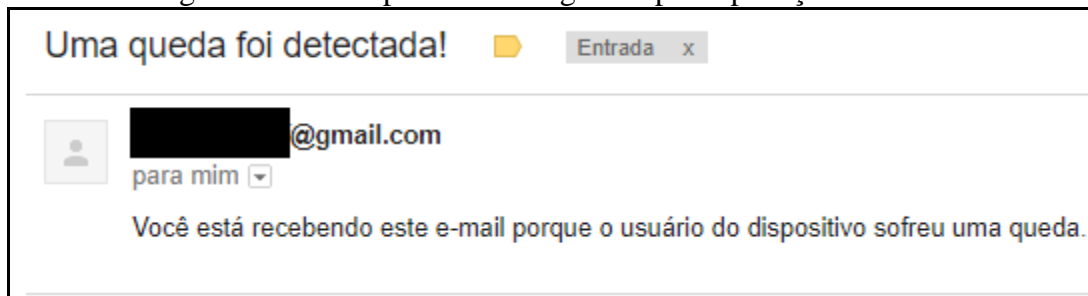
### 3.4 ANÁLISE DOS RESULTADOS

O trabalho aqui apresentado atingiu todos os objetivos propostos. Contudo, os testes de acessibilidade do equipamento para o usuário final não foram realizados. O equipamento não

é configurável, sendo necessário conectar ao computador com o código fonte para definir a rede a ser acessada. Ainda assim, o monitoramento consegue executar em tempo hábil, permitindo monitoramento em tempo real.

O envio de e-mail se provou confiável, pois o atraso até o recebimento é mínimo, e apenas em casos de problemas de rede o e-mail é sempre enviado. A Figura 23 apresenta o resultado final recebido pelo usuário.

Figura 23 – Exemplo de e-mail gerado pela aplicação servidor



Fonte: Do autor.

Com a transmissão dos dados sendo feita através de TCP, foram observadas algumas vantagens. A principal é a garantia da ordem de entrega dos dados, deixando um trabalho mínimo sobre o tratamento dos dados recebidos no servidor. Outra vantagem também foi o aproveitamento de componentes e bibliotecas já existentes que o implementam.

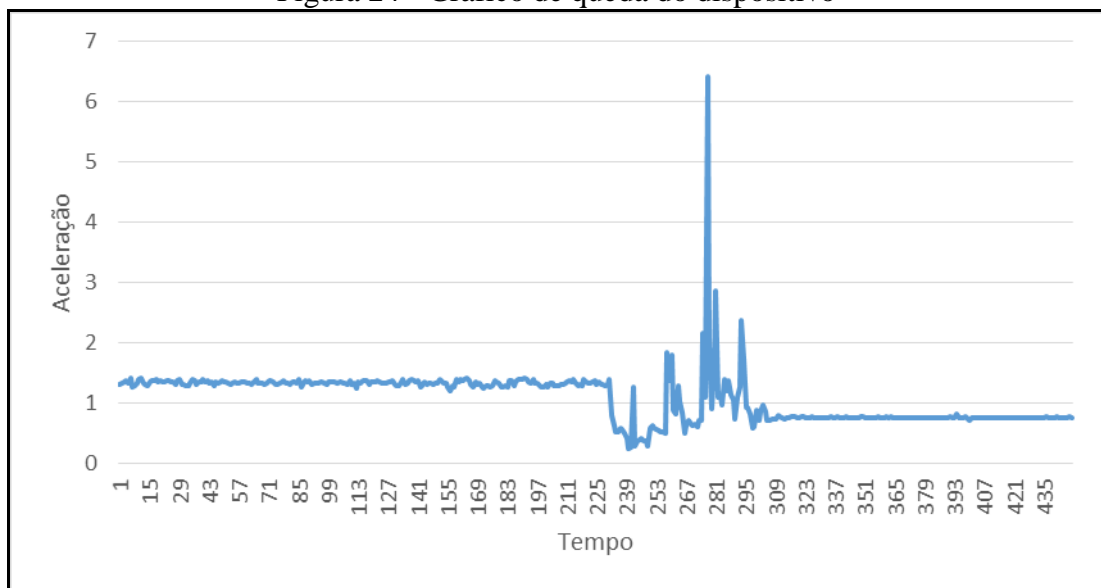
Com o uso do TCP também foi percebido algumas dificuldades em utilizá-la, sendo a primeira o seu funcionamento. Um socket TCP, após conectado cliente e servidor, não transmite os dados conforme é alimentado o stream, na verdade, quando e quantos bytes serão transmitidos fica por conta do próprio protocolo. Isto significa que os dados não são enviados completos da mesma maneira que é escrito no stream, e é por isso que a serialização se tornou necessária. Apesar de o C# .Net permitir uso amplo de ferramentas e bibliotecas para gerenciamento de socket, não se tem da mesma praticidade nas bibliotecas do Espruino, tornando o processo de serialização manual, assim como feito utilizando a concatenação de um ponto-e-vírgula no fim de cada valor enviado. Apesar de ser um método simples e não convencional, ele supriu as necessidades para que os dados fossem lidos corretamente no lado servidor.

O monitoramento é realizado a partir do momento em que a aplicação servidor esteja executando e o equipamento é conectado a bateria. A partir disto, o equipamento transmite os dados do acelerômetro a uma frequência média de 50Hz. Entretanto, também foi observado que a placa Espruino não tem memória suficiente para executar o código a mais de 50Hz, onde um erro de estouro de memória acontecia após alguns segundos de execução.

Com os dados em mãos, foi percebido o desafio de neutralizar o ruído gerado pelo acelerômetro, onde haviam picos nos valores obtidos. Por isto é aplicado o filtro de passa baixa, com  $\alpha$  de 0,1, um valor que se mostrou suficiente para separar um ruído de uma eventual queda nos testes realizados.

Foi observado que o acelerômetro absorve a força do impacto, gerando um pico nos valores. Estes valores tem uma pequena duração após o impacto, por isto a necessidade de executar o código do equipamento o mais rápido possível. A Figura 24 mostra o comportamento de uma queda de 1 metro do equipamento a um colchão:

Figura 24 - Gráfico de queda do dispositivo

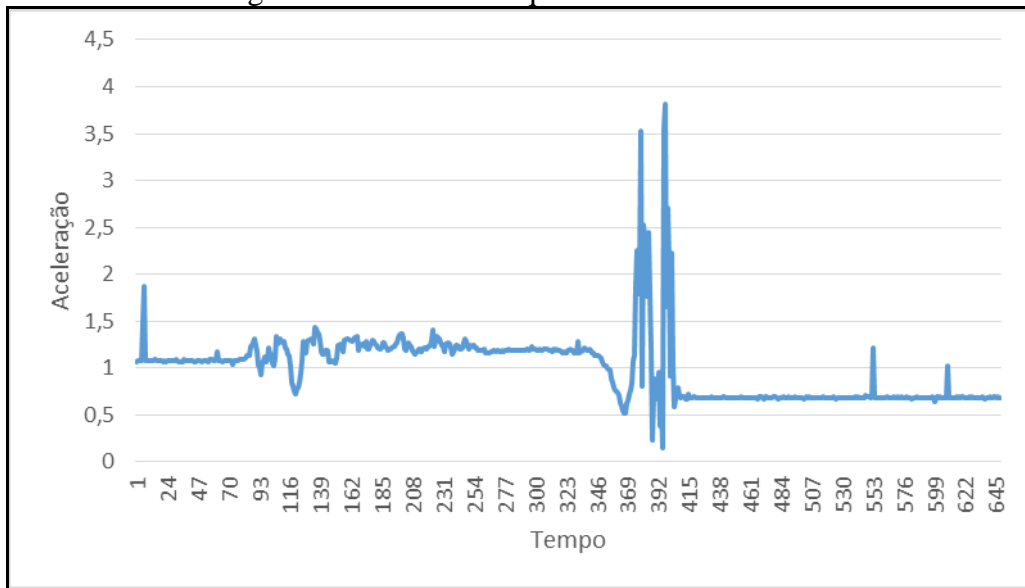


Fonte: Do autor.

Como é possível observar, o impacto de uma queda livre do equipamento teve seu pico acima de 6. Também foi observado alguns padrões, como os momentos em que o protótipo era carregado antes da queda, geram um valor com pouca variação, entre 1 e 2, enquanto após a queda, durante a inércia no solo, o valor é entre 0 e 1. Os valores observados durante a queda, a redução para quase zero seguido do pico, dependem do quão rápido e por quanto tempo a queda dura.

Para buscar o valor ideal que separa o que é uma queda e o que não é, foram realizados testes de queda utilizando o protótipo preso a uma vassoura, conforme Figura 25. Observando os valores, foi inconclusivo, pois a vassoura, por ser muito leve e pelo formato, quica ao colidir com o solo, fazendo com que o pico seja menor, mas o tornando mais prolongado.

Figura 25 – Gráfico de queda de uma vassoura



Fonte: Do autor.

Com o fim de otimizar este teste, foi preso a vassoura, além do dispositivo, duas chaves inglesas para aumentar a massa. Outra otimização também realizada é que a vassoura agora cai sobre uma almofada pequena, para amortecer o mínimo possível a queda e evitar que fique quicando ao ocorrer o choque com o solo. O resultado dessas alterações pode ser visto na Figura 26.

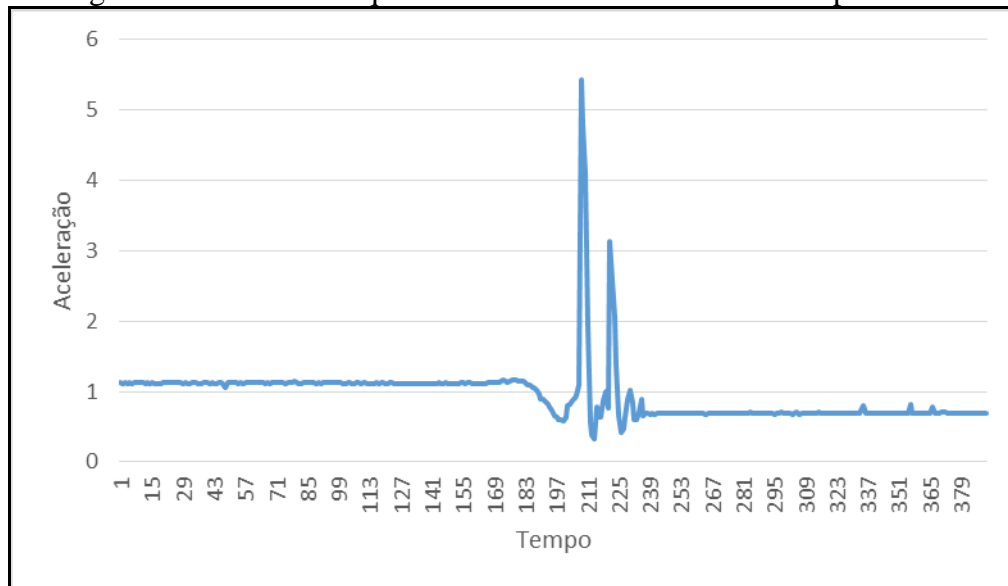
Figura 26 – Vassoura acoplada do dispositivo e de peso extra para testes



Fonte: Do autor.

Feito isto, o teste foi refeito, e os resultados obtidos podem ser vistos na Figura 27. A vassoura quicou uma única vez, graças ao amortecimento mínimo da queda pela almofada e do peso extra preso à vassoura. Isto fez com que o primeiro impacto seja de um valor mais significativo, apresentando o resultado desejado e fazendo com que passe do limite de 4m/s, detectando a queda.

Figura 27 – Gráfico de queda da vassoura na almofada com peso extra



Fonte: Do autor

Para obter melhores resultados, o protótipo foi preso a uma pessoa, na altura do peito com fita adesiva para manter firme o dispositivo, conforme Figura 28. Simulando a queda desta maneira, reproduziu melhores resultados por se tornar mais próxima de uma situação real.

Figura 28 – Dispositivo preso na altura do peito

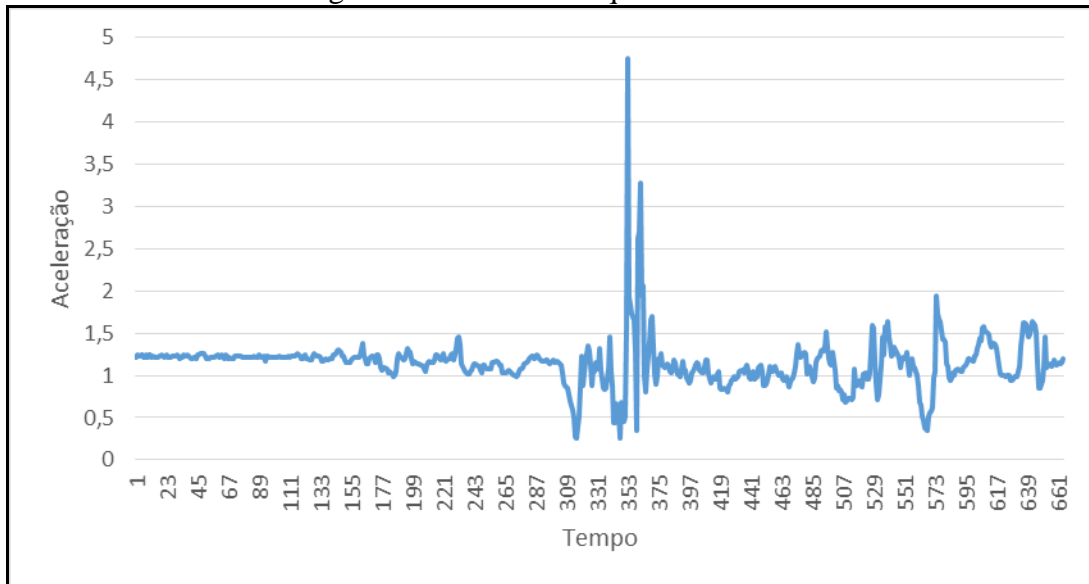


Fonte: Do autor.

A pessoa com o dispositivo preso ao corpo simulou uma queda sobre almofadas para amortecer o mínimo possível a queda. Conforme a Figura 29, esta queda foi um cenário mais aproximado do real, visto que há apenas um único pico, causando o valor maior.

Considerando uma margem de erro visto que são inúmeras as formas de uma pessoa cair, foi adotado como limite 4 o valor da aceleração para notificar sendo uma queda.

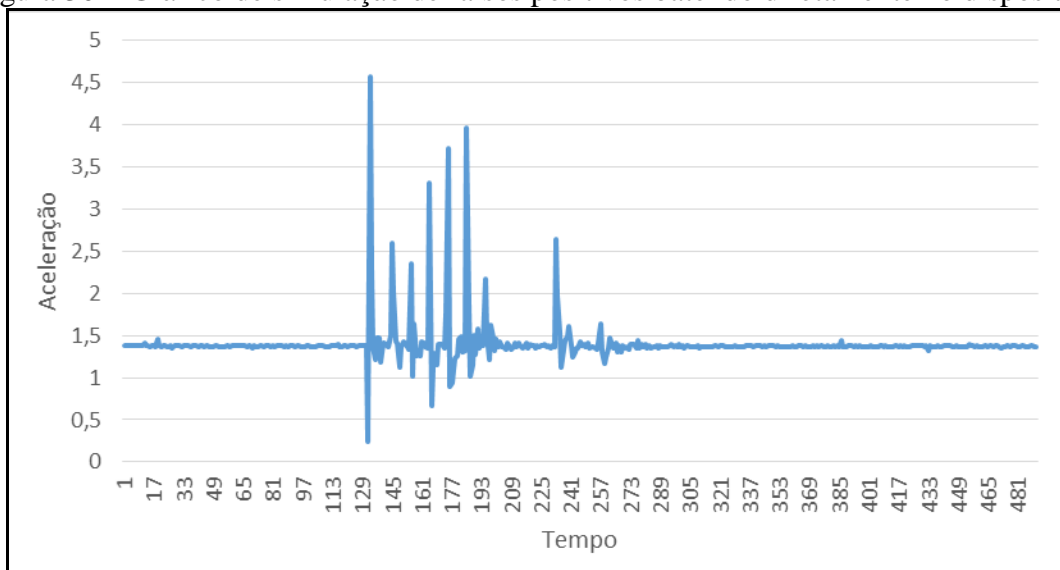
Figura 29 – Gráfico de queda simulada



Fonte: Do autor.

O protótipo também foi testado para falsos positivos, onde o protótipo recebia pequenas batidas com a mão, para simular colisões do protótipo caso o usuário acidentalmente colidisse com o dispositivo em algo. Como se pode observar na Figura 30, o sistema acabaria reconhecendo um dos impactos como uma queda, pois um dos valores passam do limite de 4, sendo falso positivo.

Figura 30 – Gráfico de simulação de falsos positivos batendo diretamente no dispositivo

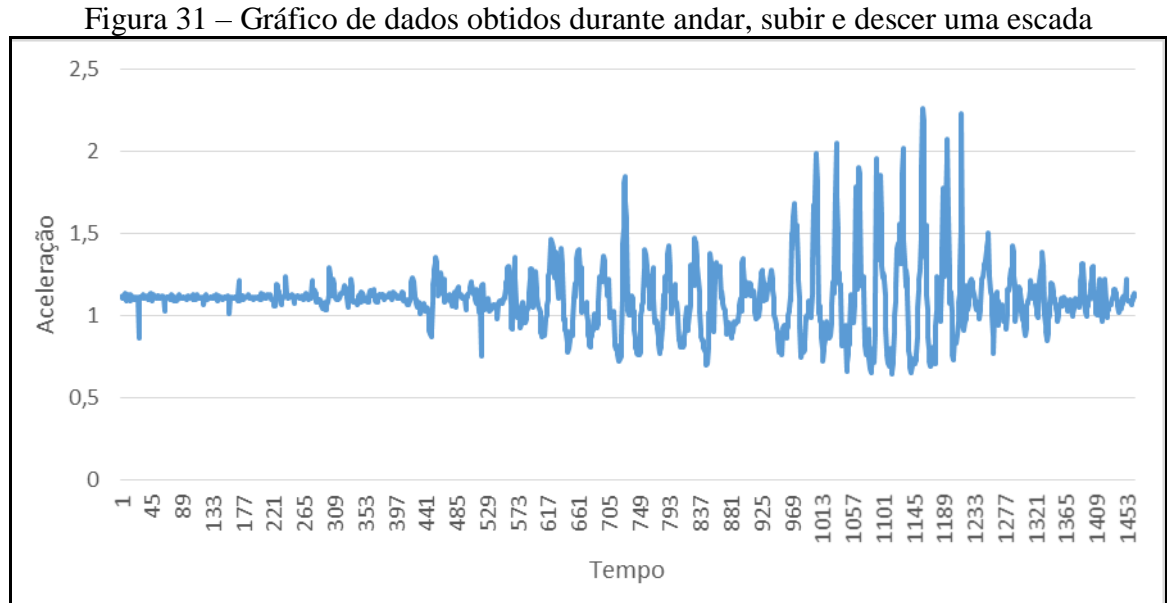


Fonte: Do autor.

Com o dispositivo novamente preso ao peito, outro teste realizado por esta pessoa foi de caminhar até uma escada, subir 10 degraus e em seguida descer os mesmos 10 degraus.



Apesar de descer a escada ter sido a atividade que gerou valores mais altos como mostra a Figura 31, estes ainda estavam longe do limite de 4, ou seja, não causou nenhum falso positivo.



Fonte: Do autor.

Não foi possível tratar os falsos positivos existentes, pois eles exigiriam outras verificações através de outros componentes, como por exemplo, um giroscópio, e não haviam mais entradas disponíveis na Espruino Pico e saídas de 3,3V. Isto acabaria exigindo a ampliação do protótipo do dispositivo a compartilhar a bateria e talvez até mesmo usar outra microcontroladora.

Em relação aos trabalhos correlatos, foram levantados os principais pontos a que se refere uma detecção de queda. O Quadro 8 mostra o levantamento e comparação dos pontos levantados.

Quadro 8 – Comparativo do trabalho presente e os correlatos

<b>Características</b>	<b>Trabalho presente</b>	<b>PIVA et al. (2014)</b>	<b>ROCHA, MARRANGHELLO (2014)</b>	<b>BOURKE, O'BRIEN, LYONS (2007)</b>
Analisa dados de pico do acelerômetro para queda	Sim	Sim	Sim	Sim
Aplica algum filtro sobre os dados	Sim	Não	Não	Sim
Realiza alguma validação adicional para evitar falsos positivos	Não	Sim	Sim	Sim
Funciona sem conexão à internet	Não	Sim	Sim	Sim
Monitoramento é feito em tempo real	Sim	Sim	Sim	Não

Fonte: Do autor.

A principal característica que se refere aos trabalhos é como a detecção de queda é realizada, e é possível notar que todos eles adotam o mesmo método de identificar o pico no acelerômetro durante a queda. O trabalho presente e o trabalho de Bourke, O'Brien, Lyons realizaram filtros sobre os dados para obter melhores resultados, enquanto o trabalho de Piva et al. não o realizou, confiando apenas as verificações extras identificando o ângulo em que o aparelho se encontra. O trabalho desta monografia não realizou nenhuma verificação adicional, o que compromete a confiabilidade, pois acusa falsos positivos, algo que o Arduino para aulas de Física também possui em comum, enquanto o trabalho de Bourke, O'Brien, Lyons teve outras análises realizadas em cima dos dados, como além de avaliar o pico máximo do momento da queda, também analisar o pico mínimo que ocorre durante a queda. O trabalho desta monografia gerou um problema que não é encontrado nos trabalhos correlatos, que é a dependência de uma conexão à internet, consequência de o desenvolvimento ser feito em uma aplicação cliente e servidor. Apesar de o trabalho e Bourke, O'Brien, Lyons se mostrar superior em todos os aspectos, o único ponto negativo é que os dados eram coletados e então depois processados para verificar a detecção de queda, onde que o trabalho apenas validava a viabilidade do método, algo que não ocorre nos outros trabalhos correlatos e no trabalho presente, onde a implementação é visada para o uso direto no usuário e com acompanhamento em tempo real.

Além disto, diferente dos outros trabalhos, o trabalho desta monografia também testou a viabilidade do uso do microcontrolador Espruino Pico. Ela demonstrou algumas limitações por exemplo na capacidade de componentes conectados a ela, visto que só existem duas saídas de 3.3V, mas ao mesmo tempo possui várias entradas e saídas analógicas, o que demonstra que ele deve ser utilizado em conjunto com componentes que tenham sua alimentação separada. Por outro lado, o desenvolvimento foi facilitado tendo em vista a disponibilidade da linguagem Javascript e da IDE exclusiva que realiza o *deploy* do código no microcontrolador através do USB.

## 4 CONCLUSÕES

Com a elaboração deste trabalho, foi construído um protótipo, com um dispositivo baseado no microcontrolador Espruino Pico, junto a um ESP8266, um acelerômetro ADXL335 e uma bateria, para monitoramento e uma aplicação servidora em C# tendo sido atingidos os objetivos principais do trabalho.

Uma dificuldade encontrada foi relativa ao método de detecção de queda, pois não foi localizado outro método que seja tão eficaz quanto simplesmente medir o pico da aceleração durante a queda. Então, utilizando o método mais comumente utilizado para esta finalidade, o maior desafio acaba sendo validar a eficácia, garantindo o menor número possível de falsos positivos para que a confiabilidade do dispositivo não seja comprometida. Isto se tornou em uma tarefa complexa, pois apenas com os dados do acelerômetro não foi identificado nenhum meio de diferenciar quedas reais de apenas colisões do próprio acelerômetro. Esta questão ainda permanece em aberto e deve ser objeto de estudos mais aprofundados.

A transmissão de dados via *socket* TCP também apresentou alguns desafios, pois como era necessário programar diretamente o envio de bytes para o *stream* do TCP, o tratamento para identificar o que são os bytes transmitidos no lado do servidor demanda processamento adicional. Para isto adotou-se uma serialização simples, onde o servidor e o cliente obedecem rigidamente ao padrão estabelecido.

Apesar destas dificuldades, o Espruino Pico demonstrou-se um microcontrolador adequado ao escopo do projeto a partir da disponibilidade da linguagem Javascript e em função da documentação disponível. O conjunto de recursos de hardware disponíveis possibilita o desenvolvimento de protótipos acadêmicos em função do tamanho reduzido.

Do ponto de vista da pesquisa acadêmica, a contribuição do presente projeto está relacionada aos aspectos que demandam aprofundamento na questão da minimização de falsos positivos, além do sentido de projetar uma interface de uso do sistema de forma simples de modo a facilitar a sua utilização por pessoas idosas visto que este é um público que demanda muito cuidado e atenção.

### 4.1 EXTENSÕES

Como futuras extensões sugere-se:

- a) implementar outros meios de notificação e alerta;
- b) aprimorar o filtro sobre os dados do acelerômetro para diminuir a margem de erro;
- c) usar outros canais de acesso à internet;
- d) desenvolver mecanismos de tratamento de erros de conexão do cliente com o

servidor;

- e) desenvolver um módulo de configuração de acesso à rede no dispositivo;
- f) ampliar o escopo dos testes com dados de situações reais.

## REFERÊNCIAS

- BOURKE, A.K.; O'BRIEN, J.V.; LYONS, G.M. Evaluation of a threshold-based tri-axial accelerometer fall detection algorithm. **Gait & Posture**, Limerick, v. 26, n. 2, 2007. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0966636206001895>>. Acesso em: 18 maio 2016.
- COUTINHO, Evandro da Silva Freire; SILVA, Sidney Dutra. Uso de medicamentos como fator de risco para fratura grave decorrente de queda em idosos. **Cad. Saúde Pública**, Rio de Janeiro, v. 18, n. 5, p. 1359-1366, 2002. Disponível em: <<http://www.scielo.br/pdf/%0D/csp/v18n5/11009.pdf>>. Acesso em: 18 maio 2016.
- DAMIÁN, J. et al. Factors associated with falls among older adults living in institutions. **BMC Geriatrics**, v. 13, n. 6, 2013. Disponível em: <<http://www.biomedcentral.com/content/pdf/1471-2318-13-6.pdf>>. Acesso em: 3 dez. 17.
- D'ORSI, Eleonora; XAVIER, André Junqueira; RAMOS, Luiz Roberto. Trabalho, suporte social e lazer protegem idosos da perda funcional: estudo epidioso. **Revista Saúde Pública**, v. 45, n. 6, p. 685-692, 2011. Disponível em <[http://www.scielo.br/scielo.php?script=sci\\_arttext&pid=S0034-89102011000400007&lng=pt&tlng=pt](http://www.scielo.br/scielo.php?script=sci_arttext&pid=S0034-89102011000400007&lng=pt&tlng=pt)>. Acesso em: 3 dez. 17.
- ESPRUINO. **Espruino**, Abingdon, 2017. Disponível em: <<http://www.espruino.com/>>. Acesso em: 8 ago. 2017.
- EXPRESSIF. **ESP8266 Overview**, Xangai, 2017. Disponível em: <<http://espressif.com/en/products/hardware/esp8266ex/overview>>. Acesso em: 16 dez. 2017.
- FREITAS, Mariana A. V; SCHEICHER, Marcos E. Preocupação de idosos em relação a quedas. **Revista Brasileira de Geriatria e Gerontologia**, Rio de Janeiro, v. 11, n. 1, 2008. Disponível em: <[http://revista.unati.uerj.br/scielo.php?script=sci\\_arttext&pid=S1809-98232008000100006&lng=pt&nrm=iso](http://revista.unati.uerj.br/scielo.php?script=sci_arttext&pid=S1809-98232008000100006&lng=pt&nrm=iso)>. Acesso em: 3 dez. 2017.
- GASPAROTTO, Livia Pimenta Renó; FALSORELLA, Glaucia Regina; COIMBRA, Arlete Maria Valente. As quedas no cenário da velhice: conceitos básicos e atualidades da pesquisa em saúde. **Revista Brasileira de Geriatria e Gerontologia**, v. 17, n. 1, p. 201-209. Rio de Janeiro, 2014. Disponível em: <<http://www.scielo.br/pdf/rbgg/v17n1/1809-9823-rbgg-17-01-00201.pdf>>. Acesso em: 3 dez. 2017.
- IEEE, Remote Control using Mobile trough Spoken Commands. **International Conference on Signal Processing, Communication and Networking**, Chennai, p. 622-625, 2007.
- LEE, Jer-Vui; CHUAH, Yea-Dat; CHIENG, Kenny T.H. Smart Elderly Home Monitoring System with an Android Phone. **International Journal of Smart Home**, v. 7, n. 3, 2013. Disponível em: <[http://www.sersc.org/journals/IJSH/vol7\\_no3\\_2013/2.pdf](http://www.sersc.org/journals/IJSH/vol7_no3_2013/2.pdf)>. Acesso em: 20 mar. 2016.
- LOPES, KT et al. Prevalência do medo de cair em uma população de idosos da comunidade e sua correlação com mobilidade, equilíbrio dinâmico, risco e histórico de quedas. **Revista brasileira de fisioterapia**, São Carlos, v. 13, n. 3, p. 223-229, Junho 2009. Disponível em: <[http://www.scielo.br/scielo.php?script=sci\\_arttext&pid=S1413-35552009000300006&lng=en&nrm=iso](http://www.scielo.br/scielo.php?script=sci_arttext&pid=S1413-35552009000300006&lng=en&nrm=iso)>. Acesso em: 3 dez. 2017.
- KICKSTARTER, **Espruino Pico: JavaScript on a USB Stick**, 2015. Disponível em: <<https://www.kickstarter.com/projects/gfw/espruino-pico-javascript-on-a-usb-stick>>. Acesso em: 7 maio 2016.

MENDES, Márcia R.S.S. Barbosa et al. A situação social do idoso no Brasil: uma breve consideração. **Acta paul. enferm.**, São Paulo, v. 18, n. 4, p. 422-426, 2005. Disponível em: <[http://www.scielo.br/scielo.php?script=sci\\_arttext&pid=S0103-21002005000400011&lng=en&nrm=iso&tlng=pt](http://www.scielo.br/scielo.php?script=sci_arttext&pid=S0103-21002005000400011&lng=en&nrm=iso&tlng=pt)>. Acesso em: 16 maio 2016.

NOURY, N. et al., Fall detection – Principles and Methods. **Conference of the IEEE EMBS**, Lyon, v. 29, p. 1663 – 1666, Aug. 2007. Disponível em: <<http://citeseerx.ist.psu.edu/viewdoc/download?rep=rep1&type=pdf&doi=10.1.1.212.8327>>. Acesso em: 2 de dez. 2017.

MICROSOFT DEVELOPER NETWORK, **MSDN – C#**, 2015. Disponível em: <<https://msdn.microsoft.com/pt-br/library/kx37x362.aspx>>, Acesso em: 31 mar. 2016.

MINERVINO, Diego Ramalho. **Desenvolvimento de Estruturas Integradas de Filtros e Antenas para Aplicações em Sistemas de Comunicações sem Fio**. 2013. 70 f. Dissertação (Mestrado em Engenharia Elétrica e Computação) – Curso de Mestrado de Engenharia Elétrica e Computação, Universidade Federal do Rio Grande do Norte, Natal.

PEREIRA, Alexandre Alves; CEOLIM, Maria Filomena; NERI, Anita Liberalesso. Associação entre sintomas de insônia, cochilo diurno e quedas em idosos da comunidade. **Cad. Saúde Pública**, Rio de Janeiro, v. 29, n. 3, p. 535-546, Mar. 2013. Disponível em: <<http://dx.doi.org/10.1590/S0102-311X2013000300011>>. Acesso em 3 dez. 2017.

PIVA, Leonardo Sabadini et al., fAlert: Um sistema android para monitoramento de quedas em pessoas com cuidados especiais. **Simpósio Brasileiro de Multimídia e Páginas Web**, João Pessoa, v. 20, p. 111-114, 2014. Disponível em: <<http://www.lbd.dcc.ufmg.br/colecoes/wfa/2014/004.pdf>>. Acesso em: 24 mar. 2016

PUR3 Limited. **Espruino Web IDE: help**. Version 0.67, 2017. Documento eletrônico disponibilizado com o ambiente Espruino Web IDE.

RIBEIRO, Adalgisa Peixoto et al. A influência das quedas na qualidade de vida de idosos. **Ciência & Saúde Coletiva**, Rio de Janeiro, v. 13, n. 4, 2006. Disponível em: <[http://saude.sp.gov.br/resources/ses/perfil/profissional-da-saude/grupo-tecnico-de-acoes-estrategicas-gtae/saude-da-pessoa-idosa/artigos-e-teses/a\\_influencia\\_das\\_quedas\\_na\\_qualidade\\_de\\_vida\\_de\\_idosos.pdf](http://saude.sp.gov.br/resources/ses/perfil/profissional-da-saude/grupo-tecnico-de-acoes-estrategicas-gtae/saude-da-pessoa-idosa/artigos-e-teses/a_influencia_das_quedas_na_qualidade_de_vida_de_idosos.pdf)>. Acesso em: 25 mar. 2016.

ROCHA, Fábio Saraiva; MARRANGHELLO, Guilherme Frederico. Acelerômetro eletrônico e a placa Arduino para ensino de Física em tempo real. **Caderno Brasileiro de Ensino de Física**, Florianópolis, v. 31, n. 1, 2014. Disponível em: <<https://periodicos.ufsc.br/index.php/fisica/article/view/2175-7941.2014v31n1p98/26427>>. Acesso em: 17 de maio de 2016.

ROLITA, Lydia et al. Greater Number of Narcotic Analgesic Prescriptions for Osteoarthritis Is Associated with Falls and Fractures in Elderly Adults. **Journal of the American Geriatrics society**, v. 61, n. 3, p. 335–340, 2013. Disponível em: <<http://onlinelibrary.wiley.com/doi/10.1111/jgs.12148/abstract;jsessionid=A6E9710198112BA1BFCC17634701C691.f02t01>>. Acesso em: 3 dez. 2017.

SANTANA, Carla da Silva et al. Uso de equipamentos de monitoramento da saúde por idosos no ambiente doméstico. **Revista brasileira de geriatria e gerontologia**, Rio de Janeiro, v. 17, n. 2, p. 383-393, 2014. Disponível em: <[http://www.scielo.br/scielo.php?script=sci\\_arttext&pid=S1809-98232014000200383&lng=en&nrm=iso](http://www.scielo.br/scielo.php?script=sci_arttext&pid=S1809-98232014000200383&lng=en&nrm=iso)>. Acesso em: 25 maio 2016.

SAÚDE BUSINESS, **Desafio Pfizer premia startups no HIS**, 2015. Disponível em: <<http://saudebusiness.com/noticias/desafio-pfizer-premia-startups-no-his/>>. Acesso em: 25 maio 2016.

SILVA NETO, Félix Ferreira. **Aplicando as Propriedades dos Vetores a Problemas da Geometria Clássica**. 2014. 82 f. Trabalho de Conclusão de Curso (Mestrado Profissional em Matemática) – Curso de Mestrado em Matemática, Universidade Federal da Paraíba, João Pessoa. Disponível em: <<http://tede.biblioteca.ufpb.br/bitstream/tede/7528/2/arquivototal.pdf>>. Acesso em: 18 dez. 2017.

SIQUIERA, Fernando V. et al., Prevalência de quedas em idosos e fatores associados. **Revista de Saúde Pública**, v. 41, n. 5, p. 749-756, 2007. Disponível em: <<https://dx.doi.org/10.1590/S0034-89102007000500009>>. Acesso em: 3 dez 2017.

SMITH, Steven W., **The Scientist and Engineer's Guide to Digital Signal Processing**, p. 277 – 284, 2011. Disponível em: <<http://www.dspguide.com/pdfbook.htm>>. Acesso em: 2 dez. 2017.

SPARKFUN ELECTRONICS. **WiFi Module ESP8266**, Niwot, 2017. Disponível em: <<https://www.sparkfun.com/products/13678>>. Acesso em: 16 dez. 2017.

VRIES, O. J. et al., Does frailty predict increased risk of falls and fractures? A prospective population-based study. **Osteoporosis International**, Londres, v. 24, n. 9, p. 2397-2403, 2013. Disponível em: <<https://link.springer.com/article/10.1007%2Fs00198-013-2303-z>>. Acesso em: 3 dez. 17.

WILLIAMS, Gordon, What version of JS does Espruno implement?. **Espruno Forums**, 2017. Disponível em: <<http://forum.espruno.com/conversations/309783/>> Acesso em: 17 dez. 2017.