



**Centro de Tecnologia
Curso de Engenharia de Computação**

Cristiana Soares de Vargas

**SISTEMA DE MONITORAMENTO REMOTO DE
QUEDAS DE PACIENTES**

Trabalho de Conclusão de Curso

Santa Maria, RS

2016



CRISTIANA SOARES DE VARGAS

SISTEMA DE MONITORAMENTO REMOTO DE QUEDAS DE PACIENTES

Trabalho de Conclusão de Curso apresentado ao Curso de Graduação em Engenharia de Computação da Universidade Federal de Santa Maria (RS) como requisito parcial para obtenção do grau de Engenheira de Computação.

Orientador: Prof. Dr. José Eduardo Baggio

Santa Maria, RS

2016



CURSO DE ENGENHARIA DE COMPUTAÇÃO

A Banca Examinadora, abaixo assinada, aprova o Trabalho de Conclusão de Curso

SISTEMA DE MONITORAMENTO REMOTO DE QUEDAS DE PACIENTES

Elaborado por

Cristiana Soares de Vargas

Como requisito parcial para obtenção do grau de Engenheira de Computação.

Banca Examinadora:

Prof. Dr. José Eduardo Baggio – UFSM

Prof. Dr. Carlos Henrique Barriuello – UFSM

Prof. Dr. Cesar Augusto Prior – UFSM

Santa Maria, RS, Brasil.

2016

AGRADECIMENTOS

Agradeço primeiramente a Deus por permitir o cumprimento dessa etapa da minha vida.

Aos meus pais, Claudeci e Beatriz, que foram fundamentais para o meu desenvolvimento até aqui. À minha mãe, principalmente pelo amor e paciência que demonstrou comigo ao longo do curso. Agradeço pelo carinho incondicional nas horas difíceis, de desânimo e cansaço. Ao meu pai que, mesmo de longe, esteve presente em cada etapa dessa trajetória e me deu todo o apoio necessário.

À minha família por sempre acreditarem em mim.

Um agradecimento especial às minhas amigas Laís, Josiane, Thaís, Patrícia e Danielle que sempre estiveram ao meu lado, me apoiaram, me aconselharam e me ajudaram a colocar as coisas sob a perspectiva correta.

Aos meus colegas, principalmente Sandra Cossul e Vinícius Azzolin, pelos trabalhos que realizamos juntos, pelas risadas, pelo apoio e pelos conhecimentos trocados. Com certeza tornaram essa caminhada muito mais agradável.

Ao meu orientador, José Eduardo Baggio, por ter aceitado esse convite. Agradeço por ter confiado em mim, por sempre estar disposto a esclarecer as minhas dúvidas e por todo o conhecimento transmitido.

À Universidade Federal de Santa Maria, ao curso de Engenharia da Computação e ao corpo docente, pela oportunidade de realizar esse curso e por me proporcionarem tanto conhecimento.

"Precisamos dar um sentido humano às nossas construções. E quando o amor ao dinheiro, ao sucesso, nos estiver deixando cegos, saibamos fazer pausas para olhar os lírios do campo e as aves do céu." (Érico Veríssimo)

RESUMO

SISTEMA DE MONITORAMENTO REMOTO DE QUEDAS DE PACIENTES

Autora: Cristiana Soares de Vargas

Orientador: José Eduardo Baggio

O presente trabalho apresenta um sistema de monitoramento de pacientes, principalmente de pessoas idosas, utilizado para detectar quedas em tempo real e, assim, evitar longas esperas por auxílio. Esse é um serviço extremamente importante, considerando que um rápido atendimento ao paciente contribui para a sua recuperação e diminuição de sequelas. Pessoas de todas as idades apresentam risco de sofrerem quedas, porém, para os idosos, elas possuem um significado muito relevante, pois podem levá-los à incapacidade, injúria e morte. O sistema conta com, basicamente, quatro módulos: um acelerômetro, um módulo WiFi, um Arduino e um dispositivo Android. O dispositivo, que deve se manter preso na altura do tórax do usuário, é capaz de se comunicar a um smartphone via Wifi. O objetivo dessa conexão é manter o status do paciente atualizado no aplicativo para Android e, caso o paciente sofra uma queda, o smartphone envia automaticamente uma mensagem de SMS para o cuidador ou responsável pelo paciente. Assim, é proporcionado um rápido atendimento ao idoso evitando danos maiores recorrentes da demora em receber socorro. Além disso, o dispositivo conta com um pequeno botão e uma luz vermelha. Quando ocorre uma queda, a luz vermelha se acende e só é desligada quando o paciente receber ajuda e o botão for, então, pressionado.

Palavras-chave: Monitoramento de Pacientes. Android. Queda de Pacientes.

ABSTRACT

REMOTE MONITORING SYSTEM OF PATIENTS FALL

Author: Cristiana Soares de Vargas

Advisor: José Eduardo Baggio

This project introduces a patient monitoring system directed mainly to elderly people, used in order to detect falls in real-time and avoid long waits for help. This is an extremely important service considering that when the patient is readily aided that contributes to his recovery and decreases the chances to have a permanent sequel. People of all ages are in risk when it comes to falls, however for the elderly it can have a more relevant meaning since it can cause them disabilities, serious injuries or even the death. The system has basically four modules: an accelerometer, a WiFi module, an Arduino and an Android device. The developed device, that must be located in the patient's chest area, is capable of communicating to a smartphone via WiFi. The main goal of this connexion is to keep the patient status updated at the Android application, and if a fall occurs the smartphone will automatically send a SMS text to the patient caregiver or someone responsible for him. In this way, a fast aid is provided to the elder avoiding damages resulting of a long wait for help. Furthermore, the device has got a small pushbutton and a red light. When an event of fall occurs the red light will turn on and it will be turned off only when the patient receives help and the button is pressed.

Keywords: Patients Monitoring. Android. Patients Fall.

LISTA DE FIGURAS

Figura 1- Ilustração do Sistema de Monitoramento Remoto baseado em sensores portáteis.	13
Figura 2- Acelerômetro - ADXL345.....	19
Figura 3- Módulo EP8266, modelo ESP-01.....	20
Figura 4- Diagrama da ligação de pinos entre ESP8266 e FTDI.	22
Figura 5- Arduino Nano	22
Figura 6- Modo Designer, App Inventor.....	23
Figura 7- Editor de Blocos, App Inventor.....	24
Figura 8- Código QR gerado para o download do arquivo apk.	24
Figura 9- Tela do aplicativo MIT App Inventor 2 Companion	25
Figura 10- Diagrama de blocos do Sistema de Monitor de Pacientes.....	25
Figura 11- Circuito inicial construído em cima de uma protoboard.	25
Figura 12- Esquemático do projeto criado utilizando o Eagle.....	26
Figura 13- Layout elaborado a partir do esquemático do Eagle.	27
Figura 14- Posicionamento do acelerômetro levando em consideração seus eixos x, y e z.	28
Figura 15- Fluxograma do software Arduino.	29
Figura 16- Fluxograma do aplicativo Android.	31
Figura 17- Mensagem impressa ao não detectar nenhum IP cadastrado.	33
Figura 18- Novo campo para inserção de IP.....	33
Figura 19- Tela do aplicativo quando o paciente se encontra estável.	34
Figura 20- Tela do aplicativo quando ocorre uma queda.	34
Figura 21- Aviso de mensagem enviada após o sinal de SOS ser detectado.	35

LISTA DE TABELAS

Tabela 1-Conexão entre o módulo ESP8266 e o conversor USB/Serial.....	21
------------------------------------------------------------------------------	-----------

LISTA DE ABREVIATURAS

TA	Tecnologia Assitiva
IoT	Internet das Coisas (Inglês: <i>Internet of Things</i>)
SMS	<i>Short Message Service</i>
ADC	Conversor Analógico-Digital (Inglês: <i>Analog-to-digital Converter</i>)
SPI	<i>Serial Peripheral Interface</i>
I2C	<i>Inter Integrated Circuit</i>

SUMÁRIO

1. INTRODUÇÃO.....	12
2. REVISÃO BIBLIOGRÁFICA	15
3. PROJETO DO MONITOR DE PACIENTES	19
3.1. Módulos Componentes.....	19
3.1.1. Acelerômetro ADXL345	19
3.1.2. Módulo WiFi.....	20
3.1.3. Arduino.....	22
3.1.4. MIT App Inventor	23
3.2. Monitor de Pacientes.....	25
3.2.1. Hardware	25
3.2.2. Software Arduino	27
3.2.3. Software Android.....	30
4. RESULTADOS EXPERIMENTAIS.....	33
CONCLUSÃO.....	36
BIBLIOGRAFIA	37
APÊNDICE A- CÓDIGO ARDUINO.....	39

INTRODUÇÃO

O crescimento da população de idosos, em números absolutos e relativos, é um fenômeno mundial e está ocorrendo a um nível sem precedentes. Em 1950, eram cerca de 204 milhões de idosos no mundo e, já em 1998, quase cinco décadas depois, este contingente alcançava 579 milhões de pessoas, um crescimento de quase oito milhões de pessoas idosas por ano. As projeções indicam que, em 2050, a população idosa será de 1.900 milhões de pessoas, montante equivalente à população infantil de 0 a 14 anos de idade (IBGE, 2002).

Já no Brasil, os idosos - pessoas com mais de 60 anos – em 2011, somavam 23,5 milhões, mais que o dobro do registrado em 1991, quando a faixa etária contabilizava 10,7 milhões de pessoas (Humanos, 2012). Segundo dados da Organização Mundial da Saúde, o número de idosos no Brasil quase triplicará até 2050.

Apesar desse crescente número de idosos, muitos deles em condições frágeis, ou seja, apresentam modificações determinadas por afecções que tornam a pessoa idosa debilitada e dependente do auxílio de terceiros, acabam por serem relegados a casas de repouso ou hospitais por não terem condições de viverem sem um cuidador. Uma das razões para isso é que tem sido verificado nos serviços de emergência dos EUA, por exemplo, que as quedas são eventos frequentes causadores de lesões, constituindo a principal etiologia de morte acidental em pessoas com idade acima de 65 anos (Fabrício, et al., 2004).

Além disso, quedas em hospitais são apontadas como responsáveis por dois em cada cinco eventos indesejáveis relacionados à segurança do paciente. A *National Patient Safety Agency* estima que milhares de pacientes lidam com fraturas decorrentes de quedas todos os anos, sendo que alguns morrem em resultado da queda (Patient Safety Observatory, 2007). As lesões decorrentes de quedas ocorrem entre 15% a 50% dos eventos, resultando em grande variedade de danos, como síndrome pós-queda, aumento da comorbidade e comprometimento da recuperação, aumento do tempo de hospitalização e dos custos assistenciais, ansiedade da equipe assistencial e perda da confiança na instituição e processos legais (Correa, et al., 2012).

Tendo em vista esse cenário, a Tecnologia Assistiva (TA) vem para promover a reabilitação ou inclusão de pessoas portadoras de alguma deficiência ou com sua mobilidade de alguma forma reduzida. Pode ser definida como uma ampla gama de equipamentos, serviços, estratégias e práticas concebidas e aplicadas para minorar os problemas encontrados pelos indivíduos com deficiências (Cook & Polgar, 1995).

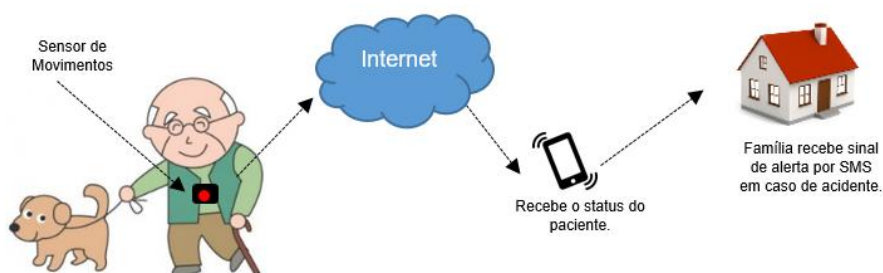
Assim, visando a máxima diminuição das lesões causadas por quedas, é necessário um rápido atendimento ao paciente, esteja ele em sua própria casa, em uma clínica de repouso ou hospital. Para tornar possível esse atendimento, são necessários dispositivos de TA para que cuidadores e familiares possam monitorar qualquer evento de queda.

Diante desse contexto, foi desenvolvido um sistema de monitoramento remoto em tempo real de pacientes com o objetivo de possibilitar o rápido atendimento, em caso de quedas, por parte de um profissional habilitado. É um sistema não invasivo, seu manuseio não requer nenhum método complexo e não há contraindicação. Além disso, conta com a vantagem de não utilizar um número excessivo de caixas, fios e aparelhos aparentes, tornando, assim, o monitoramento da forma mais natural e confortável possível.

O aparelho desenvolvido deve ficar posicionado preferencialmente no peito do paciente de maneira firme, a fim de que não haja interferências. Além disso, o aparelho conta com um LED vermelho e um botão, sendo que quando o LED se acende significa que a ajuda foi requisitada. O botão deve ser utilizado pelo profissional ou familiar encarregado dos cuidados do paciente. Assim, ao atender o paciente, o botão deve ser pressionado para que a luz vermelha se apague e o *status* do aplicativo retorne para “OK”.

O sistema ainda conta com um aplicativo Android utilizado para receber o *status* do paciente e capaz de enviar um SMS (*Short Message Service*) de alerta a um cuidador ou familiar com o número já cadastrado. As informações trocadas entre o aparelho monitor e o smartphone utilizam troca de dados por Wifi. Portanto, aplicativo e hardware devem estar conectados à rede de internet. Na Figura 1 é possível ter uma ideia do funcionamento geral do dispositivo de monitoramento.

Figura 1- Ilustração do Sistema de Monitoramento Remoto baseado em sensores portáteis.



Inicialmente o trabalho apresenta uma revisão bibliográfica com as principais abordagens do tema, bem como pesquisas realizadas no campo de monitoramento de movimentos e compartilhamento de dados. Após isso, serão considerados os principais módulos utilizados na execução e montagem do projeto e as principais ferramentas. Em

seguida, são apresentados o desenho do hardware do projeto e do software, sendo logo em seguida explicados o funcionamento do sistema como um todo. Por fim, os resultados obtidos com os testes do sistema e a conclusão encerram o conteúdo escrito desse trabalho.

1 REVISÃO BIBLIOGRÁFICA

A ideia de utilizar sensores de movimentos não é novidade, e uma grande quantidade de artigos e trabalhos acadêmicos tem abordado o assunto. Além disso, a miniaturização dos sensores tem despertado o interesse de pesquisadores em desenvolver sistemas de monitoramento portáteis ou vestíveis.

Apesar da grande variedade de sensores de movimento disponíveis, os seguintes quatro sensores apresentados são os mais utilizados na detecção de movimentos em dispositivos eletrônicos: acelerômetros, giroscópios, sensores magnéticos e sensores de pressão.

Acelerômetros medem a aceleração linear e o ângulo de inclinação, detectam a magnitude e a direção combinadas da aceleração linear e gravitacional, podendo, assim, serem usados para detectar posições e quedas.

Giroscópios medem a taxa de rotação angular sobre um ou mais eixos. Diferentemente de acelerômetros e bússolas digitais (e-Compasses), os giroscópios não dependem de nenhuma força externa, como gravidade ou campos magnéticos, podendo, assim, funcionar de forma autônoma.

Sensores magnéticos são usados para detectar uma posição com base no campo magnético da Terra. Suas aplicações comuns incluem a orientação correta em mapas disponíveis na tela do *smartphone* e fornecer informações de localização para aplicações de navegação.

Por fim, os sensores de pressão medem a altitude relativa e absoluta utilizando uma análise da mudança na pressão atmosférica. Esses sensores têm sido amplamente utilizados nos dispositivos para análises em esportes e atividades físicas e em aplicativos baseados na localização.

Apesar dessa variedade de sensores de movimento, tem-se notado a utilização de acelerômetros na maioria dos estudos envolvendo monitoramento remoto de pacientes, incluindo a pesquisa de (Barbosa, et al., 2013), que mostrou um sistema multicanal de telemetria. Esse estudo teve como objetivo ajudar a realizar a supervisão regular da saúde. Uma placa eletrônica foi desenvolvida com o objetivo de captar, pré-processar, armazenar e enviar os sinais para um repositório central via Bluetooth. A validação de aplicação de detecção de quedas foi feita com testes do acelerômetro triaxial, para detectar a queda de um indivíduo e analisar os valores limiares para detecção da queda em três diferentes situações.

Essa aplicação mostrou-se auxiliar para o monitoramento de queda de idosos e na prevenção de doenças não transmissíveis.

Na pesquisa de (Choudhury, et al., 2008), foi demonstrada a capacidade de reconhecimento das atividades do usuário utilizando um sistema vestível baseado em acelerômetros, microfones, sensores de luz e assim por diante. Ao final da pesquisa, o sistema foi capaz de identificar as atividades mais comuns de cada usuário, como cozinhar, caminhar ou conversar, utilizando algoritmos probabilísticos. Entretanto, para a captação dos dados, foi necessário que se usasse um cabo USB, pois a utilização do protocolo Bluetooth não se mostrou suficientemente confiável para a transmissão contínua dos dados. Dessa maneira, se tornou necessário que o usuário carregasse uma combinação de dispositivos volumosos e com fios apenas para coletar os dados.

Em (Bao & Intille, 2004) foram utilizados cinco acelerômetros biaxiais posicionados em diferentes partes do corpo com o intuito de detectar em tempo real a atividade que está sendo realizada pelo paciente, como caminhar, sentar, assistir TV, correr, andar de bicicleta, comer e ler. Os dados do acelerômetro foram capturados a partir de 20 voluntários sem a supervisão ou observação dos pesquisadores. Os voluntários deveriam executar suas tarefas diárias sem nenhuma orientação de como ou onde realizá-las. Após alguns testes, o melhor desempenho que o dispositivo apresentou foi de 84% de precisão ao classificar as atividades de cada voluntário.

O sistema desenvolvido por (Ravi, et al., 2005) se baseou em um acelerômetro triaxial, localizado na região pélvica, para identificar as atividades diárias do usuário. Os dados do acelerômetro foram transmitidos via Bluetooth para um iPAQ da HP e então convertidos para ASCII utilizando um script Python. Ao final do trabalho, chegou-se a conclusão de que as atividades podem ser reconhecidas com uma boa precisão utilizando-se apenas um acelerômetro triaxial. Porém, atividades limitadas ao movimento das mãos e braços (como escovar os dentes) se tornam mais complexas de se identificar.

Já os estudos de (Miranda & Marik, 2008), descrevem um sistema completo que tem como objetivo comunicar a queda de um paciente idoso aos seus familiares utilizando o envio de SMS para um telefone pré-cadastrado. O sistema é composto por dois blocos, o primeiro constituído por dois acelerômetros e o segundo é uma central móvel, que receberá os sinais em caso de queda do paciente e enviará o SMS. Esses sinais de queda são recebidos via transmissão RF, que utiliza modulação em amplitude ou, simplesmente AM (Amplitude

Modulada). É a forma de modulação em que a amplitude de um sinal senoidal, varia em função do sinal de interesse, que é o sinal modulador.

Pôde-se notar que grande parte dos estudos realizados no campo de tecnologia assistiva, como os trabalhos anteriormente comentados, tem utilizado o protocolo Bluetooth para a transmissão dos dados. Porém, algumas dificuldades relacionadas a esse protocolo foram notadas e algumas alternativas foram propostas. Para esse trabalho, entretanto, foi utilizado o protocolo WiFi, que possibilita a conexão de diversos dispositivos em uma rede de internet. Essa nova fase da internet, que tem possibilitado conectar milhares de objetos do nosso dia a dia e processar essas informações para a nossa facilidade, é chamada de Internet das Coisas ou IoT (Inglês: *Internet of Things*).

Segundo (Haller, et al., 2009), IoT é um mundo onde objetos físicos estão integrados a uma rede de informação e, onde esses objetos podem se tornar participantes ativos em processos de negócio. Em toda a internet são oferecidos serviços para a interação com esses ‘objetos inteligentes’, obter seu status e qualquer informação associada a eles, levando em conta questões como privacidade e segurança. Dessa maneira, pode-se dizer que IoT serve para cobrir o vão entre os objetos no mundo físico e a suas representações em sistemas de informação.

Além disso, a IoT tem o potencial de rastrear pessoas, equipamentos, suprimentos, ou até mesmo animais de serviços e, também, analisar os dados capturados. É com isso em mente que muitas pesquisas tem sido realizadas utilizando a IoT na monitorização de pacientes.

Quando a IoT é utilizada em dispositivos médicos e aplicações voltadas para a saúde, ela tem sido chamada de Internet das Coisas Médicas (IoMT, do Inglês: *Internet of Medical Things*). Alguns exemplos de aplicações da IoMT são consideradas a seguir.

Para um paciente que sofre de bulimia (distúrbio de alimentação) em um hospital ou residência, os sensores no ambiente do paciente podem detectar o aumento da temperatura do corpo ou pressão sanguínea, ou até mesmo o odor de vômito.

Sensores poderiam, ainda, serem utilizados para detectar o abuso de exercícios físicos tal como um treinamento cardiológico excessivo ou uma atividade de caminhada acelerada quando comparada a uma caminhada em ritmo normal. Os dados obtidos dessas atividades podem proporcionar informações de valor no diagnóstico e gerenciamento de doenças (Prado, 2016).

Além disso, a área de cuidados para idosos é muito oportuna para a utilização da tecnologia de IoT. Recentemente, na Feira Internacional de Tecnologia em Las Vegas, foi

anunciado pela Consumer Technology Association (CTA) um projeto em colaboração com a IBM que vai pesquisar como a computação cognitiva pode fornecer melhores informações para ajudar a transformar a vida das populações cada vez mais longevas do mundo e das pessoas com deficiência (Consumer Technology Association, 2016).

Pesquisadores e profissionais da área de saúde visualizam que muitas outras aplicações de IoT na área de saúde poderiam substancialmente melhorar o cuidado do paciente, otimizar a utilização de recursos, e economizar quantidades de dinheiro – se apenas os sistemas fossem implementados (Prado, 2016).

2 PROJETO DO MONITOR DE PACIENTES

A seguir são apresentados os principais materiais utilizados e conceitos abordados para o desenvolvimento desse trabalho. São apresentadas algumas das ferramentas utilizadas na implementação do sistema.

2.1 Módulos Componentes

A fim de se obter o funcionamento do projeto, foram necessários pelo menos quatro módulos principais: um acelerômetro, um módulo WiFi, um Arduino, utilizado para processar e reunir todas as informações e, por último, o aplicativo Android, utilizado para visualizar os dados captados pelo Arduino. A partir de agora, cada um desses módulos será explorado, provendo uma visão geral do seu funcionamento.

2.1.1 Acelerômetro ADXL345

O detector de quedas do sistema é baseado nos dados de um acelerômetro, que são computados por um Arduino Nano. O acelerômetro escolhido é o ADXL345, mostrado na Figura 22, produzido pela SparkFun. É um acelerômetro triaxial com alta resolução (13 bits) de medida de até $\pm 16g$, baixo consumo de potência e oferece um Conversor Analógico-Digital (ADC) integrado. As informações medidas podem ser disponibilizadas através das interfaces de comunicação I2C ou SPI em palavras de 16-bits a complemento de dois. Quando configurado para interface I2C a comunicação ocorre nas taxas de transferência de 100kHz ou 400kHz, casos em que a máxima taxa de saída de dados é de 200Hz e 800Hz, respectivamente. No caso desse projeto, foi utilizado o protocolo de comunicação I2C.

Figura 2- Acelerômetro - ADXL345



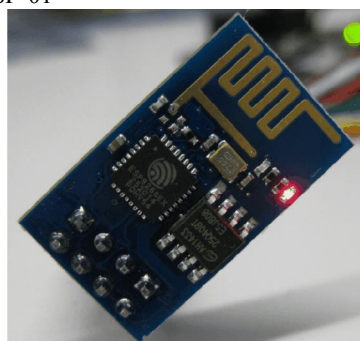
Esse sensor mede a aceleração estática da gravidade em aplicações utilizando o sensor de inclinação, bem como aceleração dinâmica resultante de movimento ou choque. Além disso, sua alta resolução permite a detecção de mudanças de inclinação menores de 1.0° .

Algumas funções sensoriais são disponibilizadas pelo ADXL345 como sensor de atividade e inatividade que detectam a presença ou falta de movimento e se a aceleração em algum dos eixos excedeu o limite estabelecido pelo usuário. Além disso, conta com sensores capazes de detectar um ou dois toques e queda-livre. Essas funções podem ser mapeadas para a saída de um dos dois pinos de interrupção disponíveis.

2.1.2 Módulo WiFi

A fim de estabelecer uma conexão entre o sensor de movimentos foi utilizado um módulo WiFi de baixo custo. O ESP8266, que pode ser visto na Figura 33, é um módulo fabricado pela Espressif e tem sido amplamente utilizado para explorar a crescente onda de Internet das Coisas.

Figura 3- Módulo EP8266, modelo ESP-01



O que tem tornado o módulo particularmente atrativo são, principalmente, dois aspectos. O primeiro é a faixa de preço, podendo chegar a cinco dólares ou menos. E o segundo, é o seu tamanho reduzido. Além disso, é um módulo fácil de integrar a outras soluções, bastando o uso de uma comunicação serial UART.

O módulo Wifi ESP8266 suporta o protocolo TCP/IP integrado e consegue oferecer a qualquer microcontrolador (Arduino, PIC etc) acesso à sua rede Wifi. O ESP8266 é capaz tanto de hospedar uma aplicação quanto descarregar todas as funções de redes Wifi a partir de outro processador de aplicação. Algumas informações técnicas são listadas abaixo:

- Ligação e transmissão de pacotes em menos de 2ms;

- Consumo em standby: menos de 1,0mW (DTIM3);
- CPU 32 bits de baixo consumo integrada – dispensa microcontroladores;
- Tensão de operação: 3,3V;
- Alcance: 50m, aproximadamente;
- Conectores: GPIO, I2C, SPI, UART, Entrada ADC, Saída PWM e Sensor de

Temperatura interno;

- Dimensões: 25 x 14 x 2mm;
- Peso: 7g.

O ESP-01 vem com um firmware atualizável, porém, a sua fabricante disponibiliza no github, serviço web para hospedar arquivos, o firmware de comandos AT. Os comandos AT são a forma básica de configurar e acionar o ESP8266 quando ele está sob o controle de um dispositivo externo, como o Arduino. Esses comandos são descendentes diretos do chamado, Padrão de Hayes, de 1981, utilizado para permitir que os computadores pessoais interagissem com conexões telefônicas controlando diretamente um modem (Anon., 2016). Assim, para esse projeto, o firmware do ESP-01 foi atualizado para os comandos AT.

Para a atualização do firmware foi necessária a utilização de um conversor USB/Serial (FTDI FT232RL) para tornar possível a comunicação entre o módulo ESP8266 e o PC. Essa comunicação é feita utilizando-se o pino RX do ESP conectado ao pino TX do conversor e o pino TX do ESP conectado ao pino RX do conversor. Na Tabela 1 é possível conferir a correta ligação entre os pinos do ESP e do conversor serial. Caso o conversor serial tenha como única opção o fornecimento de uma tensão de 5V, então é necessário que se utilize divisores de tensão no pinos VCC, CH_PD e RX do ESP, para que não seja causado nenhum dano ao módulo.

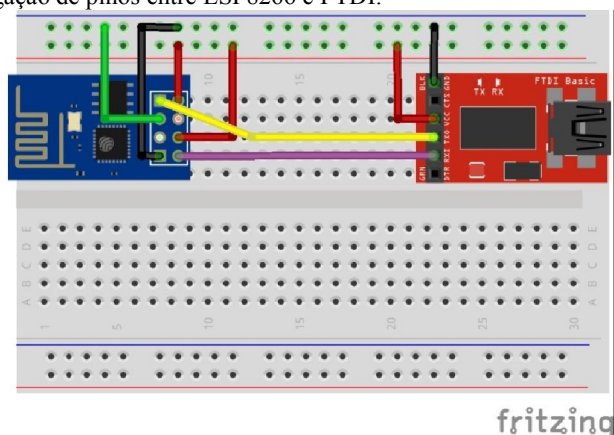
Tabela 1-Conexão entre o módulo ESP8266 e o conversor USB/Serial.

ESP8266	FTDI FT232RL
VCC	3.3V
CH_PD	3.3V
GND	GND
TX	RX
RX	TX
GPIO0	GND

O ESP entra no modo flash (onde é permitido modificar o firmware) ao conectar o pino GPIO0 ao *ground*. Para que essa tarefa seja executada corretamente, é necessário ligar o pino GPIO0 ao GND e logo após reiniciar o módulo. Ao final do processo de atualização, é preciso retirar o GPIO0 do GND e reiniciar novamente o módulo ESP.

Na Figura 4 é possível visualizar o diagrama com a ligação entre os pinos do ESP8266 e o módulo conversor serial/USB.

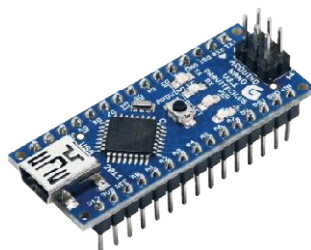
Figura 4- Diagrama da ligação de pinos entre ESP8266 e FTDI.



2.1.3 Arduino

A versão do Arduino utilizada nesse projeto é o Arduino Nano, pois é um módulo pequeno, completo e totalmente adaptável à *protoboard*. Foi projetado e produzido pela Gravitech, contudo sua fabricação foi descontinuada. Na Figura 5 é possível conferir o modelo do Arduino utilizado.

Figura 5- Arduino Nano



O Nano conta com um microcontrolador desenvolvido pela Atmel, o ATmega328, que possui 32KB de memória, 2KB de SRAM e 1KB de EEPROM. Possui 14 pinos digitais que podem ser usados como entrada ou saída, desses foram utilizados três pinos. Além disso, possui oito pinos de entrada analógicos, dos quais foram utilizados dois.

A programação da aplicação foi feita utilizando o ambiente de desenvolvimento interativo para Arduino (IDE), que é uma aplicação multiplataforma escrita em Java. Inclui um editor de código com recursos de realce de sintaxe e biblioteca Wiring para programar em C/C++ (Barbosa, et al., 2013).

2.1.4 MIT App Inventor

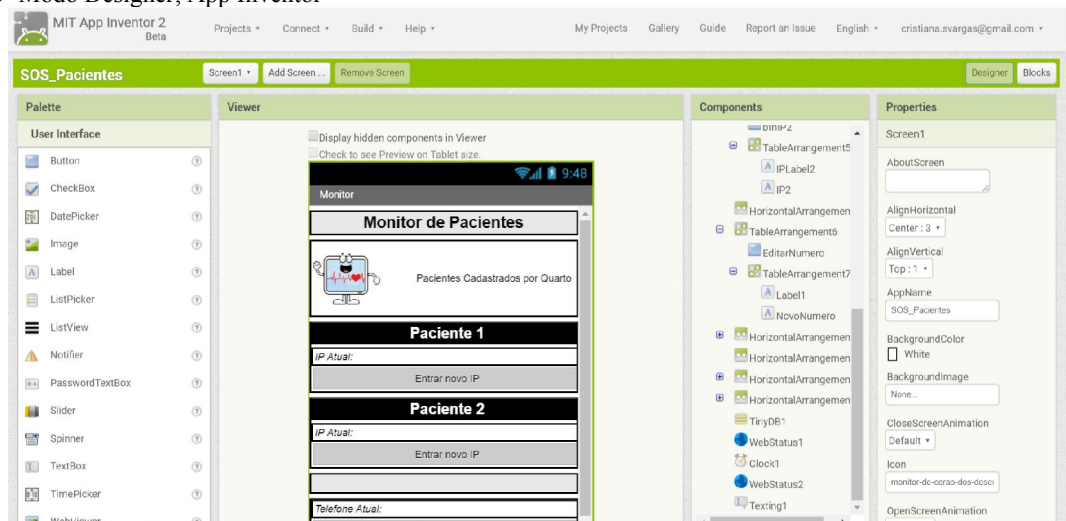
O aplicativo de monitoramento de pacientes, chamado *SOS_Pacientes*, foi criado para permitir ao cuidador, ou enfermeiro, monitorar a condição do paciente em tempo real. A fim de desenvolver esse aplicativo foi utilizada a plataforma de desenvolvimento online App Inventor.

O *MIT App Inventor* é uma plataforma inovadora de introdução a programação e criação de aplicativos, ela transforma a complexa linguagem de programação baseada em texto, em blocos de construção visual (Team, 2012-2015). Com essa ferramenta, até mesmo pessoas sem experiência em programação podem criar aplicativos para celulares Android.

A interface gráfica do *MIT App Inventor* permite ao usuário usar o mouse para arrastar e soltar os componentes do aplicativo, comandos e funções. Os componentes são posicionados diretamente na tela e os comandos e funções são formados por uma linguagem visual que lembra peças de um quebra-cabeça.

A plataforma consiste em dois modos principais: o *Designer* e o *Blocks*. No modo *Designer*, como é possível visualizar na Figura 6, é possível criar a interface do usuário do aplicativo. Pode-se escolher os componentes necessários para a aplicação e editar as suas propriedades.

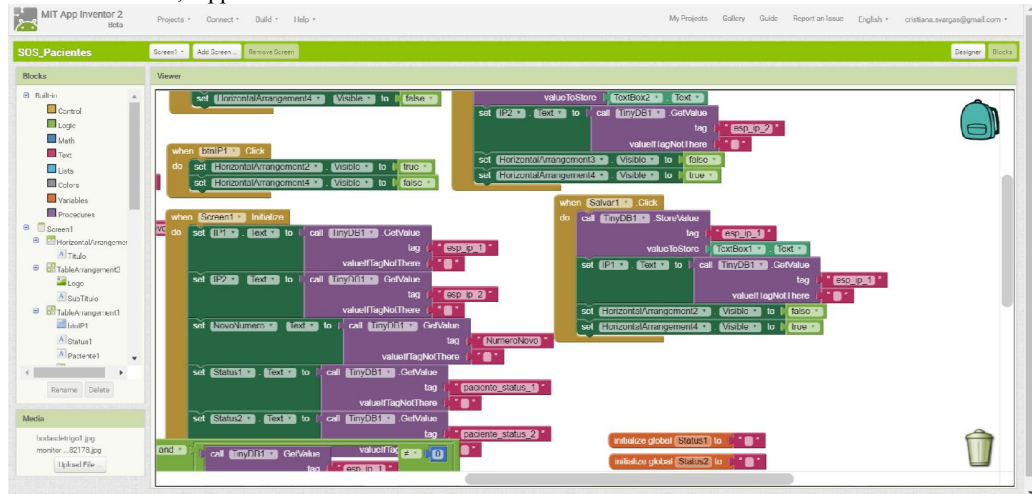
Figura 6- Modo Designer, App Inventor



No editor de blocos, na aba *Blocks*, ilustrada na Figura 77, é possível programar o comportamento do aplicativo por unir os blocos. São disponibilizados blocos para determinar

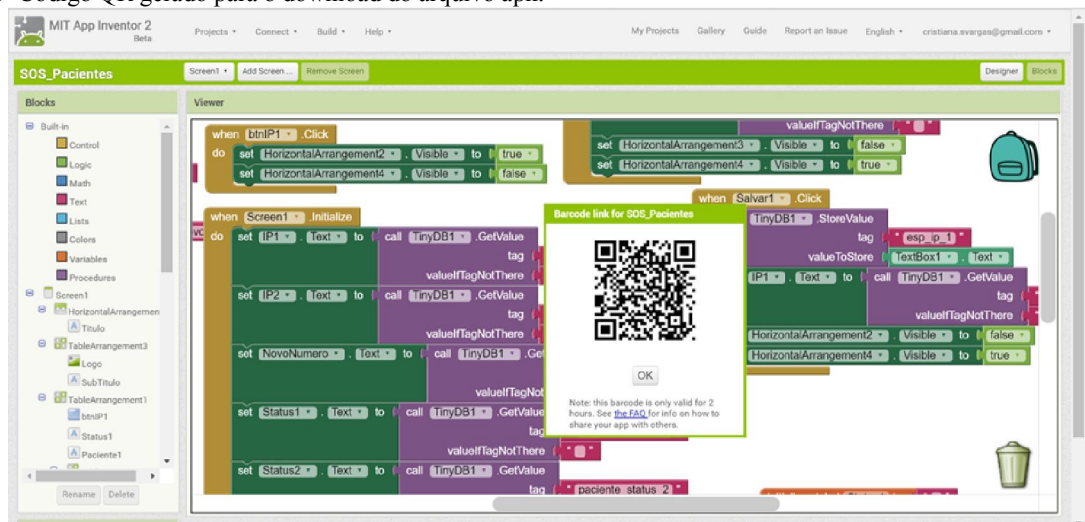
variáveis, blocos de controle, lógica, textos, etc. Além disso, é no modo de blocos que se determinam os eventos a se realizarem ao clique de um botão.

Figura 7- Editor de Blocos, App Inventor



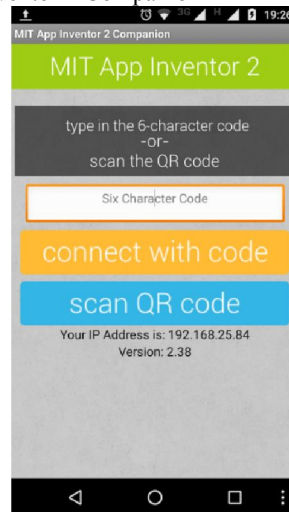
Após o aplicativo ser finalizado é possível solicitar o código QR do aplicativo e baixar o arquivo *.apk* para a instalação no smartphone. O código QR é gerado conforme ilustra a Figura 8.

Figura 8- Código QR gerado para o download do arquivo apk.



Outra maneira de baixar o aplicativo, é manter o smartphone sincronizado à plataforma, assim, todas as alterações feitas no App Inventor serão automaticamente instaladas no smartphone. Para isso, é necessário baixar o aplicativo *MIT App Inventor 2 Companion* na Play Store. Assim, a tela mostrada na Figura 99 ficará visível no smartphone, e, então, após clicar em AI Companion na plataforma de desenvolvimento, um código será gerado. É possível digitar o código ou escanear o código QR.

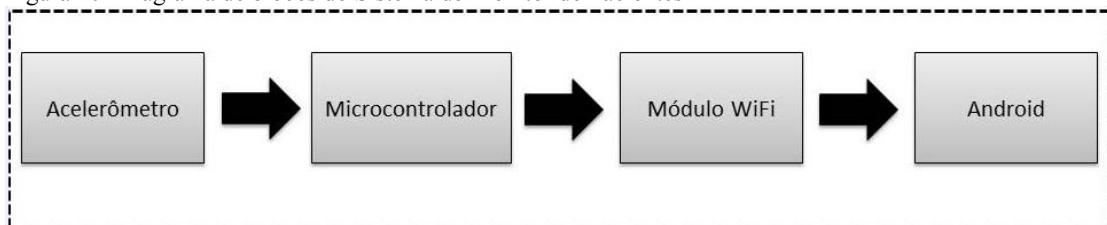
Figura 9- Tela do aplicativo MIT App Inventor 2 Companion



2.2 Monitor de Pacientes

O sistema Monitor de Pacientes pode ser ilustrado segundo o diagrama de blocos na Figura 10. Ali é possível perceber a presença de um acelerômetro, do Arduino, de um módulo Wifi e de um dispositivo Android.

Figura 10- Diagrama de blocos do Sistema de Monitor de Pacientes

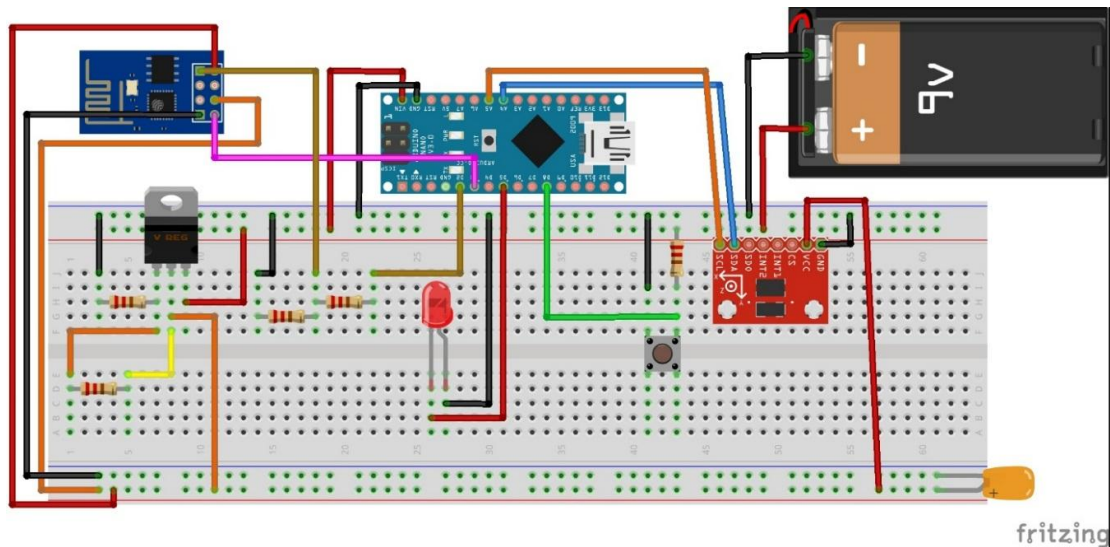


A seguir são detalhados os métodos utilizados para o desenvolvimento desse sistema. Inicialmente trata-se do desenvolvimento do Hardware, onde se tem a conexão dos módulos previamente explicados; logo após, apresenta-se o software desenvolvido para o Arduino e, por fim o software implementado para o sistema operacional Android.

2.2.1 Hardware

Inicialmente, a fim de testar o funcionamento geral do projeto, foi construído em uma *protoboard* o circuito mostrado na Figura 111.

Figura 11- Circuito inicial construído em cima de uma protoboard.

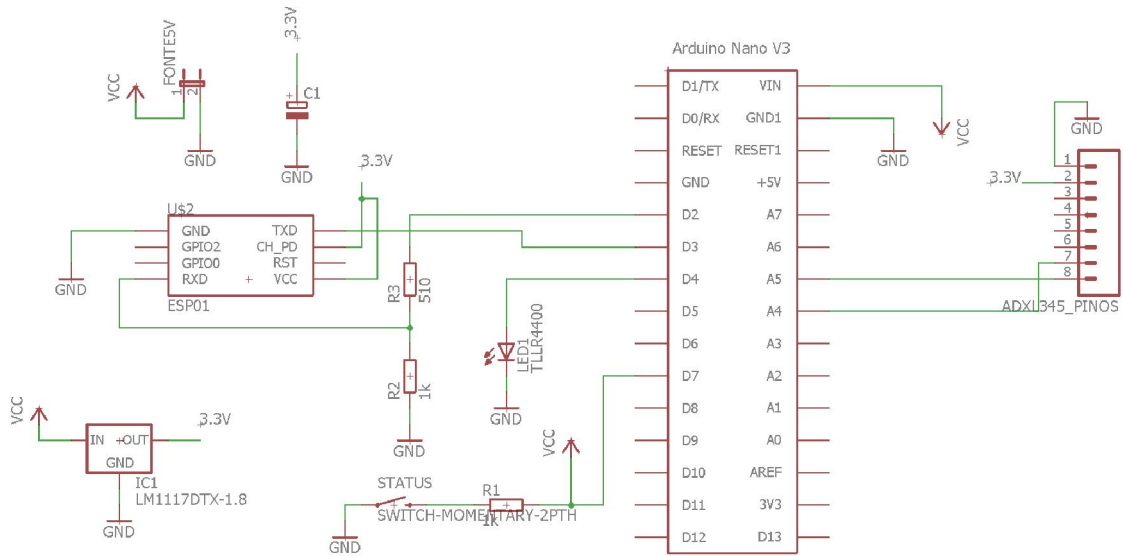


É importante salientar que os valores dos resistores são diferentes, mas devido a limitações do software de desenho, eles foram representados da mesma cor. Além disso, o Arduino está sendo alimentado através do pino Vin, por uma bateria de 9V. Após verificado o correto funcionamento do circuito por meio de testes na protoboard, foi elaborado o esquemático do circuito, utilizando a ferramenta de desenho de placas de circuito, Eagle. Na Figura 12 é possível visualizar o circuito e, assim, conferir os valores corretos de cada resistor.

Além dos módulos principais já mencionados, foi utilizado no circuito um regulador de tensão (LM1117) que converte a tensão de entrada para uma tensão de saída de 3.3V. Esse controle é necessário, pois tanto o módulo WiFi quanto o acelerômetro operam em uma tensão de 3.3V, já o Arduino opera entre 7V e 12V, por isso a alimentação externa escolhida foi de 9V.

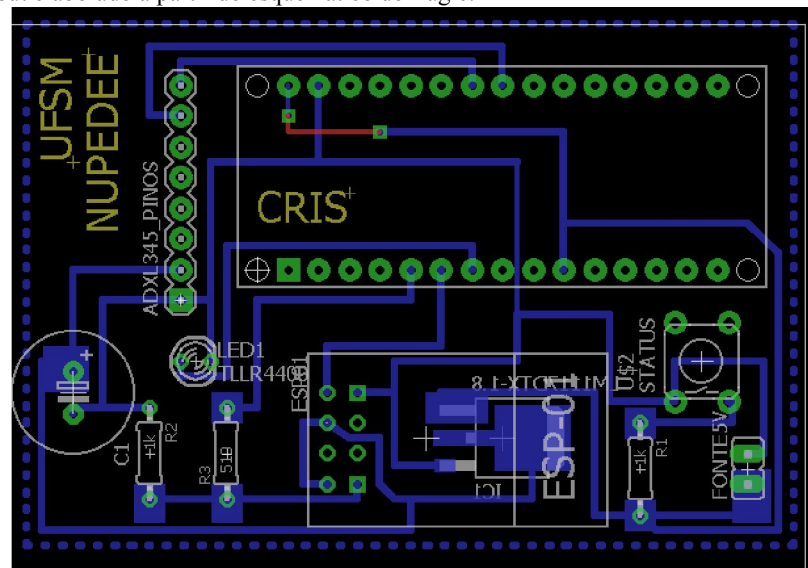
Para o correto funcionamento do módulo WiFi, foi necessário um divisor de tensão entre os pinos TX do Arduino e RX do ESP8266, para impedir que o sinal transmitido pelo Arduino não ultrapasse 3.3V.

Figura 12- Esquemático do projeto criado utilizando o Eagle.



Após a criação do esquemático, agora utilizando a ferramenta de layout do software Eagle, foi elaborado o desenho da placa final, para, assim, ser enviado para o Nupedee da UFSM onde é feita a prototipagem das placas. Na Figura 13 é representado o layout final da placa.

Figura 13- Layout elaborado a partir do esquemático do Eagle.



A pilha utilizada para alimentação foi anexada posteriormente utilizando um adaptador para bateria com dois fios.

2.2.2 Software Arduino

O software projetado para operar no Arduino consiste em, basicamente, três tarefas principais: inicializar os módulos, verificar a ocorrência de queda do paciente e, caso tenha ocorrido a queda, verificar se o atendimento já foi prestado.

A primeira etapa da execução inicializa o módulo Wifi utilizando os comandos AT para verificar a rede escolhida e a senha da mesma. Além disso, responde com uma mensagem de aviso no caso de não conseguir detectar o módulo ligado.

Para conectar o módulo Wifi, ESP8266, ao Arduino, foi utilizada a biblioteca *softwareSerial*. Essa biblioteca é fácil de encontrar, pois vem com a IDE do Arduino e não precisa ser instalada à parte. Ela é necessária para que o programa possa ser “debugado” no monitor serial, pois o Arduino Nano possui apenas uma porta serial nativa.

Na segunda etapa, acontece um laço de repetição enquanto não for detectada a queda do paciente. Esse laço consiste na captação dos dados do acelerômetro e, logo após, a verificação da queda.

Para calcular o momento exato que ocorre o evento de queda no paciente, utilizaram-se conceitos de média aritmética e variância, que é uma medida de dispersão. Para esse cálculo foram utilizados os valores captados no eixo x do acelerômetro. Esse acelerômetro deve ficar posicionado preferencialmente na região do tórax do paciente a fim de não interferir em suas atividades cotidianas e movimentos. Na Figura 14 é possível visualizar o posicionamento do acelerômetro em relação ao corpo do paciente.

Figura 14: Posicionamento do acelerômetro levando em consideração seus eixos x, y e z.



Ao iniciar a captação, um vetor de 50 posições é completado, esse valor foi fixado levando em conta o tempo para manter o Arduino atualizado. A captação dos dados pelo Arduino ocorre em uma frequência padrão de 909,09 Hz e o acelerômetro está com seu intervalo de leitura definido em $\pm 2g$. A partir daí, calcula-se a média aritmética desse vetor, após isso, aplica-se a fórmula da variância, descrita abaixo:

$$s^2 = \sum_{i=1}^n \frac{(x_i - \bar{x})^2}{(n - 1)}$$

Onde \bar{x} é a média aritmética.

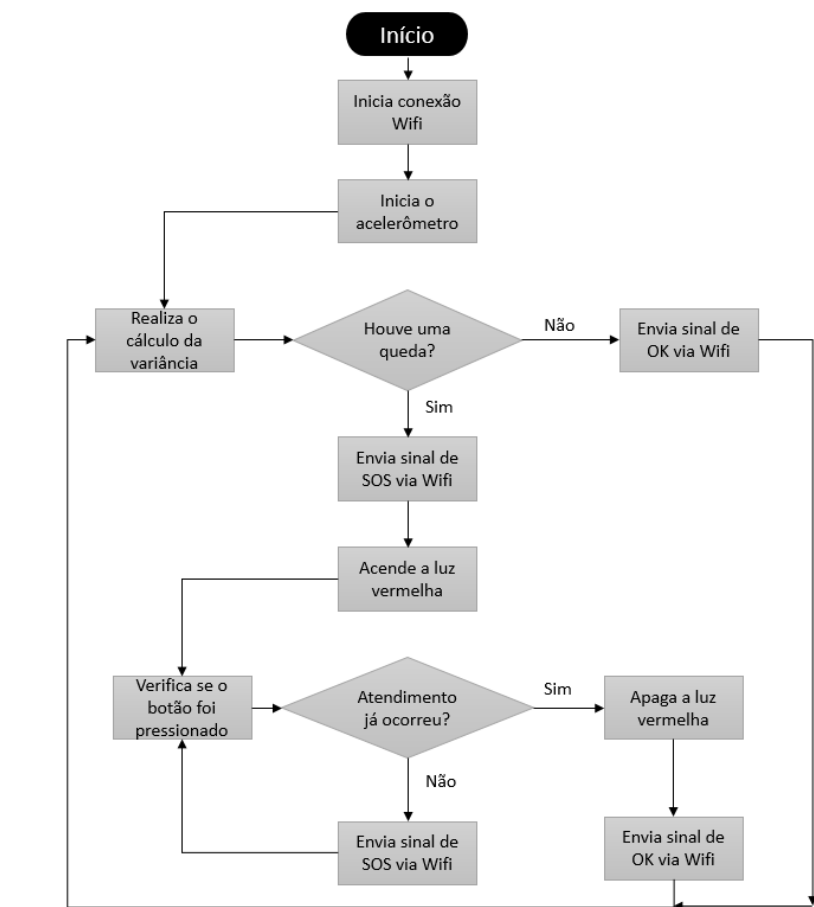
A variância é a soma dos quadrados dividida pelo número de observações do conjunto menos um. A partir da variância é possível verificar a medida de dispersão que mostra o quão distante os valores estão da média. Para verificar que resultado de variância indicaria uma queda foram feitos testes com o sensor simulando um paciente caindo de uma cadeira e um paciente sofrendo uma queda a partir de sua cama. A partir desses testes, chegou-se no valor de aproximadamente 1000 para a variância. Esse valor leva em conta que em pacientes idosos os movimentos são mais lentos causando, assim, um valor de variância menor para atividades diárias quando comparado a um adulto saudável.

Ainda dentro desse laço, é enviada uma mensagem para o módulo Wifi de “OK”, representando a condição do paciente. Caso o resultado da variância ultrapasse 1000, automaticamente a queda é detectada e a próxima etapa é executada.

Já na terceira etapa, após a execução sair do laço anterior, ela entra em um novo laço de repetição, agora verificando se houve o atendimento ao paciente. Assim, enquanto a luz vermelha no dispositivo estiver acesa, o microcontrolador estará enviando um sinal de “SOS” ao módulo Wifi. Além disso, o programa se mantém verificando se o botão, referente ao atendimento do paciente, foi pressionado. Caso o responsável pelo paciente tenha efetuado o atendimento, ele deve pressionar o botão e verificar se a luz vermelha foi apagada. Quando isso acontecer, o microcontrolador enviará automaticamente uma mensagem “OK” para o módulo Wifi e reiniciará a segunda etapa.

A fim de possibilitar uma melhor visualização do funcionamento do software, o fluxograma na Figura 15, pode ser usado como referência.

Figura 15- Fluxograma do software Arduino.



Além disso, durante toda a execução do programa do Arduino, a conexão com a rede WiFi é verificada, caso ocorra uma queda de rede ele tenta automaticamente se conectar novamente.

No Apêndice A é possível verificar o código completo para o Arduino.

2.2.3 Software Android

O aplicativo desenvolvido para o sistema operacional Android foi desenvolvido para servir como principal ferramenta para enviar as informações relativas ao paciente para a sua família ou cuidador. É composto por uma única tela onde é possível cadastrar até dois IPs e um número de celular.

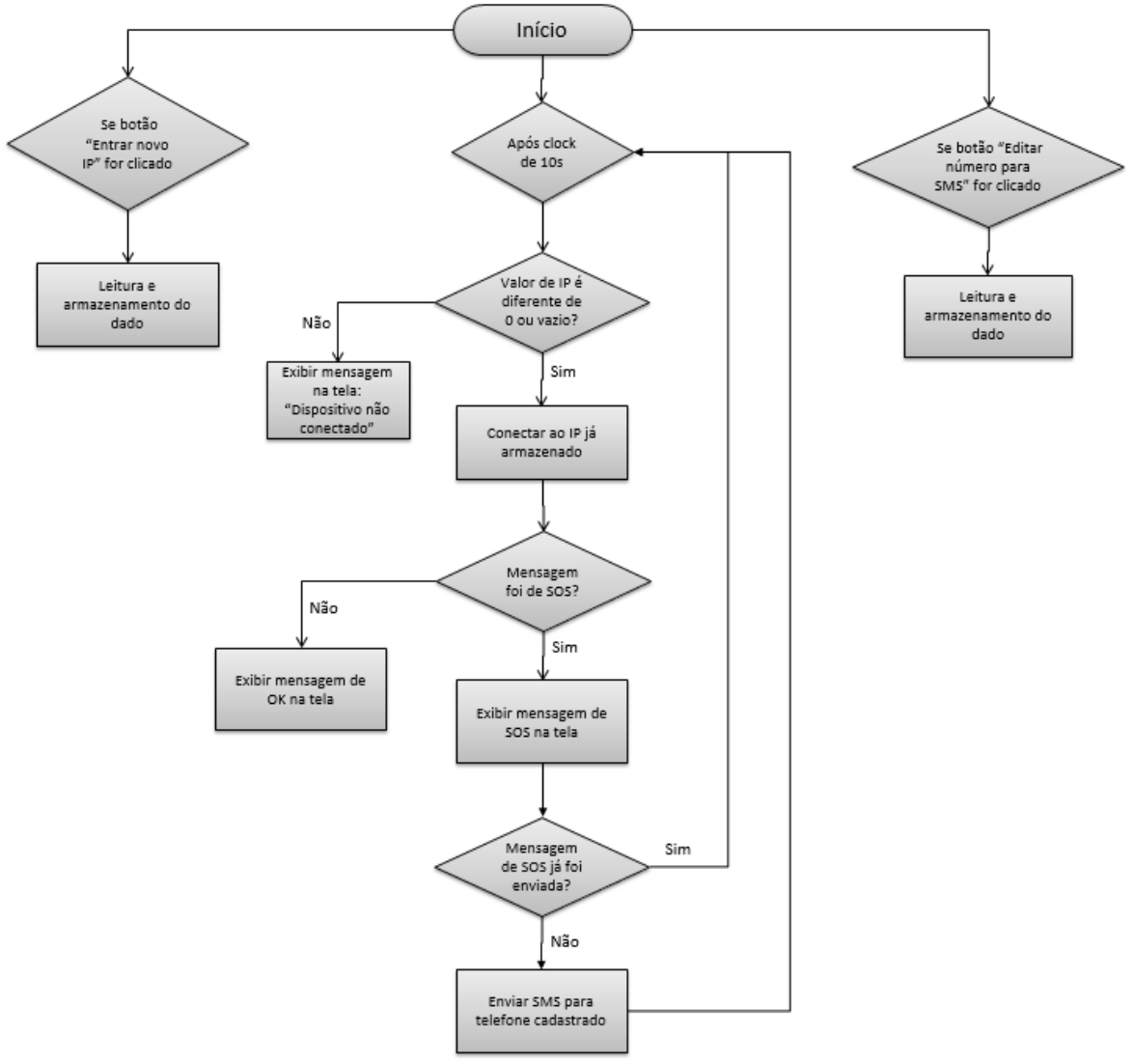
Após o número do IP ser cadastrado, ele é automaticamente armazenado no dispositivo, para que sempre que for inicializado ele “lembre” dos dados previamente cadastrados. As variáveis utilizadas no *App Inventor* são armazenadas na memória RAM do smartphone ou tablet e seu conteúdo é excluído quando o aplicativo é fechado. Porém, existe a

opção de armazenar um dado no *TinyDB* para que quando o aplicativo for fechado ou o dispositivo desligado, o valor armazenado ainda esteja disponível. O *TinyDB*, pode ser usado como ferramenta para armazenar dados persistentes, ou seja, que permanecem no *smartphone* até que seja feita uma nova instalação do aplicativo ou uma atualização do dado. O mesmo ocorre com o número de celular armazenado para o recebimento do SMS.

O aplicativo tem como principal função um *timer* que, a cada dez segundos, verifica o status do paciente. Ao ocorrer a interrupção do *timer*, haverá a tentativa de se estabelecer uma conexão via Wifi com o IP já cadastrado. Caso o campo dedicado ao IP esteja vazio ou o valor seja zero, será exibida a mensagem de “Dispositivo não conectado” no campo de status. Se a conexão for estabelecida com sucesso, então será exibida a mensagem com o status atual do paciente.

No caso de ser detectada uma mensagem de “SOS”, o aplicativo enviará automaticamente, quando houver um número de celular previamente cadastrado, uma mensagem de SMS para o responsável ou familiar do paciente. Essa mensagem é enviada apenas uma vez a cada queda. Um fluxograma mostrando o funcionamento do aplicativo pode ser verificado na Figura 16.

Figura 16- Fluxograma do aplicativo Android.



3 RESULTADOS EXPERIMENTAIS

A primeira atividade do aplicativo é verificar se há algum IP já previamente cadastrado, lembrando que uma vez cadastrado um IP, ele ficará gravado mesmo que o aplicativo seja fechado. Caso nenhum IP tenha sido salvo, a mensagem, conforme a Figura 17, será impressa.

Figura 17- Mensagem impressa ao não detectar nenhum IP cadastrado.



O cadastro de um novo IP, acontece de maneira simples clicando-se no botão “Entrar novo IP”, e logo depois de inserido o número, é possível salvá-lo. Na Figura 18 é possível observar o campo para inserção do novo IP sendo disponibilizado.

Figura 18- Novo campo para inserção de IP



Após a entrada de um IP válido, o aplicativo começará a enviar o status atual do paciente. Enquanto o paciente estiver estável, ou seja, nenhuma queda houver acontecido, o monitor continuará exibindo uma mensagem de OK, conforme é possível observar na Figura 19.

Figura 19- Tela do aplicativo quando o paciente se encontra estável.



No caso de uma queda ocorrer, o aplicativo receberá uma mensagem de SOS e em seguida exibirá na tela, conforme Figura 20.

Figura 20- Tela do aplicativo quando ocorre uma queda.



Além disso, na Figura 21 é possível visualizar o momento em que um SMS é enviado para o familiar ou cuidador do paciente. Dessa forma, é possível providenciar o rápido atendimento para o paciente, evitando sequelas maiores.

Figura 21- Aviso de mensagem enviada após o sinal de SOS ser detectado.



O sistema se mostrou eficiente em detectar a queda do paciente nos testes em que a autora simula diferentes tipos de queda utilizando o aparelho de monitoramento preso ao corpo. Alguns dos testes incluíram simular quedas a partir de poltronas, sofás e camas. A conexão do módulo ESP8266 se mostrou instável algumas vezes, mas de forma geral, funcionou corretamente.

CONCLUSÃO

A IoT em conjunto com a TA, vem mudando a maneira como realizamos nossas atividades cotidianas. Isso inclui atividades como a prática de esportes, alimentação, cuidados da saúde e assim por diante. Porém, uma das grandes contribuições dessas novas tecnologias é a melhora da qualidade de vida de pessoas idosas e com limitações físicas.

Esse trabalho apresentou um sistema cuja finalidade principal é monitorar pacientes com suas atividades motoras debilitadas ou que corram algum risco de saúde ao morarem sozinhos. O sistema desenvolvido não necessita de um número excessivo de fios e caixas e nem de um alto nível de conhecimento da tecnologia para manuseá-lo. Dessa maneira, se obtém como resultado um dispositivo intuitivo e amigável, tornando mais fácil a aceitação pelo seu público alvo.

O sistema cumpriu todas as tarefas requisitadas, como enviar o status do paciente via WiFi e enviar um SMS ao responsável pelo paciente. É um sistema que não possui um custo elevado para aquisição, pois todos os módulos e componentes utilizados tem como característica principal seu baixo-custo. Tendo em vista que atualmente grande parte da população possui um *smartphone*, o custo inicial de se aderir ao sistema acaba se tornando relativamente baixo.

O sistema proposto pode ser usado como referência para futuros projetos e pode ser aprimorado em alguns aspectos. Por exemplo, em relação à precisão na detecção de queda, é possível acrescentar os outros dois eixos do acelerômetro ao sensor. Ainda é possível utilizar um algoritmo de aprendizagem de máquina para disponibilizar as tarefas comumente realizadas pelo usuário, bem como tornar conhecidas as tarefas que mais causam acidentes e sugerir maneiras de realizá-las com segurança.

Adicionalmente, é possível melhorar o estabelecimento da conexão do dispositivo monitor à rede WiFi disponibilizando-se uma interface para que o próprio usuário possa executar a conexão. Isso dispensaria a necessidade do programador alterar a configuração diretamente no código do Arduino.

BIBLIOGRAFIA

ANON., 2016. History of Computers, Hardware, Software, Internet.... Disponível em: <http://history-computer.com/ModernComputer/Basis/modem.html>. Acesso em: 10 Agosto 2016.

BAO, Ling; INTILLE, Stephen S. Activity Recognition from User-Annotated, Cambridge: Massachusetts Institute of Technology, 2004.

BARBOSA, J. V; LOPEZ, L. R. R; GOROSO, D. G. Aplicação de sistema multicanal na detecção de quedas para idosos. Revista Brasileira de Inovação Tecnológica em Saúde, 3(2), pp. 38-47, 2013.

CHOUDHURY, T. et al. The mobile sensing platform : An embedded activity recognition system. IEEE Pervasive Computing, 7(2), pp. 32-41, 2008.

CONSUMER TECHNOLOGY ASSOCIATION, 2016. CES. Disponível em: <https://www.ces.tech/News/Press-Releases/CES-Press-Release.aspx?NodeID=a21fb12e-9ee3-4417-a612-fb7dbd995e10>. Acesso em: 16 Novembro 2016.

COOK, A. M; POLGAR, J. M. Cook & Hussey's Assistive Technologies: Principles and Practice. s.l.:Mosby Elsevier, 1995.

CORREA, A. D. et al. Implantação de um protocolo para gerenciamento de quedas em hospital: resultados de quatro anos de seguimento. Revista da Escola de Enfermagem da USP, 13 Abril, 1(46), pp. 67-74, 2012.

FABRÍCIO, S. C. C; RODRIGUES, R. A. P; COSTA, M. L. d. Causas e conseqüências de quedas de idosos atendidos em hospital público. Revista Saúde Pública, 1(38), pp. 93-99, 2004.

HALLER, S; KARNOUSKOS, S; SCHROTH, C. The Internet of Things in an Enterprise Context, s.l.: s.n, 2009.

HUMANOS, S. d. D., 2012. Secretaria Especial de Direitos Humanos. Disponível em: <http://www.sdh.gov.br/assuntos/pessoaidosa/dadosestatisticos/DadossobreoenvelhecimentoonoBrasil.pdf>. Acesso em 14 Setembro 2016.

IBGE, 2002. Perfil dos Idosos Responsáveis pelos Domicílios. Disponível em: <http://www.ibge.gov.br/home/presidencia/noticias/25072002pidoso.shtm>. Acesso em: 05 Outubro 2016.

MIRANDA, A. d; MARIK, P. Sistema Detector de Quedas (SDQ) , Curitiba: Pontifícia Universidade Católica do Paraná - PUCPR, 2008.

PATEL, S. et al. A review of wearable sensors and systems with application in rehabilitation. Journal of NeuroEngineering and Rehabilitation, 2012.

PATIENT SAFETY OBSERVATORY. Slips, trips and falls in hospital, England: National Patient Safety Agency (NPSA), 2007.

PRADO, E., 2016. Convergência Digital. Disponível em: http://sis-publique.convergenciadigital.com.br/cgi/cgilua.exe/sys/start.htm?from_info_index=41&info_id=43313&sid=15. Acesso em: 05 Outubro 2016.

RAVI, N; DANDEKAR, N; MYSORE, P; LITTMAN, M. L. Activity Recognition from Accelerometer Data. IAAI'05 Proceedings of the 17th conference on Innovative applications of artificial intelligence, Volume 3, pp. 1541-1546, 2005.

TEAM, T. A. I., 2012-2015. MIT App Inventor. Disponível em: <http://appinventor.mit.edu/explore/about-us.html>. Acesso em: Junho 2016.

APÊNDICE A- CÓDIGO ARDUINO

```

#include <SoftwareSerial.h>
#include <Wire.h>
#define ACC (0xA7>>1)
#define A_TO_READ (6)
#define SSID "ssID"
#define PASS "Password"
#define debug true
SoftwareSerial ESP8266(3,2);

void initAcc(){
  writeTo(ACC, 0x2D, 1<<3);
  writeTo(ACC, 0x31, 0x0B);
  writeTo(ACC, 0x2C, 0x09);
}

void getAccelerometerData(int * result){
  int regAddress = 0x32;
  byte buff[A_TO_READ];
  readFrom(ACC, regAddress, A_TO_READ, buff);
  result[0] = (((int)buff[1]) << 8) | buff[0];
  result[1] = (((int)buff[3])<< 8) | buff[2];
  result[2] = (((int)buff[5]) << 8) | buff[4];
}

int inPin = 8;
int val = 0;
int RedPin = 4;
float x[50];
int index = 0;
int total = 0;
int average = 0;
float var_x = 0;
String fall="";

void setup(){
  Serial.begin(230400);
  ESP8266.begin(115200);
  Wire.begin();
  initAcc();
  pinMode(inPin, INPUT);
  pinMode(RedPin, OUTPUT);
  digitalWrite(RedPin, LOW);
  connectWiFi();
  for (int i = 0; i < 50; i++)
    x[i] = 0;
  delay(500);
  Serial.println("Pronto para receber comandos");
}

```

```

void loop(){
  while(fall==""){
    int hx,hy,hz;
    int acc[3];
    getAccelerometerData(acc);
    hx = acc[0];
    hy = acc[1];
    hz = acc[2];
    Serial.print(" X=");
    Serial.print(hx);
    Serial.print(" Y=");
    Serial.print(hy);
    Serial.print(" Z=");
    Serial.println(hz);
    if(hx<0){
      hx=hx*(-1);
    }
    total -= x[index];
    x[index] = hx;
    total += x[index];
    index = (index + 1);
    if (index >= 50)
      index = 0;
    average = total / 50;
    for(int j=0; j<50; j++){
      var_x = var_x + (pow((x[j]-average),2))/49;
    }
    Serial.println(var_x);
    if(var_x>=1000){
      fall = "SOS";
      Serial.print(fall);
      Serial.print(" ");
      Serial.print(var_x);
      digitalWrite(RedPin, HIGH);
    }
    var_x=0;
    if (ESP8266.available()){
      if (ESP8266.find("+IPD,")){
        delay(300);
        int connectionId = ESP8266.read() - 48;
        String content = "OK";
        sendHTTPResponse(connectionId, content);
      }
    }
  }
  if (ESP8266.available()){
    if (ESP8266.find("+IPD,")){
      delay(300);
      int connectionId = ESP8266.read() - 48;

```



```

        String content = fall;
        sendHTTPResponse(connectionId, content);
    }
}
val = digitalRead(inPin);
if (val == LOW) {
    var_x = 0;
    fall = "";
    index = 0;
    total = 0;
    average = 0;
    for (int i = 0; i < 50; i++)
        x[i] = 0;
    digitalWrite(RedPin, LOW);
}
}

void writeTo(int DEVICE, byte address, byte val) {
    Wire.beginTransmission(DEVICE);
    Wire.write(address);
    Wire.write(val);
    Wire.endTransmission();
}

void readFrom(int DEVICE, byte address, int num, byte buff[]) {
    Wire.beginTransmission(DEVICE);
    Wire.write(address);
    Wire.endTransmission();
    Wire.beginTransmission(DEVICE);
    Wire.requestFrom(DEVICE, num);
    int i = 0;
    while(Wire.available()){
        buff[i] = Wire.read();
        i++;
    }
    Wire.endTransmission();
}

boolean connectWiFi(){
    ESP8266.println("AT");
    delay(1000);
    if (ESP8266.find("OK")){
        ComandoAT("AT+RST", 2000);
        ComandoAT("AT+CWMODE=1", 1000);
        String cmd = "AT+CWLAP=\"";
        cmd += SSID;
        cmd += "\",\"";
        cmd += PASS;
        cmd += "\"";
        ComandoAT(cmd, 8000);
    }
}

```

```

    ComandoAT("AT+CIFSR", 1000);
    ComandoAT("AT+CIPMUX=1", 1000);
    ComandoAT("AT+CIPSERVER=1,80", 8000);
    Serial.println("LIGACAO CONCLUIDA COM SUCESSO.");
    return true;
}
else{
    ComandoAT("AT+CWMODE=1", 1000);
    ComandoAT("AT+RST", 2000);
    Serial.println("\n!!!! ESP DESLIGADO !!!!");
    delay(3000);
    connectWiFi();
    return false;
}
}

String ComandoAT(String comando, const int timeout){
    String resposta = "";
    ESP8266.println(comando);
    long int time = millis();
    while ( (time + timeout) > millis()){
        while (ESP8266.available()){
            char c = ESP8266.read();
            resposta += c;
        }
    }
    if (debug){
        Serial.println(resposta);
    }
    return resposta;
}

void sendHTTPResponse(unsigned int connectionId, String content){
    String httpHeader;
    httpHeader = "HTTP/1.1 200 OK\r\nContent-Type: text/html; charset=UTF-8\r\n";
    httpHeader += "Content-Length: ";
    httpHeader += content.length();
    httpHeader += "\r\n";
    httpHeader += "Connection: close\r\n\r\n";
    httpHeader += content;
    httpHeader += " ";
    String cipSend = "AT+CIPSEND=";
    cipSend += connectionId;
    cipSend += ",";
    cipSend += httpHeader.length();
    ComandoAT(cipSend, 1000);
    ComandoAT(httpHeader, 1000);
}

```