

DIEGO MUNDURUCA DOMINGUES

Sistema com funcionalidades automáticas para auxílio no teste do degrau

São Paulo
2017

DIEGO MUNDURUCA DOMINGUES

Sistema com funcionalidades automáticas para auxílio no teste do degrau

Trabalho de Conclusão de Curso apresentado à
Escola Politécnica da Universidade de São Paulo
para obtenção do título de Bacharel em Engenharia

Orientador: Prof. Dr. Henrique Takachi Moriya

São Paulo
2017

DIEGO MUNDURUCA DOMINGUES

Sistema com funcionalidades automáticas para auxílio no teste do degrau

Trabalho de Conclusão de Curso apresentado à
Escola Politécnica da Universidade de São Paulo
para obtenção do título de Bacharel em Engenharia

Área de Concentração: engenharia biomédica.

Orientador: Prof. Dr. Henrique Takachi Moriya.

São Paulo

2017

Agradecimentos

A Deus por me dar saúde e me abençoar todos os dias.

Aos meus pais, Laercio e Leide, que sempre me apoiaram em tudo, me deram amor e todo o suporte que precisei durante todas as fases da minha vida. Muito obrigado por seus esforços. Tudo o que conquistei e ainda vou conquistar é graças aos dois. Vocês são ótimos pais e tenho muito orgulho de ser filho de vocês.

À minha irmã Luana e prima Renata, por também sempre me ajudarem, principalmente quando ligo super cedo para casa pedindo para tirarem foto de um papel que eu esqueci de trazer comigo.

À minha namorada, Bruna, pelo seu amor e por sempre me deixar calmo e me apoiar nos momentos difíceis. Sou muito grato por Deus ter te colocado na minha vida. Obrigado por entender os momentos em que precisei estudar e me ausentar. Obrigado também por ser minha revisora oficial de textos.

Aos meus amigos do bairro e da escola, que também me deram suporte e sempre estiveram disponíveis para um bom bate-papo e momentos de descontração. Aos meus amigos do cursinho, que fizeram ser divertida aquela fase da minha vida mesmo com as pressões e dificuldades daquela época. Também aos meus amigos da faculdade, pelas risadas, apoio e incentivos para continuar estudando e por aquelas brejas pós semanas de provas.

A todos os professores que tive durante a vida. Cada um de vocês contribuiu com uma parte da minha formação e sou muito grato por isso. Obrigado principalmente ao professor Henrique Moriya, orientador da minha iniciação científica, deste trabalho de conclusão de curso e também de outros que estão por vir. Agradeço por toda a sua paciência e dedicação em me ajudar.

*A imaginação é mais importante
que o conhecimento.*

Albert Einstein

Resumo

O teste do degrau é um teste ergométrico que busca avaliar a capacidade funcional de pacientes. É uma opção de baixo custo, baixa complexidade e fácil realização. Contudo, durante o exame, há várias atividades que devem ser realizadas simultaneamente pelo profissional responsável por acompanhá-lo. Para não haver sobrecarga de trabalho, portanto, geralmente há a necessidade de mais de um profissional para realizar cada teste, o que acaba por consumir recursos e gerar custos nem sempre necessários. Outro problema é a imprecisão dos dados registrados, uma vez que eles são obtidos manualmente. O objetivo deste projeto é desenvolver um sistema com funcionalidades automáticas para auxiliar a realização deste teste.

Palavras-Chave: Teste do degrau. Teste ergométrico. Automatização.

Sumário

1. Introdução	10
1.1 Funcionalidades do Teste do Degrau	10
2. Objetivo	11
3. Requisitos de marketing.....	12
4. Estado da arte	14
5. Materiais e métodos	15
6. Revisão bibliográfica.....	16
6.1 Sistema Android	16
6.2 Arquitetura da plataforma	16
6.3 Arquitetura de segurança	17
6.4 Arquitetura básica de um aplicativo	18
6.5 Atividade.....	18
6.6 Implementação de uma interface	20
6.7 Declaração de uma atividade no manifesto.....	20
7. Implementação	22
7.1 Finalização e upgrade do protótipo anterior	22
7.2 Aplicativo Android.....	23
7.3 Nova implementação com sensoramento utilizando acelerômetro....	24
7.4 Sensores.....	24
7.5 Princípio de funcionamento do sensor.....	25
7.6 Aplicativo.....	27

8. Detalhamento da solução desenvolvida	32
8.1 Comunicação entre aplicativo e circuito sensores.....	32
8.2 Lógica para máquina de estados.....	33
8.3 Lógica de detecção de erros	34
8.4 Lógica para reconhecimento de movimentos	35
9. Resultados	37
9.1 Detecção de movimentos.....	37
9.2 Testes funcionais.....	38
9.3 Requisitos de marketing.....	39
Conclusão.....	41
Trabalhos futuros.....	43
Bibliografia	44

1. Introdução

A intolerância ao exercício nos indivíduos com doença pulmonar obstrutiva crônica (DPOC) tem importante implicação na qualidade de vida, no número de hospitalizações e na sobrevida. Dessa maneira, os testes físicos funcionais têm sido considerados componentes essenciais na rotina clínica de avaliação da capacidade de exercício dos pacientes.

Entre eles, o teste cardiopulmonar (TCP) é considerado o ideal para avaliar a tolerância máxima ao exercício e determinar a etiologia da limitação ao exercício, mas, pelo alto custo e complexidade do equipamento, ainda é pouco utilizado fora dos grandes centros de pesquisas e está longe de ser eficiente, principalmente para a realidade do serviço público de saúde brasileiro.

Como formas alternativas de avaliação aos testes máximos, destacam-se os testes funcionais submáximos. Um exemplo é o teste do degrau (TD), uma opção de simples realização, baixa complexidade e custo reduzido, o que facilita sua aplicação na prática clínica.

1.1 Funcionalidades do Teste do Degrau

O teste consiste em fazer o paciente subir e descer de um degrau, obedecendo a comandos dados por um profissional de saúde. A frequência das ordens de subir e descer é pré-estabelecida de acordo com um protocolo experimental. Em alguns protocolos, a frequência não é alterada durante todo o exame e em outros é alterada de tempos em tempos.

O profissional deve ficar atento se o paciente está conseguindo fazer os movimentos de subir e descer do degrau, obedecendo à frequência estabelecida. Caso o paciente falhe em acompanhar o ritmo desejado, pode ser um indicativo de que ele está chegando a um esforço máximo e, portanto, o instante de tempo em que essa falha aconteceu deve ser registrada para análise.

2. Objetivo

O objetivo desta pesquisa é produzir um dispositivo que possibilite a análise automática do teste do degrau. Deseja-se desenvolver uma maneira de monitorar a movimentação do paciente e de fornecer estes dados como entrada para um algoritmo, que faça um registro preciso de quando o paciente falhou em seguir o ritmo proposto. Da mesma forma, espera-se conseguir registrar determinados dados que não podem ser obtidos de forma manual, o que possibilitaria um fornecimento mais completo de informações a respeito do condicionamento cardiopulmonar do paciente.

Além disso, o professor orientador deste projeto tem parceria em pesquisas de professores do curso de fisioterapia do Centro Universitário São Camilo (CUSC). Alguns alunos orientados por estes docentes se propuseram a realizar um trabalho de formatura em que eles analisariam os dados obtidos através do dispositivo desenvolvido neste trabalho. Assim, foi proposto o fornecimento de um protótipo para este grupo, em tempo hábil para realizarem a sua pesquisa.

3. Requisitos de marketing

Os requisitos de marketing foram levantados a partir das necessidades declaradas pelo cliente e também de características consideradas essenciais à solução proposta neste projeto. São eles:

Tabela 1 – Requisitos de marketing

Requisitos de marketing	
1	Baixo Custo
2	Fácil Uso
3	Portátil
4	Interatividade
5	Confiabilidade do resultado

Fonte: o autor.

Fácil uso – Seguindo as características do teste do degrau padrão, deseja-se manter a simplicidade da realização dos exames, desenvolvendo um sistema que não necessite um treinamento técnico muito específico para o profissional de saúde. Dessa forma, pretende-se desenvolver um dispositivo intuitivo, com interface amigável e manipulação simples.

Portátil – Ainda respeitando a facilidade desejada do teste do degrau, a ideia foi criar um sistema portátil, para que o exame possa ser realizado sem a necessidade de uma aparelhagem complexa e de difícil transporte.

Interatividade – Deseja-se criar um sistema em que o profissional possa inserir informações, tais como dados do paciente, a frequência de movimentação requerida, o tipo de protocolo que deve ser adotado, etc. Da mesma forma, objetiva-se também que o sistema informe constantemente ao usuário o seu status, o momento exato em que um erro de movimentação ocorrer, o tempo do exame, etc.

Confiabilidade do resultado – Por se tratar de um exame médico, é de grande importância que os resultados fornecidos pelo sistema tenham uma boa precisão, pois isso pode influenciar o diagnóstico do paciente. Assim, é necessário que o reconhecimento do movimento seja confiável, bem como o registro do tempo total do exame. Considerando que mais de um teste será registrado na memória do sistema, é necessário desenvolver uma base de dados bem organizada e eficiente.

4. Estado da arte

O primeiro projeto de automatizar funcionalidades do teste do degrau foi iniciado no Trabalho de Formatura de dois alunos do curso de graduação em Engenharia Elétrica da Escola Politécnica da Universidade de São Paulo, no ano de 2016. Estes alunos desenvolveram um protótipo de uma solução que consistia em um degrau de madeira com sensores, que funcionam por contato mecânico; e um Arduino, utilizado para o processamento da movimentação e o cálculo dos erros cometidos pelo paciente (figura 1).

A solução possui alguns problemas: os sensores não são muito eficientes e, geralmente, falham em detectar a posição do paciente no decorrer do exame; o degrau é pesado e difícil de ser transportado; e, por fim, praticamente não há interatividade com o usuário - a única informação que ele recebe é um aviso de que cometeu um erro, caso uma luz LED acenda.

Figura 1 - Protótipo desenvolvido em Trabalho de Formatura na EPUSP, em 2016



Fonte: o autor.

5. Materiais e métodos

Na fase inicial deste trabalho, o objetivo central foi aprimorar o protótipo que já havia sido desenvolvido. A principal razão para esta decisão foi a identificação de melhorias que poderiam ser implementadas, além da facilidade em poder reaproveitar conteúdos que já haviam sido desenvolvidos. Outro fato relevante para a decisão foi que o grupo que pretende fazer as análises dos dados obtidos pela solução proposta nesse trabalho necessitavam de um protótipo funcional a curto prazo, para poderem realizar as suas respectivas pesquisas. Assim, consertar e melhorar o projeto inicial seria mais rápido do que desenvolver outra solução do princípio.

Numa segunda fase, a ideia foi desenvolver um novo projeto que reparasse os problemas identificados no trabalho anterior, como as falhas no sensoriamento, portabilidade e interatividade com o usuário.

Por fim, foram realizados testes de funcionalidades com ajuda de voluntários. Isso permitiu avaliar o desempenho da solução desenvolvida e verificar se os requisitos de marketing propostos foram cumpridos.

Também serão realizados testes com o acompanhamento das estudantes de fisioterapia da CUSC, com as quais há uma parceria. Elas ajudaram, principalmente, utilizando e fornecendo *feedbacks* sobre o protótipo antigo. No entanto, também auxiliaram no desenvolvimento do novo projeto, sobretudo nas especificações de marketing e por meio de impressões sobre o novo protótipo desenvolvido, após a realização de alguns testes de funcionalidades.

6. Revisão bibliográfica

A revisão bibliográfica consiste na apresentação de conceitos relacionados à implementação do aplicativo Android. Neste capítulo, pretende-se apresentar os recursos e ferramentas utilizados para desenvolver o projeto. Para que o capítulo não fique demasiadamente longo, algumas questões básicas sobre o desenvolvimento de *software* não serão abordadas em detalhes.

6.1 Sistema Android

O Android é uma plataforma aberta, desenvolvida pela empresa Google, que possui uma estrutura avançada, permitindo o desenvolvimento de vários tipos de aplicativos. São fornecidas várias opções de linguagens de programação que o desenvolvedor pode optar para criar aplicativos, sendo o Java a oficial e mais utilizada.

O sistema foi projetado a partir de uma versão do *kernel* do Linux, aproveitando seus recursos de gerenciamento de atividades centrais do sistema, como gestão de processos, gestão de memória e segurança. O *kernel* também atua como uma camada de abstração entre o *hardware* e o resto do *software*.

6.2 Arquitetura da plataforma

O Android possui camadas com diferentes níveis de complexidade. Isso fornece maiores facilidades ao desenvolvedor para criar suas aplicações, pois há diferentes ferramentas com variados níveis de abstrações.

A camada de mais alto nível é composta por aplicações básicas do sistema, como e-mail, calendário e etc. O desenvolvedor pode simplesmente requisitar a utilização de funcionalidades dessas aplicações. Por exemplo, se o aplicativo a ser desenvolvido precisa enviar um e-mail, o desenvolvedor não necessita implementar uma ferramenta para envio de e-mail, basta solicitar que o aplicativo de e-mail padrão do sistema realize esta tarefa.

Em camadas intermediárias, o Android possui APIs Java, com blocos de códigos para reutilização de componentes. Ela é composta, por exemplo, por um sistema de visualização, listas, caixas de textos e botões, assim como APIs de gerenciamento de recursos, gerenciamento de notificações, gerenciamento de atividades, entre outros.

Outra camada intermediária é composta por bibliotecas nativas das linguagens C e C++, facilitando o uso de código e de funcionalidades dessas linguagens. Também há a camada de abstração de *hardware*, capaz de gerenciar acesso a serviços como *Bluetooth* e câmera. Assim, o desenvolvedor conta com uma interface de abstração para que possa trabalhar com essas ferramentas diretamente no código Java.

Por fim, a camada mais inferior é o *kernel* do Linux, que realiza funcionalidades como encadeamento e gerenciamento de memória de baixo nível. Além disso, fornece os recursos de segurança principais e facilidades para os fabricantes desenvolverem *drivers* de *hardware* para *kernel* Linux.

6.3 Arquitetura de segurança

O princípio básico de segurança dentro do Android é que nenhum aplicativo tenha permissão para realizar operações que prejudiquem outros aplicativos. Assim, não é permitido a um aplicativo acessar arquivos de outro aplicativo, ou usar recursos do sistema sem permissão prévia fornecida pelo usuário.

Para isolar um aplicativo, para que este só possa acessar seus arquivos, o Android usa a base do conceito de usuários do Linux. No Linux, os usuários não têm permissão de acesso sobre arquivos de outros usuários. Assim, o Android trata cada aplicativo como se fosse um usuário, com identificação distinta dentro do sistema.

Para acessar recursos adicionais do sistema, como uso da câmera, GPS, manter o dispositivo ativo e etc., é necessário que o aplicativo solicite permissão ao sistema. Os aplicativos declaram as permissões que exigem e o sistema Android solicita o consentimento do usuário. Quando um aplicativo é executado, a primeira vez

que é necessário utilizar o recurso adicional desejado, é solicitado que o usuário autorize o seu acesso.

Cada aplicativo recebe um certificado, para possibilitar tanto a identificação do desenvolvedor do aplicativo, quanto que o sistema negue solicitação de aplicativos fraudulentos, que queiram receber a mesma identidade de “usuário” de outro aplicativo. Assim, não é possível, por exemplo, criar um aplicativo que tente ter a mesma identidade de um aplicativo de banco, para acessar seus arquivos com informações sigilosas do usuário.

6.4 Arquitetura básica de um aplicativo

Nessa seção, serão exploradas as ferramentas básicas que constituem um aplicativo e os recursos fornecidos pelo Android para facilitar o seu desenvolvimento.

6.5 Atividade

Uma atividade representa uma tela de um aplicativo. Basicamente, é a partir dela que se inicia o desenvolvimento de um aplicativo, que necessita de uma interface gráfica com o usuário. Cada atividade é vinculada a um arquivo de interface gráfica, que será melhor detalhado em uma seção posterior.

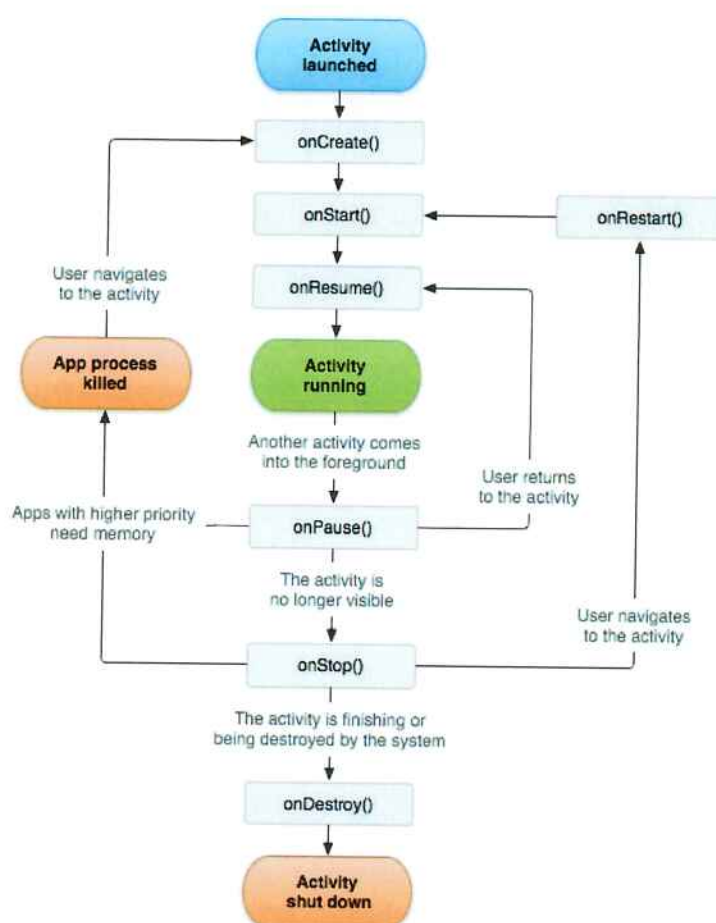
Um aplicativo pode ter várias atividades, sendo uma a principal, que é apresentada ao usuário ao iniciar o aplicativo pela primeira vez. Cada atividade pode iniciar outras atividades que executam funções diferentes e estão relacionadas a diferentes arquivos de *layout*.

O sistema possui uma pilha de atividades do tipo LIFO (*last in, first out*). Sempre que uma atividade é iniciada, ela entra na pilha e quando outra é iniciada, aquela é interrompida e a nova atividade também entra na pilha. Quando o usuário terminar a atividade atual, ela é destruída e sai da fila, e a atividade anterior é retomada.

As atividades possuem ciclos de vida, notificados pelo sistema sempre que for necessário alterar o seu estado. São eles: criação, interrupção, retomada e destruição.

Quando uma atividade é interrompida, devido ao início de uma nova atividade, ela é notificada acerca dessa alteração de estado, por meio de métodos de retorno de chamada do ciclo de vida da atividade. Há diversos métodos de retorno de chamada que uma atividade pode receber devido a uma alteração em seu estado — quando o sistema a está criando, interrompendo, retomando ou destruindo — e cada retorno de chamada oferece uma oportunidade de executar trabalhos específicos adequados a essa alteração de estado. Por exemplo: quando interrompida, a atividade deve liberar todos os objetos grandes, como conexões com a rede ou com um banco de dados. Quando a atividade for retomada, será possível readquirir os recursos necessários e retomar as ações interrompidas. Essas transições de estado são parte do ciclo de vida de cada atividade.

Figura 2 – Ciclo de vida de um aplicativo



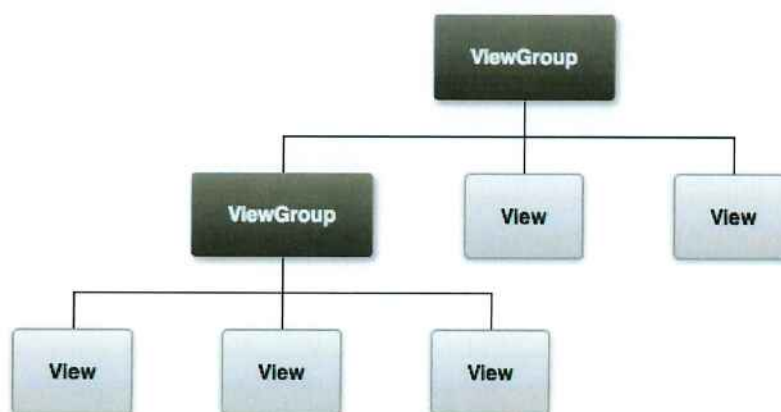
Fonte: site <developer.android.com>.

6.6 Implementação de uma interface

Todos os elementos da interface do usuário em um aplicativo para Android são criados usando objetos *View*. Uma *View* é um objeto que desenha algo na tela com o qual o usuário pode interagir. As subclasses do objeto *View* fornecem funções para controle de entradas. Um botão, por exemplo, é um objeto derivado da classe *View*, que possui um método executado quando o usuário o pressiona.

O *layout* da interface é definido com uma hierarquia de objetos *View*. Há a possibilidade de controle da posição na tela em objetos que se encontram em um mesmo nível da hierarquia.

Figura 3 – Hierarquia de objetos *View*



Fonte: site <developer.android.com>.

6.7 Declaração de uma atividade no manifesto

O arquivo *manifest* é onde são declaradas as informações essenciais sobre o aplicativo ao sistema Android. Esse arquivo é acessado pelo sistema antes mesmo de ser executado o código do aplicativo.

No arquivo manifesto, é especificado o nome do pacote do aplicativo, que é um identificador exclusivo. Além disso, são descritos todos os componentes, incluindo as atividades detalhadas anteriormente. Também é nesse arquivo que são declaradas as permissões exigidas pelo aplicativo. No aplicativo criado nesse trabalho, por

exemplo, foi necessária uma permissão para usar o *Bluetooth* e para gravar e ler arquivos no armazenamento interno.

Neste arquivo também é declarado o nível mínimo da versão do Android que o aplicativo pode ser executado. É importante declarar essa restrição, pois versões antigas do sistema Android podem carecer de determinadas ferramentas utilizadas pelo aplicativo desenvolvido.

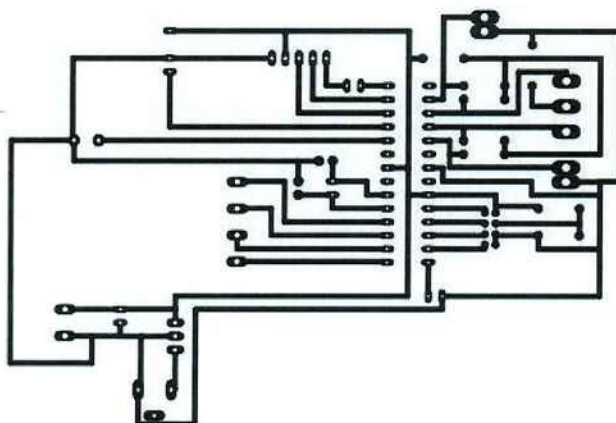
7. Implementação

7.1 Finalização e *upgrade* do protótipo anterior

O Arduino Uno, que era utilizado neste protótipo, foi substituído por um circuito com micro controlador atmega328, e o degrau feito de madeira foi reaproveitado, bem como os tapetes emborrachados e os sensores que funcionam por contato.

O circuito foi esquematizado utilizando o *software* Multisim 14.0 (NI, EUA) e desenhado com o *software* Ultiboard 14.0 (NI, EUA).

Figura 4 – Layout do circuito



Fonte: o autor.

Figura 5 – Circuito eletrônico



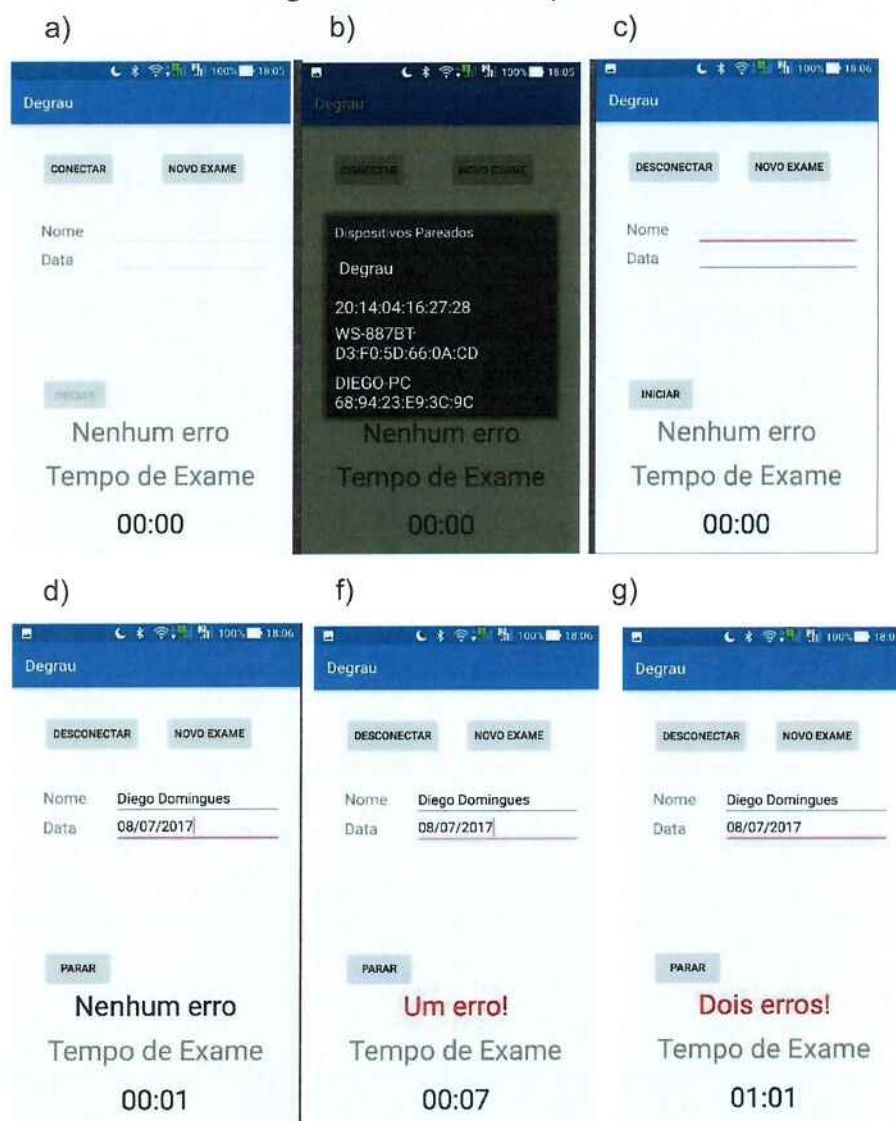
Fonte: o autor.

Os componentes desse circuito são: microprocessador ATmega328, *shield* para cartão de memória, botões *push*, LED, resistores, capacitores, conector para fonte de alimentação e conectores para ligações em geral. Este circuito foi instalado no degrau de madeira, reaproveitado do projeto anterior.

7.2 Aplicativo Android

O aplicativo foi desenvolvido em linguagem de programação Java, através da interface de desenvolvimento Android Studio (Google, EUA). A comunicação entre o *smartphone* e o sistema é feita por *Bluetooth*.

Figura 6 - Telas do aplicativo



Fonte: o autor.

7.3 Nova implementação com sensoriamento utilizando acelerômetro

O dispositivo desenvolvido possui duas partes principais: um conjunto de sensores e um aplicativo Android. Os sensores são instalados nos tornozelos do paciente e têm como função determinar a posição dos pés no momento do exame, bem como enviar a informação para o aplicativo, que processa os dados e apresenta os resultados ao usuário.

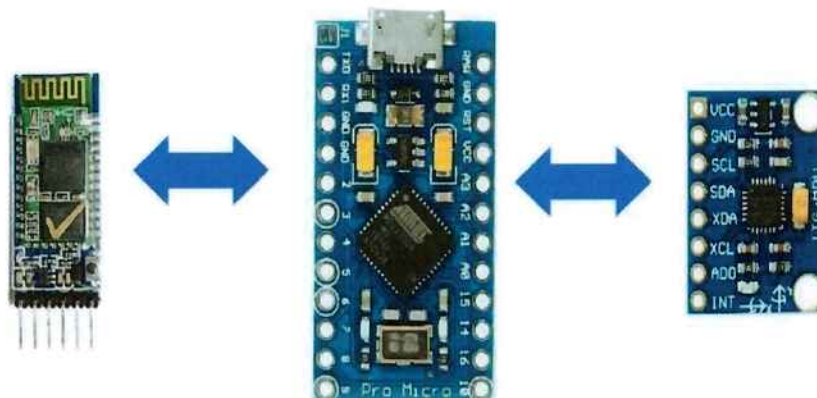
7.4 Sensores

Considerando a necessidade de reconhecer a posição dos pés do paciente, para poder informar se ele está realizando uma movimentação sem erros, e o fato do sensor por contato mecânico apresentar falhas durante a realização do exame - além do grande volume e peso do degrau em que ele foi instalado - um sensor mais eficaz foi desenvolvido neste projeto.

Primeiro, foi verificada a possibilidade de monitorar a movimentação dos pés através de dados de aceleração, já que, conhecendo a posição inicial dos pés, é possível processar os dados de aceleração advindos dos sensores para reconhecer a realização de um movimento.

Os sensores foram desenvolvidos com o uso de 3 módulos: um Arduino micro, um módulo *Bluetooth* e um módulo acelerômetro (Figura 7). Além disso, foi utilizada também uma bateria de lítio para alimentação.

Figura 7 – Componentes do circuito sensor



Fonte: sites <autocorerobotica.com.br>, <banggood.com> e <embarcg.com.br>.

Figura 8 – Sensor finalizado



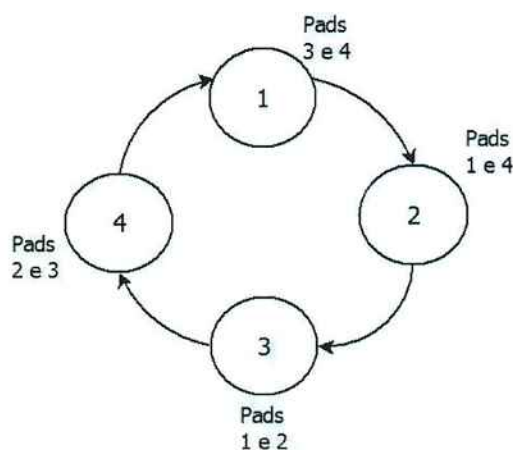
Fonte: o autor.

7.5 Princípio de funcionamento do sensor

Há 4 posições possíveis no teste do degrau. O trabalho do Arduino empregado neste projeto é processar os dados advindos dos 3 eixos, reconhecer se houve movimentação e reconhecer se a posição atual era realmente a posição em que o paciente deveria estar.

O reconhecimento de falha funciona com base em uma máquina de estado (figura 9). Em cada estado, é esperado um posicionamento específico do paciente.

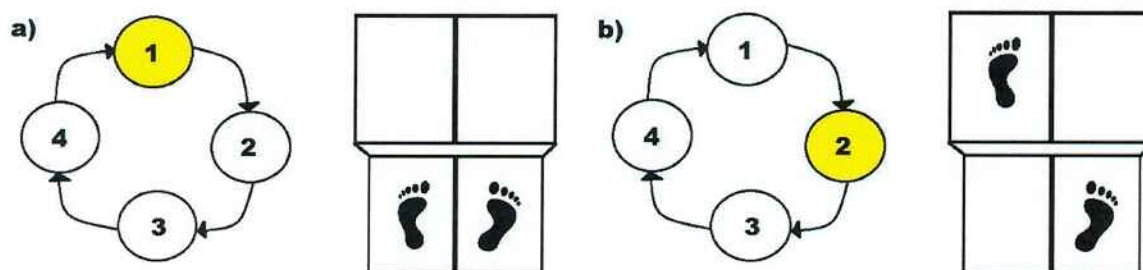
Figura 9 – Máquina de estado



Fonte: o autor.

Por exemplo, no estado 1, é esperado que o paciente esteja sobre os *pads* 3 e 4 (figura 10-a). Em seguida, o sistema gera um sinal sonoro, avança para o estado 2 e espera que o paciente se posicione sobre os *pads* 1 e 4 (figura 10-b). Se essa sequência não for cumprida, o sistema registra um erro. A regra geral definida pelo cliente é que, na ocorrência do primeiro erro, o exame não seja interrompido. Porém, é gerado um sinal ao profissional médico, indicando que houve erro. Na ocorrência do segundo erro, então, o exame é interrompido – um novo sinal indicativo de erro é gerado e, por fim, é registrado o tempo total de exame.

Figura 10 – Máquina de estados e o posicionamento esperado por parte do paciente



Fonte: o autor.

O momento de transição para um novo estado, ou seja, a velocidade em que o paciente deve movimentar-se, foi feita com base em um arquivo de áudio contendo centenas de sinais sonoros de curta duração. Este arquivo foi fornecido pela equipe médica que acompanha este projeto e trata-se do mesmo arquivo utilizado nos testes do degrau realizados da maneira tradicional.

Cada vez que um sinal sonoro é reproduzido, o aplicativo envia uma solicitação ao Arduino para que ele mude para a posição seguinte, dentro da máquina de estados.

Para reconhecer se houve uma movimentação, o Arduino usa um limiar de movimentação: caso a variação da aceleração em dois eixos seja maior que um certo limiar definido empiricamente, entende-se que o usuário realizou um movimento.

Maiores detalhes sobre o funcionamento desse algoritmo serão fornecidos em capítulos posteriores.

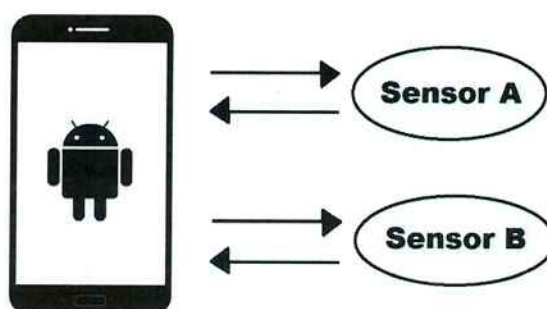
7.6 Aplicativo

No aplicativo, a escolha pelo sistema operacional Android foi feita com base na sua popularidade – pesquisas indicam que mais de 90% dos celulares no Brasil possuem esse sistema. A linguagem de programação utilizada foi a Java e a interface de desenvolvimento escolhida foi a Android Studio (Google, EUA).

A comunicação entre o smartphone e os sensores é feita por *Bluetooth*. O aplicativo possui os endereços MAC dos módulos *Bluetooth* dos sensores e, dessa forma, quando é requisitado que o aplicativo inicie a conexão, não é necessário que o usuário realize previamente o pareamento com os dispositivos.

O aplicativo gera duas instâncias de conexão, uma para cada sensor. Assim, é possível enviar comandos para cada um dos sensores separadamente, bem como reconhecer de onde partiu um pacote de informação recebida (figura 11).

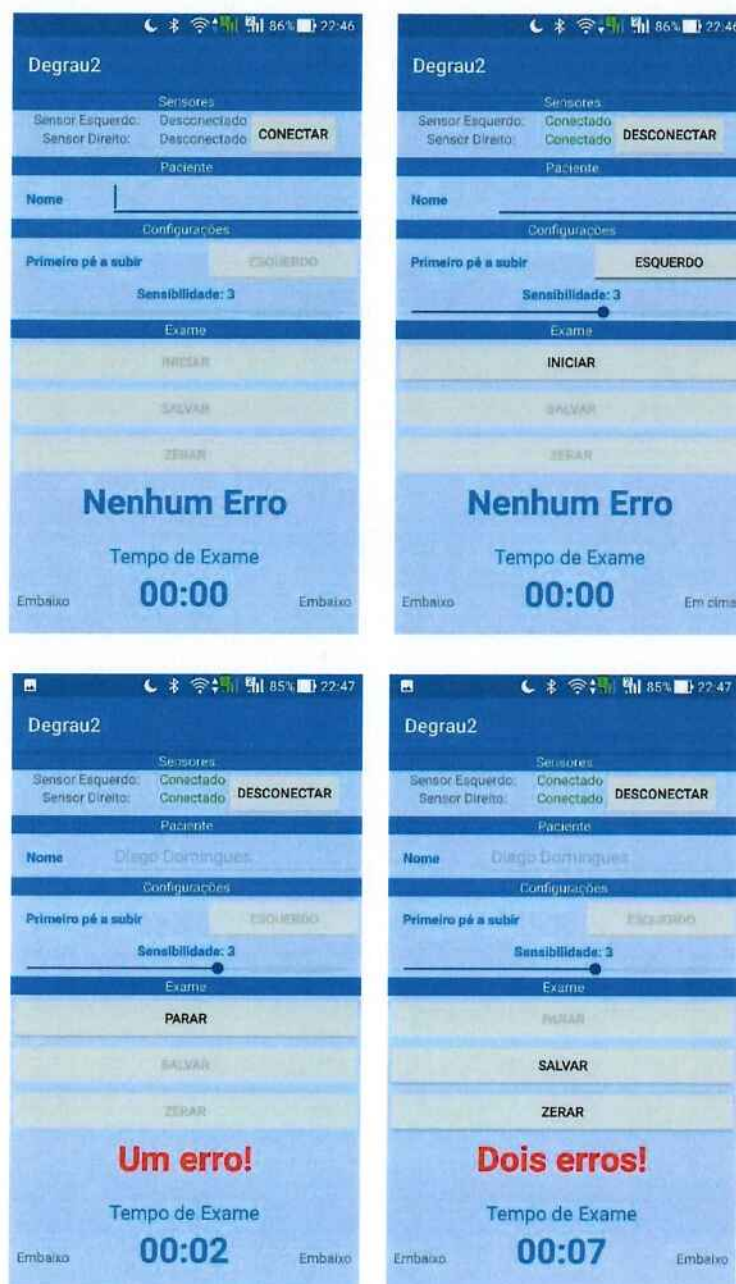
Figura 11 – Comunicação *Bluetooth* entre celular e sensores



Fonte: o autor.

Depois de ser devidamente conectado, o aplicativo precisa de 3 entradas do usuário: uma identificação do paciente, a escolha de qual pé o paciente deve iniciar o movimento e o grau de sensibilidade do sensor.

Figura 12 – Telas do aplicativo



Fonte: o Autor

A identificação do paciente pode ser feita de acordo com o que o usuário definir. O texto pode conter letras, números, caracteres especiais e espaço. A identificação é registrada em um arquivo de formato texto (.txt), gerado quando o usuário escolhe salvar o exame, em conjunto com outras informações.

Escolher um pé para iniciar o movimento é necessário, por causa do protocolo do exame: o paciente deve descer do degrau com o mesmo pé que escolheu para subir. Para o correto funcionamento do algoritmo de detecção de erros, esse pé deve ser informado antes de iniciar o teste.

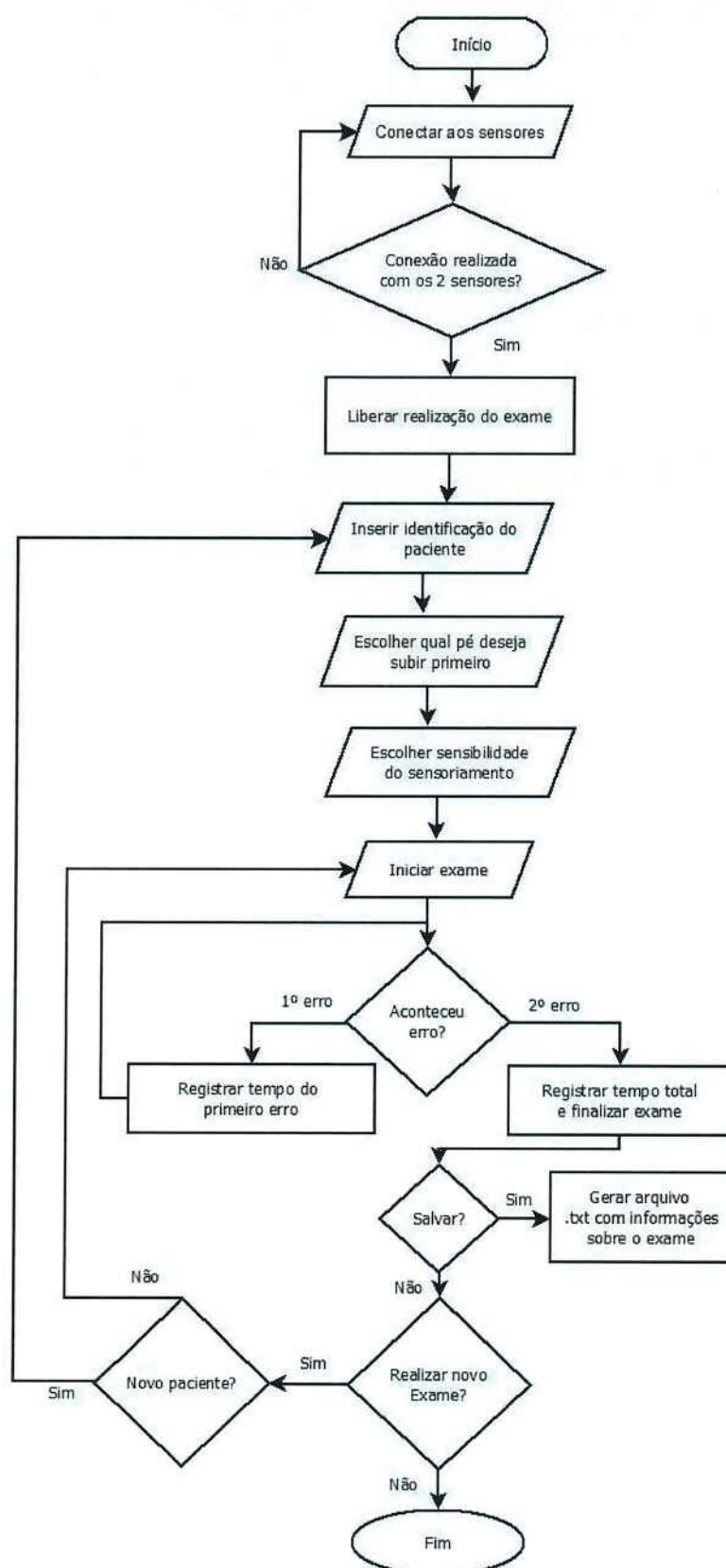
Por fim, antes de iniciar o exame o usuário deve escolher também a sensibilidade do sensor. A sensibilidade varia numa escala de 1 a 5 e, quanto maior o valor escolhido, mais facilmente o aplicativo vai registrar um movimento realizado como sendo um passo. Portanto, é recomendado que seja realizado um breve teste com o paciente, fazendo-o subir e descer algumas vezes do degrau, para que seja escolhido um valor de sensibilidade adequado.

Feito isso, o usuário pode iniciar o exame. O aplicativo mostra na tela um cronômetro com o tempo decorrido e a quantidade de erros cometidos pelo paciente até o momento. Durante a realização, a principal atividade do aplicativo é reproduzir sinais sonoros, que indicam quando o paciente deve executar um movimento.

Conforme definido no protocolo, é permitido que o paciente cometa somente um erro durante todo o teste. Quando o segundo erro ocorrer, o aplicativo para o cronômetro e o exame é finalizado. Então, é permitido ao usuário salvar os dados ou iniciar um novo exame.

O aplicativo funciona de acordo com o seguinte fluxograma:

Figura 13 - Fluxograma de funcionamento do aplicativo



Fonte: o autor.

O arquivo texto que o usuário pode salvar contém as seguintes informações: identificação do paciente, data de realização do exame, duração, quantidade de passos realizados, momento em que aconteceu o primeiro erro, o pé escolhido para subir primeiro, exame realizado até o fim ou interrompido, em quais posições o paciente estava pisando durante o segundo erro e qual a posição em que ele deveria estar pisando para não cometer o erro.

8. Detalhamento da solução desenvolvida

Neste capítulo, pretende-se apresentar de forma mais detalhada alguns conceitos importantes para realização do projeto, bem como a maneira com a qual foram implementadas as partes desta solução.

8.1 Comunicação entre aplicativo e circuito sensores

O aplicativo e os sensores precisam trocar informações durante a realização do exame, pois há diversas atividades que uma parte solicita e a outra precisa que executar. Por exemplo, o aplicativo recebe do usuário o valor da sensibilidade e deve solicitar aos sensores que atualizem o valor da variável interna de sensibilidade.

Geralmente, na mensagem que uma parte envia à outra, há a necessidade de informar qual função deve ser executada ou qual variável deve ser alterada, juntamente com os argumentos da função ou o novo valor da variável.

Assim, foi definido um formato padrão para as mensagens trocadas durante a comunicação. Uma variável do tipo *String* foi desenvolvida, concatenando uma parte composta pelo nome da função ou variável e outra parte composta pelos argumentos ou valores (figura 14).

Figura 14 – Padrão das mensagens enviadas por *Bluetooth*



Fonte: o autor.

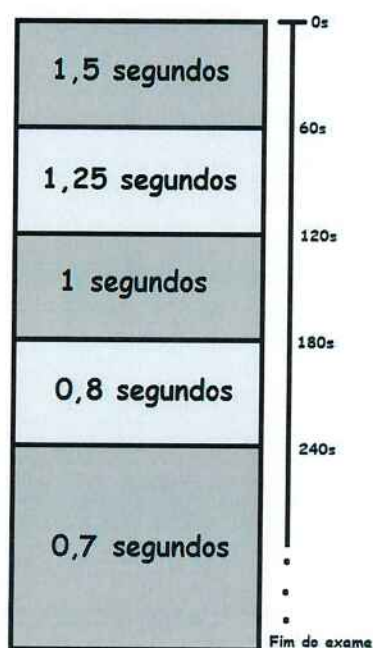
8.2 Lógica para máquina de estados

Como já citado em seções anteriores, foi definida uma máquina de estados para implementar a lógica de onde o pé do paciente deveria se encontrar durante o decorrer do exame. Essa lógica é executada dentro do código do aplicativo, e o motivo dessa escolha é que os recursos presentes no Java facilitam a sua implementação. Além disso, o aplicativo é a parte do projeto que recebe e envia as informações para as outras duas partes.

O recurso principal aproveitado do aplicativo foi a facilidade para implementar uma *thread*. A *thread* executa uma parte do algoritmo responsável por alterar o estado atual da máquina, enviar uma mensagem para os sensores do valor do novo estado e executar o comando que reproduz um sinal sonoro para o usuário.

Como no decorrer do exame a velocidade dos movimentos aumenta, uma lógica para a *thread* ser chamada com intervalos de tempos variados foi implementada. No arquivo de áudio executado no teste do degrau tradicional, a velocidade não cresce linearmente. Há intervalos de tempos definidos no qual a velocidade é alterada (figura 15).

Figura 15 – Intervalo de tempo entre sinais sonoros reproduzidos no teste do degrau



Fonte: o autor.

Portanto, no início do exame a *thread* é executada de 1,5 em 1,5 segundos. Quando se passa 1 minuto, a *thread* passa a ser chamada de 1,25 em 1,25 segundos e assim por diante.

8.3 Lógica de detecção de erros

Os circuitos dos sensores executam a lógica para detecção de erros do usuário. Nessa lógica, é fundamental saber a posição em que o pé do usuário se encontra, bem como a posição em que o pé deveria estar.

Como já mencionado nas seções anteriores, a informação de onde o pé do paciente deveria se encontrar é extraída a partir do valor atual da máquina de estados fornecido pelo aplicativo: quando há alteração entre estados, o aplicativo envia uma mensagem para os sensores, para que eles atualizem o valor do estado atual.

Como em cada um dos 4 estados, o pé do paciente deveria estar em uma posição específica, o que o algoritmo faz é comparar a posição atual do pé e a posição em que o pé deveria estar. As duas posições devem ser iguais antes que aconteça uma nova transição de estado; caso contrário, um erro é detectado.

Na prática, isso significa que, quando acontece um sinal sonoro (troca de estado), o paciente deve realizar o movimento correto até que um novo sinal sonoro aconteça (uma nova troca de estado).

Quando um erro é detectado, é interrompida a execução do algoritmo de detecção de erros por 3 segundos. Esse tempo permite que o paciente perceba que cometeu um erro e tente acelerar seus movimentos, para que não seja detectado um segundo erro, que finalizaria o exame.

O aplicativo é informado sobre os erros e apresenta essas informações para o usuário. Uma lógica foi implementada para que, se um sensor já tiver informado um erro, caso o outro sensor também informe, essa segunda informação seja ignorada, pois, na prática, os dois erros informados pelos sensores são o mesmo.

Neste caso específico, seria mais lógico que um sensor, além de informar o aplicativo sobre o acontecimento do erro, também informasse o outro sensor. Porém,

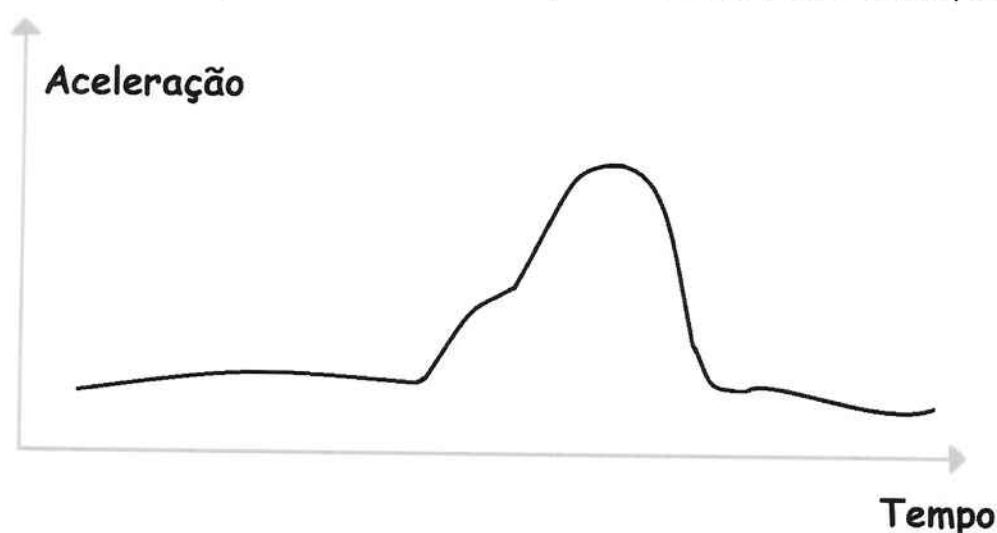
os módulos *Bluetooth* utilizados neste projeto só podem estabelecer uma conexão por vez, e esta conexão é feita com o aplicativo.

8.4 Lógica para reconhecimento de movimentos

Os movimentos são detectados a partir do sinal de aceleração, conforme já citado. Esses sinais são obtidos pelo sensor acelerômetro com frequência de amostragem de 20Hz e não passam por nenhum filtro, uma vez que se faz necessário um processamento em tempo real. Assim, tarefas que possuem um tempo computacional elevado para serem executadas, como um filtro, poderiam atrapalhar a eficiência da solução proposta.

As movimentações foram detectadas a partir da variação do valor de aceleração. Durante a execução de um movimento, o sinal se comporta conforme a figura 16. A aceleração, que era constante, sofre um incremento ou um decremento no seu valor e, em seguida, quando o movimento é finalizado, a aceleração torna-se novamente constante.

Figura 16 – Comportamento da aceleração em um eixo durante um passo



Fonte: o autor.

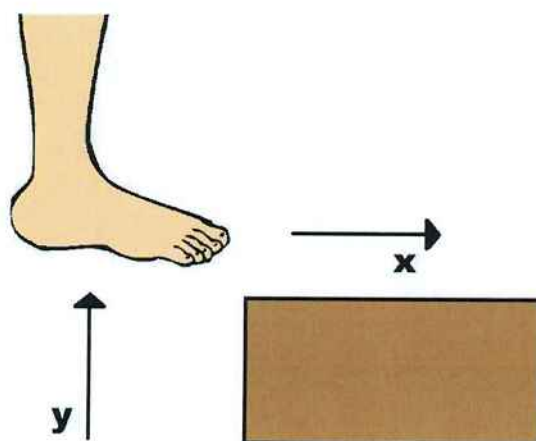
A detecção foi implementada da seguinte maneira: primeiro, foi calculada a variância de um intervalo de 600ms dos 3 eixos do acelerômetro. Se o valor da

variância ultrapassar um determinado valor, é registrado um evento naquele eixo. Para um movimento ser realizado, conforme indicado na figura 18, pelo menos dois eixos devem presenciar um evento. Assim, se dois eventos forem registrados ao mesmo tempo em eixos diferentes, um movimento é marcado.

Figura 17 – Fórmula da variância

$$Variância = \sum_{i=1}^{12} \frac{(x_i - x_m)^2}{11}$$

Figura 18 – Movimento do pé durante um passo (movimento em pelo menos 2 eixos)



Fonte: o autor.

Para reconhecer se o movimento realizado foi uma subida ou uma descida do degrau, uma lógica simples foi implementada: considerando que, no início do exame, o paciente começa fora do degrau, quando for detectado o primeiro movimento, é porque o paciente subiu no degrau; quando for detectado um segundo movimento, o paciente desceu do degrau, e assim por diante.

9. Resultados

A partir da realização deste projeto, foi desenvolvido um sistema para automatizar o teste do degrau, finalizado já em um nível de protótipo funcional. Isto significa que, apesar de o sistema ainda precisar passar por mais testes, principalmente em ambiente clínico-hospitalar, já foi possível avaliar o seu desempenho e, assim, validar o trabalho realizado.

Para testar o desempenho do protótipo, foram realizados testes para verificar se ele atendia aos objetivos do trabalho e aos requisitos de marketing. Assim, foi possível observar os aspectos que foram alcançados com sucesso e também os que necessitam de melhorias em trabalhos futuros.

9.1 Detecção de movimentos

Esse é a parte com maior relevância para que a solução desenvolvida tenha um bom desempenho, pois é onde houve maior necessidade de pesquisa e inovação, o que oferece maiores riscos para a performance.

Na realização desta pesquisa, foram testadas algumas alternativas de processamento dos dados de aceleração, para chegar a uma decisão de qual algoritmo de processamento seria utilizado. A princípio, foi escolhido um algoritmo com uma excelente capacidade em reconhecer movimentos que não eram passos reais. Porém, havia momentos em que o paciente realizava passos com movimentações peculiares, processo característico de quando o paciente já estava sentindo fadiga e dificuldade em acompanhar o ritmo. Nesses tipos de movimentos, esta versão inicial do algoritmo apresentava falhas.

Em outras palavras, isto significa que o algoritmo tinha uma especificidade muito elevada (capacidade de reconhecer movimentos que não era passos), mas uma sensibilidade não ideal. No teste do degrau, não identificar que o paciente realizou um movimento implica na detecção de um erro. Então, se houver problemas nessa detecção, o resultado do exame torna-se incorreto.

Por isso, um novo algoritmo foi desenvolvido, com menos restrições – ou seja, qualquer movimento com a perna que faça dois eixos do acelerômetro variarem acima de um limiar é registrado como um passo. Isso faz com que a especificidade da detecção de passos seja menor, mas torna a sensibilidade (capacidade de reconhecer passos) próxima a 100%.

Apesar da baixa especificidade apresentada por esse algoritmo, durante o exame o paciente não realiza movimentos falsos, ou seja, quando o paciente movimento as pernas, realmente são passos que são realizados. Assim, simplesmente detectar movimentação da perna, sem a certeza de que se trata de um passo, já seria suficiente para a realização do projeto.

9.2 Testes funcionais

Entre o desenvolvimento e a implementação de um projeto, podem surgir diversos problemas que não eram previstos. Por isso, é importante realizar testes de funcionamento, para evitar que imprevistos e problemas possam surgir quando o cliente estiver utilizando a solução.

O teste do degrau foi realizado com o protótipo desenvolvido em 14 indivíduos voluntários, com o seguinte procedimento: não foram oferecidos detalhes sobre o funcionamento do sensor, principalmente no que concerne à monitoração a partir da aceleração. A ideia foi garantir que o paciente não fizesse qualquer tipo diferente de movimento que pudesse privilegiar o sistema e contribuir com sua performance.

Em seguida, foi solicitado que os voluntários realizassem o exame, sem nenhuma restrição de tempo, até serem detectados dois erros. Paralelamente, um acompanhamento visual foi feito durante a realização do exame, registros do tempo em que ocorreram os erros. Os dados dos exames estão dispostos na tabela a seguir.

Tabela 2 – Resultado do teste funcional utilizando 14 voluntários

Indivíduo	Tempo total de exame pelo sistema (minutos)	Tempo total de exame análise visual (minutos)	Erro (minutos)
1	4:50	4:50	0:02
2	7:16	7:14	0:02
3	8:59	9:02	0:03
4	2:14	2:16	0:02
5	5:12	5:15	0:03
6	1:06	1:04	0:02
7	0:20	0:20	0:00
8	3:04	3:03	0:01
9	5:02	5:05	0:03
10	0:55	0:57	0:02
11	5:15	5:18	0:03
12	1:29	1:30	0:01
13	3:13	3:15	0:02
14	1:01	1:02	0:01
Média	3:57	3:58	0:0193

Fonte: o autor.

9.3 Requisitos de marketing

Foram definidos cinco requisitos de marketing durante a fase de projeto. A seguir, será explicado como cada um desses objetivos foi cumprido.

Baixo custo: O sistema desenvolvido é composto por componentes de baixo custo. O módulo *Bluetooth*, Arduino, bateria de lítio e o módulo acelerômetro custaram, no total, R\$ 86,00 – um preço considerado acessível.

Fácil uso: Uma das ideias de desenvolver um aplicativo para celular foi facilitar a interatividade e comunicação com o usuário. As informações são apresentadas de maneira objetiva na tela e o usuário não precisa de nenhum treinamento específico para manipular o sistema.

Portátil: Essa foi a principal característica que diferencia este projeto do desenvolvido anteriormente, com sensores instalados em um degrau físico. Os novos sensores desenvolvidos são leves e pequenos, podendo ser transportados para qualquer local sem qualquer dificuldade.

Interatividade: No sistema, o usuário pode configurar e manipular várias funções. Na solução anterior, o usuário somente pressionava um botão para iniciar o teste e, quando o exame acabava, era informado a partir da sinalização de um LED. Na solução atual, o usuário pode escolher a sensibilidade do sensor, informar o nome com o qual quer ser identificado, acompanhar na tela o tempo total de exame decorrido e os instantes em que cometeu os dois erros.

Confiabilidade do resultado: O sistema desenvolvido não apresentou falhas durante os testes realizados. No entanto, mais testes são necessários, inclusive utilizando pacientes de diferentes faixas etárias, para poder acompanhar com maior profundidade a eficácia da solução e confiabilidade dos resultados. De todo modo, acredita-se que este requisito foi cumprido.

Conclusão

O dispositivo desenvolvido pode promover avanços na utilização do teste do degrau, já que tende a facilitar a realização do exame. Esse teste, que era realizado somente com auxílio de uma equipe de profissionais, passa agora a poder ser realizado também na própria residência do indivíduo, graças à facilidade e pouca necessidade de acompanhamento médico.

Nesse Trabalho de Formatura, foi possível obter sucesso tanto em incrementar as funcionalidades de um protótipo antigo, deixando-o disponível para ser testado pelo grupo da CUSC, como também no desenvolvimento de um novo protótipo, no qual o objetivo foi trazer um maior nível de inovação.

Analisando os resultados obtidos no teste funcional, é possível concluir que a diferença entre o momento da detecção de erros pela percepção humana e pelo sistema desenvolvido é bem pequena e, possivelmente, está relacionada a erros de medição. O desempenho obtido foi o esperado durante a fase de projeto.

O sistema de detecção de movimentos teve um excelente desempenho, considerando os resultados realizados em indivíduos saudáveis. Possivelmente, no entanto, indivíduos que tenham alguma dificuldade física para realizar o exame apresentarão particularidades na movimentação, que podem atrapalhar o desempenho da solução proposta neste projeto.

Em um âmbito geral, entende-se que o uso da tecnologia pode ajudar a medicina a dar um novo passo evolutivo. A possibilidade atual de registro de dados por meio de *smartphones* e sensores sem fio pode ajudar no diagnóstico e na prevenção de doenças. É justamente essa possibilidade de registrar vários dados do paciente, correlacioná-los e contar, possivelmente, até com tecnologias mais avançadas de reconhecimento de padrões, que traz a esperança de que a medicina evolua nos próximos anos.

O desenvolvimento da tecnologia proposta não está voltado somente a aumentar a eficácia do teste do degrau. Deseja-se também estimular a facilidade para o paciente em realizar um autoexame: já que a análise será automática, isto abre a

possibilidade de que as avaliações sejam feitas na própria residência. Desta forma seria possível contar com uma quantidade maior de dados, pois o indivíduo pode realizar o exame com maior regularidade. Assim, o médico também tem a possibilidade de chegar a um diagnóstico mais preciso, com a recursos até mesmo para fornecer um laudo evolutivo de cada paciente.

Trabalhos futuros

O algoritmo de detecção de movimentos desenvolvido neste trabalho, apesar de funcional, poderia ser melhorado. Para saber se o paciente subiu ou desceu do degrau, foi suposto que, no início do exame, o indivíduo se encontrava fora do degrau. Assim, quando ele realiza o primeiro movimento, entende-se que ele subiu no degrau e assim por diante.

Essa técnica de análise funciona se o usuário não realizar qualquer movimento indevido, ou seja, se o paciente não fazer movimentos que não sejam exclusivamente subir ou desceu do degrau. Acredita-se ser possível resolver este problema fazendo uso do próprio sinal de aceleração.

Em um trabalho futuro, também pretende-se concentrar esforços em realizar testes com pacientes reais, com alguma patologia que, possivelmente, influencie os movimentos no teste do degrau. Analisar movimentos irregulares, portanto, permitirá avaliar se a solução proposta mantém o mesmo desempenho encontrado até o momento, ou se é preciso desenvolver melhorias neste sentido.

Bibliografia

- 1 – Peregrino, A.K.E. **Aplicação do teste de degrau: revisão integrativa**. 7ª jornada científica e tecnológica do Instituto Federal do Sul de Minas, Poços de caldas – MG. 2015.
- 2 – Alves, V.L.S. **Correlação entre os valores obtidos no teste do degrau e os dados espirométricos em adolescentes com escoliose idiopática**. Santa Casa de São Paulo. 2015.
- 3 – Marrara, K.T. **Responsividade do teste do degrau de seis minutos a um programa de treinamento físico em pacientes com DPOC**. UFSCar – SP. 2012.
- 4 – Pessoa, B.V. **Teste do degrau e teste da cadeira: comportamento das respostas metabólico-ventilatórias e cardiovasculares na DPOC**. UFSCar –SP. 2012.
- 5 – Android. **Guia e documentações**. Disponível em: <developer.android.com>. Acesso em: 4 dez. 2017.