

## APLICATIVO DE AQUISIÇÃO DE DADOS DE UM BAROPODÔMETRO VIA *BLUETOOTH* PARA O SISTEMA ANDROID

M.I. Nobukuni\*, A. A. de Carvalho\*, E. R. M. D. Machado\*\*, M. F. R. Urban\*, M. A. A. Sanches\*, L. H.V. Felão\* e J. V. Paulo\*

\* Departamento de Engenharia Elétrica/FEIS – UNESP, Ilha Solteira, Brasil

\*\* Departamento de Matemática/FEIS – UNESP, Ilha Solteira, Brasil

e-mail: mariainezinfo@gmail.com

**Resumo:** Este trabalho apresenta um aplicativo para o sistema Android que realiza a aquisição e armazenamento dos dados em um dispositivo móvel. O aplicativo foi implementado com a linguagem de programação orientada a objetos Java no ambiente de desenvolvimento Eclipse. A transmissão dos dados entre o microcontrolador PSoC (*Programmable System on Chip*) e um *tablet* foi realizada com tecnologia *Bluetooth*. Com o aplicativo serão realizados os testes necessários de mensuração da distribuição de força na região plantar de pacientes hígidos e com escoliose.

**Palavras-chave:** Baropodômetro, PSoC, Java-Android, Bluetooth, Escoliose.

**Abstract:** *This paper presents an application for the Android system that performs the acquisition and storage of data on a mobile device. The application was implemented with language oriented programming Java objects in the Eclipse development environment. The transmission of data between the microcontroller PSoC (Programmable System on Chip) and a tablet was accomplished with Bluetooth technology. With the application the necessary tests to measure the distribution of force in the plantar region of healthy patients with scoliosis and will be realized.*

**Keywords:** *Baropodometer, PSoC, Java, Android, Bluetooth, Scoliosis.*

### Introdução

A baropodometria é um método utilizado por profissionais da saúde para mensurar a distribuição da pressão da região plantar utilizando uma plataforma eletrônica de apoio, denominada baropodômetro. Um aspecto importante a ser considerado é a apresentação dos valores das pressões mensuradas para o usuário.

O desenvolvimento de aplicativos com uma interface amigável é relevante. Caso os aplicativos sejam implementados para dispositivos móveis, como os *tablets*, com comunicação *Bluetooth*, haverá maior facilidade na utilização do equipamento.

Vários tipos de *tablets* utilizam o sistema operacional Android. A vantagem desse sistema é que ele possui código-fonte aberto e gratuito. Em geral, os

aplicativos desenvolvidos para o sistema Android são implementados com a linguagem de programação Java, que também possui código-fonte aberto, é orientada a objetos e tem acesso a bibliotecas de classe que ajudam a desenvolver aplicativos de forma eficiente [1]. Para a implementação dos aplicativos no sistema Android é necessário a utilização de um ambiente de desenvolvimento. O *Integrated Development Environment* (IDE) Eclipse é um ambiente de desenvolvimento que inclui um conjunto de ferramentas denominado *Software Development Kit* (SDK), que incluem um emulador capaz de realizar simulações de dispositivos móveis e uma biblioteca completa para a linguagem Java, com todas as classes necessárias para desenvolver as aplicações [2].

Neste trabalho é descrita a implementação de um aplicativo em Java que realiza a transmissão de dados dos sensores de um baropodômetro para um *tablet* via *Bluetooth*. Além da transmissão de informações, o aplicativo é capaz de realizar o armazenamento dos dados obtidos e a apresentação dos mesmos em uma interface amigável de modo a facilitar a interpretação das medidas de pressão pelo usuário.

### Materiais e métodos

**O baropodômetro** - O baropodômetro utilizado foi implementado por Freitas [3] e Estremote [4], no Laboratório de Instrumentação e Engenharia Biomédica (LIEB) do Departamento de Engenharia Elétrica da Faculdade de Engenharia de Ilha Solteira – SP, da Unesp.

Na Figura 1 apresenta-se uma foto do baropodômetro que possui 48 células de carga com extensômetros resistivos, dispostos em duas plataformas.



Figura 1. Baropodômetro com 48 sensores.

Na Figura 2 é apresentado o diagrama de blocos do sistema de aquisição, transmissão e armazenamento de sinais. O sistema é constituído pelo baropodômetro, por circuito de condicionamento de sinais, por um circuito de aquisição de dados e interfaceamento, por um *tablet*, por um banco de dados e pelo o aplicativo implementado. Empregou-se comunicação *Bluetooth*.

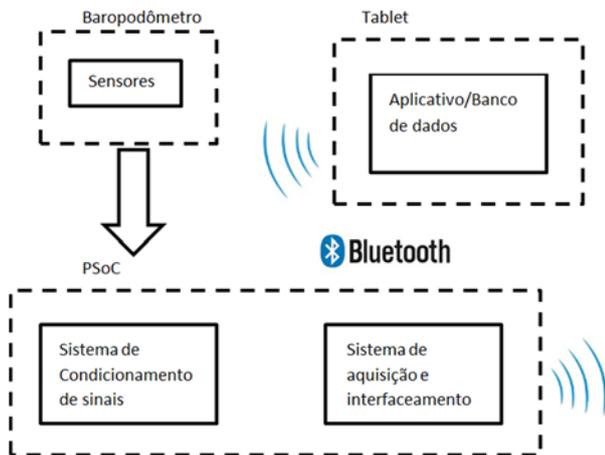


Figura 2. Diagrama de bloco do sistema.

### O sistema de interface com o usuário

O aplicativo contém as seguintes funções:

- Cadastro de dados dos voluntários de forma a facilitar o acesso aos dados armazenados;
- Aquisição dos dados dos sensores do baropodômetro via *Bluetooth*;
- Armazenamento de dados em um banco de dados SQLite;
- Apresentação dos dados na tela do *tablet*.

Na implementação do aplicativo foi utilizado o ambiente de desenvolvimento Eclipse com linguagem de programação orientada a objetos Java. Também foram utilizados o *Software Development Kit* (SDK) do Android, que fornece um conjunto de ferramentas necessárias para desenvolver, testar e depurar aplicativos Android, e o *Plugin Android Development Tools* (ADT), que permite utilizar as ferramentas do SDK do Android para desenvolver aplicativos Android no IDE Eclipse [1].

A seguir serão apresentadas as telas em XML do aplicativo desenvolvido e suas respectivas classes *Activities*. Cada tela em XML possui uma classe *Activity* para realizar as tarefas referentes à tela, como as ações dos botões e cálculos.

Na Figura 3 é apresentada a tela principal do aplicativo. Essa tela disponibiliza ao usuário acesso à tela de cadastro e de pesquisa de voluntários, através de ícones e também possibilita fechar o aplicativo. Os ícones da tela principal são programados através da classe *PrincipalActivity*, que define todos os componentes utilizados para construir a tela em XML, como a definição da imagem logotipo e os ícones, sempre referenciando-os através de seus nomes.

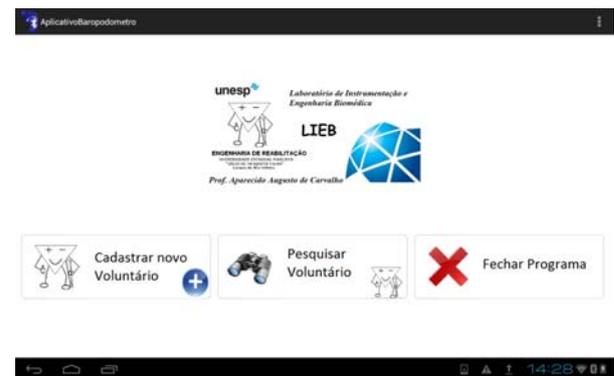


Figura 3. Tela principal do aplicativo.

A tela de cadastro de voluntários apresentada na Figura 4. Nesta tela, o usuário deve informar todos os dados de identificação do voluntário. A tela de cadastro foi programada através da classe *CadastrarActivity*, que define a imagem do logotipo do LIEB, as *editText* (nome, CPF, RG, endereço, cidade, telefone, celular, e-mail, peso, sexo, grau, localização e lado) e também as ações dos ícones cadastrar e voltar.



Figura 4. Tela de cadastro de voluntários.

Na Figura 5 mostra-se a tela de pesquisa de voluntários. Nesta tela são listados todos os voluntários cadastrados no banco de dados do aplicativo. Na lista é apresentado o nome e o telefone do voluntário. Para acessar mais dados do voluntário é necessário selecionar o voluntário desejado. A tela de pesquisa de voluntário

foi programada na classe *ListaVoluntariosActivity*, que define a imagem do logotipo do laboratório, o botão voltar e a *listView*. A *listView* realiza uma conexão com o banco de dados para buscar e listar os voluntários cadastrados.



Figura 5. Tela de pesquisa de voluntários.

A tela de apresentação de dados é acionada quando o usuário seleciona um voluntário da lista. Essa tela foi programada pela classe *ApresentaVoluntariosActivity*, que define o logotipo do LIEB e apresenta o nome, telefone e o e-mail do voluntário selecionado. Para obter dados mais detalhados é preciso acessar a tela de edição através do ícone “Editar Voluntário”. Essa tela também permite excluir o voluntário selecionado e listar os testes desse voluntário. Na Figura 6 é mostrada a tela de apresentação de dados do voluntário.

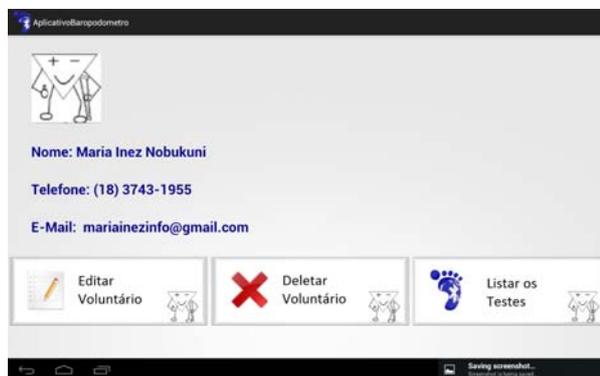


Figura 6. Tela de apresentação de voluntários.

A tela de listar os testes é apresentada na Figura 7. Essa tela é programada pela classe *ListarTestesActivity*, que define o logotipo do LIEB e apresenta o voluntário que possui os testes listados em uma *listView* de acordo com a data do teste que foi realizado. Também define o ícone “Realizar Teste” e aciona a tela de recepção dos sinais.



Figura 7. Tela de lista dos testes do voluntário.

A partir do momento em que essa tela é ativada realiza-se a conexão/emparelhamento com o módulo conversor serial/*Bluetooth*.

Para realizar a transmissão dos 48 sinais do PSoC foi utilizada a frequência de 9600bps porque dentre todas as frequências oferecidas pelo módulo *Bluetooth* é a que apresentou uma menor taxa de erro durante a transmissão dos sinais.

A tela Recepção dos Dados (Figura 8) foi programada com a classe *RecepcaoActivity*, que recebe um *buffer* de dados e converte para *String*. Através de uma função ele separa o *array* de 48 posições para apresentar os dados de forma amigável na tela do *tablet*. Além de realizar a conexão *Bluetooth* e a recepção dos sinais, essa tela permite inserir a data do teste e também apresenta o voluntário que realizou o teste.

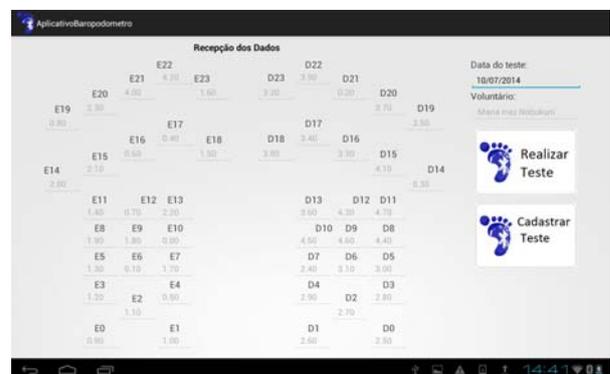


Figura 8. Tela de recepção dos valores dos sensores.

O aplicativo possibilita que sejam armazenados os testes de cada voluntário no banco de dados. Os dados dos testes são armazenados quando o ícone “Cadastrar Teste” é acionado. Esses dados armazenados podem ser acessados a qualquer momento pelo usuário.

A tela de apresentação dos resultados das medições é apresentada na Figura 9. Esta tela foi programada com a classe *ApresentaTesteActivity*. Essa classe define as *editText* dos 48 sinais (D0, D1, D2... D23 e E0, E1, E2... E23) e apresenta a data em que o teste foi realizado.



Figura 9. Tela de apresentação dos valores dos sensores.

O armazenamento dos dados foi realizado com as classes:

- Voluntário, que possui os métodos e atributos dos voluntários e instância o objeto;
- Teste, que possui os métodos e atributos dos testes e instância o objeto;
- A classe *DatabaseHelper*, que cria o banco de dados, gera as tabelas voluntário e teste, e manipula todo o banco de dados, de forma a inserir, editar, pesquisar e excluir dados.

Nas configurações necessárias para a execução do aplicativo para Android são utilizados o arquivo “*AndroidManifest.xml*”. Neste arquivo se configura os componentes da aplicação, como as *Activities* e as permissões necessárias, como o *Bluetooth*.

## Resultados

A transmissão dos dados produzidos pelos sensores e seus respectivos circuitos de condicionamento de sinais foi realizada para o *tablet* via comunicação *Bluetooth*.

Além da transmissão de informações, o aplicativo é capaz de realizar o cadastro e armazenamento das informações de descarga de peso de pacientes e apresentar os dados em uma interface amigável.

## Discussão

O aplicativo desenvolvido possibilitou uma interface amigável que facilita a interpretação dos sinais produzidos pelos sensores do baropodômetro e pode ser reconfigurada para outros dispositivos móveis que utilizam o sistema Android, como os celulares.

Com a transmissão via *Bluetooth* dos dados de descarga de pesos produzidos pelo baropodômetro foi possível eliminar o uso de cabos tornando o sistema mais versátil e portátil, facilitando o uso do equipamento pelo profissional da saúde.

A comunicação está em fase de validação e após a conclusão desta etapa serão realizados os testes com voluntários para coleta de dados.

## Agradecimentos

Agradecemos à CAPES e ao CNPq pelo apoio financeiro.

## Referências

- [1] Deitel, A.; Deitel, H.; Deitel, P.; Morgano, M. *Android para programadores: uma abordagem baseada em aplicativos*. 1. ed. Bookman, 2013. 481 p.
- [2] Burnette, E. *Hello, Android: Introducing Google's Mobile Development Platform*. 1. ed. Pragmatic Bookshelf, 2009. 228 p.
- [3] Freitas, RLB. *Plataforma de força para aplicação biomédica*. 2008. Dissertação (Mestrado) - Departamento de Engenharia Elétrica, Universidade Estadual de São Paulo de Ilha Solteira, Ilha Solteira, 2008. 136 f.
- [4] Estremote, MM. *Medição da descarga de peso na região plantar em pacientes normais e hemiplégicos utilizando plataforma de força*. 2010. Dissertação (Mestrado) - Departamento de Engenharia Elétrica, Universidade Estadual de São Paulo de Ilha Solteira, Ilha Solteira, 2010. 156 f.