

# BIODATA: ARQUITETURA E ORGANIZAÇÃO DE UM SOFTWARE WEB PARA GERENCIAMENTO DE COLETA DE DADOS BIOMÉDICOS

Adilmar. C. Dantas\*, Adriano. O. Andrade\*\*

\*Faculdade de Computação – Universidade Federal de Uberlândia, Uberlândia, Brasil

\*\*Faculdade de Engenharia Elétrica – Universidade Federal de Uberlândia, Uberlândia, Brasil  
e-mail: akanehar@gmail.com

**Resumo:** Este artigo apresenta a arquitetura e organização de um software web capaz de realizar o gerenciamento de coleta de dados biomédicos, tais como o eletrocardiograma (ECG), eletroencefalograma (EEG) ou até mesmo imagens, e arquivos texto contendo informações relevantes. Tais dados são organizados e armazenados, permitindo a realização de backup gerenciável e disponível aos usuários envolvidos na pesquisa por meio da internet, sendo possível recuperar, editar e pesquisar estes dados a qualquer momento, tendo como principal objetivo o desenvolvimento de uma ferramenta livre para uso em grupos científicos.

**Palavras-chave:** Armazenamento de dados biomédicos, Banco de dados, Software para laboratórios.

**Abstract:** *This paper presents the architecture and organization of a web software capable of managing data collection and storage of biomedical data, such as electrocardiogram (ECG), electroencephalogram (EEG) or even images, and text files containing relevant information. The data are organized and stored, allowing for the realization of manageable backup and it is available to users involved in research through the internet, allowing the researcher to retrieve, edit and search these data at any time being able to recover, edit, and search this data at any time, having as main objective the development of a free tool for use in scientific groups.*

**Keywords:** *Storage of biomedical data, database, scientific methodology.*

## Introdução

As pesquisas realizadas na área de Engenharia Biomédica produzem um grande volume de dados, que podem ser representados por séries temporais (e.g., eletrocardiograma, eletroencefalograma e eletromiograma), imagens, vídeos, sons e arquivos texto contendo informações suplementares ou descritivas.

Os dados coletados podem ser úteis tanto no momento da execução de uma pesquisa específica, ou em estudos futuros que podem utilizar estas informações. Um grande problema enfrentado por inúmeros laboratórios de Engenharia Biomédica é o gerenciamento dos dados coletados, de modo que os pesquisadores possam ter

acesso às informações quando necessário.

O acesso aos dados biomédicos de forma organizada tem inúmeras vantagens: (i) a utilização de informações de estudos já realizados para o desenvolvimento de novas pesquisas é ético, pois evita que novos participantes tenham que se envolver desnecessariamente em uma nova pesquisa; (ii) pesquisadores pertencentes a um grupo de trabalho podem compartilhar informações de forma estruturada; (iii) o histórico de pesquisas realizadas em um laboratório é construído e fica resguardado, podendo ser consultando a qualquer instante.

A motivação para a proposta deste trabalho veio da experiência de nosso grupo de pesquisa em Engenharia Biomédica, da Universidade Federal de Uberlândia. Ao longo dos anos acompanhamos inúmeros estudantes finalizando pesquisas de iniciação científica, mestrado e doutorado, e ao término dos trabalhos, geralmente, era bastante difícil recuperarmos e gerenciarmos os dados resultantes de um estudo, principalmente no desenvolvimento de uma nova pesquisa. Ressalta-se que estudos na área de Engenharia Biomédica podem envolver o uso de equipamentos de alto custo e inúmeras horas de trabalhos de técnicos e pesquisadores, o que justifica mais ainda a reutilização de dados.

Seguindo estes levantamentos, desenvolvemos a arquitetura e implementamos um software online para gerenciamento e armazenamento de dados biomédicos.

A arquitetura do sistema, denominado de BIODATA, foi concebida de acordo com a experiência de nosso grupo na execução de protocolos experimentais [1].

## Metodologia

A primeira etapa que realizamos foi o levantamento de requisitos. Nesta etapa é definido o que o sistema deve fazer (e.g., procedimentos e funções). Os requisitos podem ser funcionais, ou seja, aqueles que o sistema deve realizar, ou não funcionais, que estão relacionados aos aspectos acessórios do sistema, como segurança e confiabilidade.

A Tabela 1 descreve os requisitos funcionais e não funcionais do BIODATA.

| Requisito                       | Classificação |
|---------------------------------|---------------|
| Permitir o cadastro de usuários | Funcional     |

|   |               |
|---|---------------|
| Permitir cadastro de protocolo, equipamento, estudos e grupo de estudos | Funcional     |
| Permitir cadastro de sessões de coleta de dados com upload de arquivos  | Funcional     |
| Confiabilidade  | Não Funcional |
| Segurança   | Não Funcional |
| Backup diário   | Não Funcional |

Tabela 1: Requisitos funcionais e não funcionais do BIODATA.

Após a definição de requisitos realizamos a etapa de projetar a arquitetura do software. Primeiramente, fizemos um caso de uso, um documento narrativo que descreve a sequência de eventos de um ator que usa o sistema para completar um processo. Esta representação tem como finalidade demonstrar através de diagramas como será o funcionamento do sistema, contendo todas as classes pertencentes ao sistema e seus devidos métodos. A Figura 1 mostra o diagrama de caso de uso desenvolvido.

Finalizando a construção dos diagramas necessários ao sistema, realizamos a seleção de um banco de dados que fosse capaz de armazenar tais dados de forma consistente e segura. Optamos pelo sistema de gerenciamento de dados (SGBD) MySQL, um sistema de gerenciamento de banco de dados (SGBD), que utiliza a linguagem SQL (Linguagem de Consulta Estruturada, do inglês Structured Query Language) como interface. O MySQL é atualmente um dos bancos de dados mais populares, com mais de 10 milhões de instalações pelo mundo. Entre os usuários do banco de dados MySQL estão: NASA, Friendster, Banco Bradesco, Dataprev, HP, Nokia, Sony, Lufthansa, U.S. Army, U.S. Federal Reserve Bank, Associated Press, Alcatel, Slashdot, Cisco Systems, Google e outros [2].

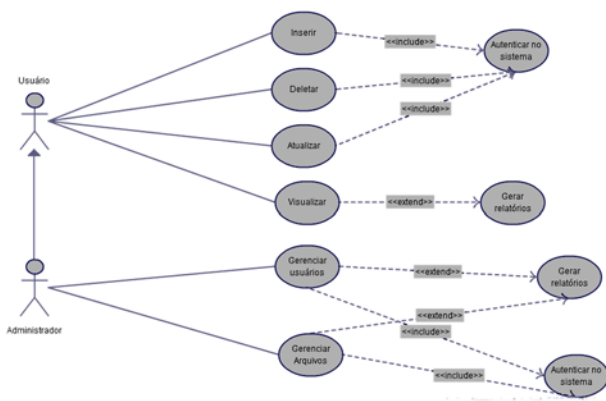


Figura 1: Diagrama de caso de uso.

Com o sistema de banco de dados devidamente definido, a segunda parte realizada foi a construção do diagrama de entidade e relacionamentos (DER), que é um modelo diagramático que descreve o modelo de dados de um sistema com alto nível de abstração [3], com o objetivo de descrever as tabelas necessárias para

o sistema e suas devidas relações. Este modelo pode ser visto na Figura 2.

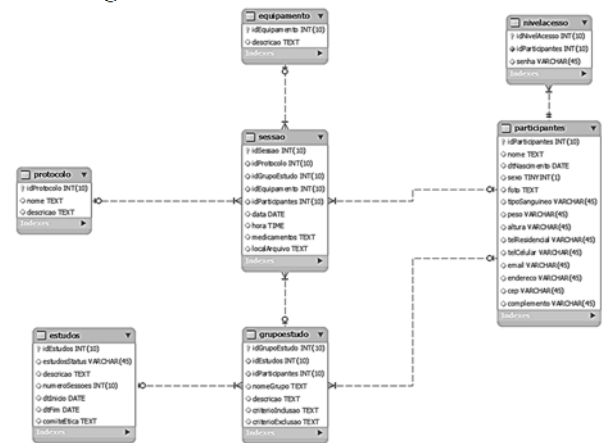


Figura 2: Diagrama de classes

Podemos observar pelo diagrama que o campo local do arquivo é do tipo STRING e não do tipo de imagem, isso se deve ao fato de que nos testes que realizamos o tempo para recuperar as imagens da base de dados era extremamente alto, sendo necessário a aplicação de outras técnicas para reduzir o mesmo. Decidimos por armazenar o diretório do arquivo, quando o usuário realiza upload (carregamento de arquivo) cria-se um diretório específico para cada usuário mantendo assim a organização no servidor e maior velocidade de recuperação destes dados. Para upload de diversos arquivos recomendamos a compactação dos mesmos com a finalidade de não sobrecarregar o sistema e agilizar o processo.

Para codificação do sistema online utilizamos a linguagem de programação PHP, que é uma linguagem interpretada livre, usada originalmente apenas para o desenvolvimento de aplicações presentes e atuantes no lado do servidor, capaz de gerar conteúdo dinâmico na World Wide Web [4]. Procuramos seguir as heurísticas propostas por Jakob Nielsen, sendo que uma delas diz respeito à relação da interface com mundo real. Isto foi necessário para que conseguíssemos interfaces com elementos devidamente ricos em detalhes e dinamismo, e manutenção da estética minimalista. Utilizamos ainda tecnologias RIA - Rich Internet Applicatios (Aplicações ricas para internet), sendo esta uma tecnologia que permite que aplicações web tenham características e funcionalidades de softwares tradicionais do tipo Desktop. RIA típicos transferem todo o processamento da interface para o navegador da internet, porém mantêm a maior parte dos dados (como por exemplo, o estado do programa, dados do banco) no servidor de aplicação [5].

Utilizamos programação orientada a objeto para que o sistema fosse desenvolvido em camadas Model-view-controller (MVC - modelo-visão-controlador). Este é um modelo de arquitetura de software que separa a representação da informação da interação do usuário com ele [6]. Na Figura 3 apresentamos a interface principal do BIODATA.

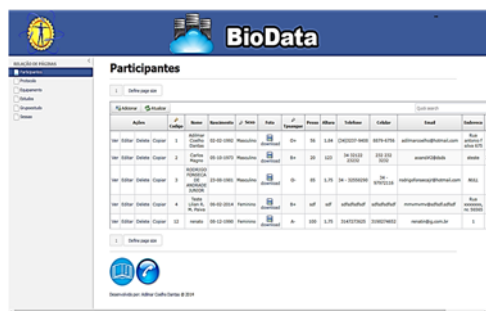


Figura 3: Tela principal do BIODATA.

Para que não fossem necessárias diversas páginas para realizar as operações desejadas pelo usuário, optamos pela tecnologia Ajax, uma vez que o PHP trata os dados de forma assíncrona, ou seja, os procedimentos de leitura e escrita não podem ser realizados simultaneamente. A tecnologia Ajax permite que o acesso a base de dados e a comunicação com o servidor seja feito de forma síncrona e transparente para o usuário, conforme detalhado na Figura 4.

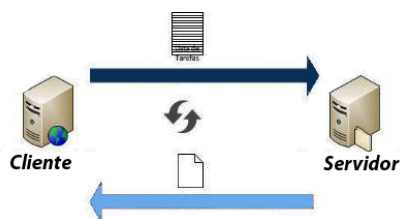


Figura 4: Comunicação cliente - servidor síncrona.

O BIODATA pode ser acessado também de dispositivos móveis graças a implantação do jQuery, que é uma biblioteca de código aberto que facilita a navegação web em diversos dispositivos, sendo usada em diversos sistemas web. Atualmente, o jQuery é a biblioteca mais utilizada do JavaScript [7]. Esta tecnologia torna os sistemas que fazem uso dela um sistema cross-browser que se refere à habilidade de um site, aplicação web, construtor HTML ou script side-client, suportar múltiplos navegadores.

Para garantir a segurança tanto no cadastro quanto na autenticação de usuários, utilizamos a criptografia MD5 que é um algoritmo de hash de 128 bits unidirecionais desenvolvido pela RSA Data Security, Inc., descrito na RFC 1321, e muito utilizado por programas com protocolo ponto-a-ponto (P2P, ou Peer-to-Peer, em inglês) na verificação de integridade de arquivos e logins.

## Resultados

Todos os resultados apresentados nesta seção consideram o uso de um hardware com as configurações discriminadas na Tabela 2. Todo serviço de rede e de largura de banda foi oferecido pela Universidade Federal de Uberlândia.

### Características do hardware

Processador Intel Core i7

4GB Memória RAM

Motherboard ASUS

HD Sata 4TB

HUB 6x

Tabela 2: Especificação do hardware utilizado na avaliação do BIODATA.

A Figura 5 mostra o tempo de resposta do BIODATA em função do número de usuários. Estes foram testes realizados utilizando-se a ferramenta JMeter, uma ferramenta utilizada para testes de carga em serviços oferecidos por sistemas computacionais, sendo parte do projeto Jakarta da Apache Software Foundation [8].

A ferramenta JMeter disponibiliza diversos tipos de requisições e *assertions* (para validar o resultado dessas requisições), além de controladores lógicos como *loops* (ciclos) e controles condicionais para serem utilizados na construção de planos de teste, que correspondem aos testes funcionais.

O JMeter disponibiliza também um controle de threads, chamado Thread Group, no qual é possível configurar o número de threads, a quantidade de vezes que cada thread será executada, e o intervalo entre cada execução, auxiliando assim, na execução dos testes de stress. E por fim, existem diversos listeners, que se baseando nos resultados das requisições ou dos *assertions*, podem ser usados para gerar gráficos e tabelas.

Para realização destes testes, simulamos através do JMeter um grupo de cem usuários acessando simultaneamente e realizando as mais diversas operações CRUD (Create, Read, Update e Delete) na base de dados com a finalidade de verificar a capacidade do sistema de suportar essas requisições, e ao mesmo tempo possível monitorar o nível de stress do servidor sobre situações anormais, podendo assim mensurar sua real capacidade. Podemos observar por meio de testes que o sistema desenvolvido suporta o acesso a cinquenta usuários com facilidade, pois não há grandes variações no tempo de resposta em função do número de usuários *online* no sistema. Esta análise foi possível observando os resultados obtidos através dos testes de simulação no qual consideramos o número de cinquenta usuários simultâneos, como um bom padrão para o funcionamento do sistema. Os testes de utilização e stress de rede demonstram que a mesma também esta preparada para receber tal fluxo de dados. A Figura 6 mostra os resultados encontrados.

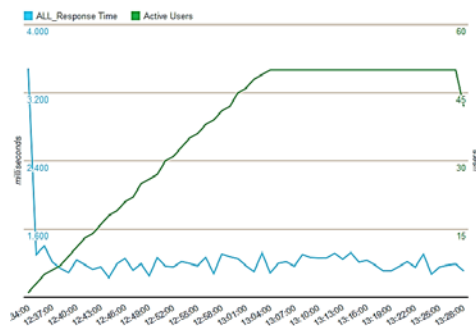


Figura 5: Gráfico tempo de resposta em função do número de usuários.

O aumento da taxa de transferência de dados não produziu variações significativas de sobrecarga à CPU.

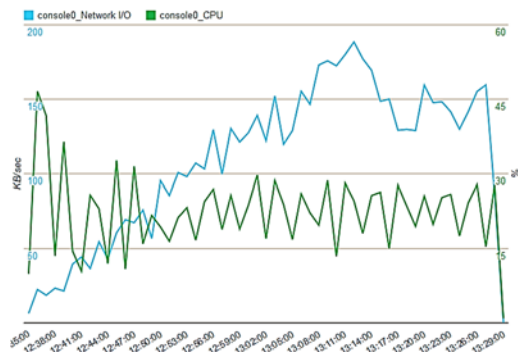


Figura 6: Gráfico do uso de CPU e fluxo de dados de Rede.

## Discussão e Conclusão

Este trabalho reportou o desenvolvimento da arquitetura e organização de um software web para gerenciamento e armazenamento de dados biomédicos.

A principal motivação para o desenvolvimento desta ferramenta foi a necessidade de se organizar e armazenar dados coletados pelo grupo de pesquisa em Engenharia Biomédica da Universidade Federal de Uberlândia. Contudo, acreditamos que esta seja uma necessidade real de inúmeros grupos de pesquisa, justificando assim o compartilhamento de nossa experiência com outros grupos, por meio deste artigo.

Os modelos empregados nas Figuras 1 e 2 foram desenvolvidos de acordo com as principais tarefas que encontramos durante o processo de coleta de dados. Esta organização é genérica de modo a contemplar qualquer tipo de protocolo experimental e dados biomédicos.

Uma vez que o usuário esteja cadastrado no sistema, ele é considerado como um pesquisador e pode gerenciar o processo de coleta de dados por meio das seguintes funções: cadastro de participantes (sujeito de pesquisas); cadastro de equipamentos utilizados em pesquisas; definição de protocolos experimentais, estudos e grupos de estudo; armazenamento de dados durante sessões de coleta de dados. Todas estas funções foram elaboradas de modo a permitir que o sistema seja versátil e genérico, de modo a contemplar os inúmeros tipos de pesquisa que realizamos em nossos laboratórios.

A versão atual do sistema armazena os arquivos obtidos durante a coleta de dados sem considerar o tipo de dados coletado. No futuro o sistema será aprimorado com o intuito de utilizarmos padrões, como por exemplo o *European Data Format* (EDF), para armazenarmos diferentes tipos de informações. Isto nos permitirá a construção de consultas inteligentes, baseada na busca de características inerentes ao tipo de informação. Por exemplo, no caso de séries temporais representando

atividade muscular (eletromiografia), poderíamos fazer buscas por grupos, estudos e participantes que possuam sinais com características em comum. Um exemplo de busca concreta poderia ser o retorno de registros do banco de dados que possuam frequências medianas dentro de um intervalo pré-estabelecido.

Para que consultas, tais a como a descrita acima, possam ser realizadas, é necessário também que o sistema possua inúmeras bibliotecas de processamento e análise de sinais. Certamente, este tipo de busca de informações nos permitirá encontrar informações valiosas que muitas vezes passam despercebidas.

Observamos com a revisão da literatura que existem poucos sistemas com as funções aqui descritas, os softwares existentes com as mesmas funcionalidades são privados e de código fechado, sendo este nosso principal diferencial possibilitando a qualquer grupo de pesquisa fazer suas adaptações de acordo com suas necessidades, e acreditamos que o mesmo possa contribuir para o controle e gerenciamento de dados biomédicos.

## Agradecimentos

Agradecemos as entidades CAPES, FAPEMIG e CNPq por todo apoio financeiro concedido às nossas pesquisas, a Universidade Federal de Uberlândia pelo espaço e equipamentos necessários.

## Referências

- [1] Daniel de Oliveira Araújo, Desenvolvimento de banco de dados para armazenamento de séries temporais biomédicas, Trabalho de Conclusão de Curso, Universidade Federal de Uberlândia, Faculdade de Engenharia Elétrica, Uberlândia, Brasil, 2011
- [2] CARDOSO, Giselle; CARDOSO, Virgínia. **Linguagem em SQL: Fundamentos e Práticas**. São Paulo: Saraiva, 2013. 195 p.
- [3] **DER Banco de dados**, disponível em: <http://www.ime.usp.br/~jef/bd02> Acesso em 05 de Julho 2014.
- [4] Welling, **php e mysql desenvolvimento web**. São Paulo: Campus - grupo elsevier, 2005. 752 p.
- [5] Benefícios das RIAs, disponível em: [https://www.adobe.com/br/resources/business/rich\\_inter\\_net\\_apps/benefits/](https://www.adobe.com/br/resources/business/rich_inter_net_apps/benefits/) Acesso: 01 Junho de 2014.
- [6] MENDES, Douglas. **Programação Java com Ênfase em Orientação a Objetos: Fundamentos e Práticas**. São Paulo: Novatec, 2008. 456 p.
- [7] **jQuery Web**, disponível em: <http://jquery.com/> Acesso em 05 de Julho 2014.
- [8] **JMeter**, disponível em: <http://jmeter.apache.org/> Acesso em 01 de Junho de 2014.