

## DESENVOLVIMENTO DE UM SISTEMA AUTOMATIZADO PARA GERENCIAMENTO DE EVENTOS ACADÊMICOS

**RAFAEL SILVEIRA<sup>1</sup>; BRUNO ZATT<sup>2</sup>; LEOMAR DA ROSA JR.<sup>3</sup>; FELIPE DE  
SOUZA MARQUES<sup>4</sup>**

<sup>1</sup>*Universidade Federal de Pelotas (UFPel) – rsilveira@inf.ufpel.edu.br*

<sup>2</sup>*Universidade Federal de Pelotas (UFPel) – zatt@inf.ufpel.edu.br*

<sup>3</sup>*Universidade Federal de Pelotas (UFPel) – leomarjr@inf.ufpel.edu.br*

<sup>4</sup>*Universidade Federal de Pelotas (UFPel) – felipem@inf.ufpel.edu.br*

### 1. INTRODUÇÃO

É comum que, em eventos acadêmicos, a presença dos participantes seja registrada de alguma forma com a finalidade de controle e emissão de certificados. Normalmente, essa necessidade gera alguns inconvenientes como atrasos, perda do registro e até mesmo burlagem do sistema.

Na Universidade Federal de Pelotas (UFPel), mais precisamente nos cursos de Computação, os registros de presenças em eventos são gerados utilizando um leitor de código de barras, o que, na Semana Acadêmica de 2014, causou filas, atrasos, reclamações e a necessidade de uma pessoa para realizar a tarefa. O que, posteriormente, resultou em um dos motivos da desistência de registro na edição do ano seguinte.

No XXIII Congresso de Iniciação Científica (CIC) e XVI Encontro de Pós-Graduação (ENPOS), onde o controle foi feito via carimbo, houve a necessidade de um organizador por sala para o controle da presença e acesso, necessitando de diversos organizadores especialmente selecionados para essa função. Além dos problemas supracitados, atualmente soluções hardware/software não estão completamente difundidas em Pelotas e região, sendo que, as existentes, são extremamente caras.

Visando solucionar esses problemas, este trabalho apresenta uma solução utilizando tecnologias inovadoras do mercado como: Identificação por radiofrequência (do inglês *Radio-Frequency IDentification* - RFID) (FINKENZELLER, 1999) e comunicação sem fio usando padrão ZigBee (FALUDI, 2010). Para processar essas informações, foi desenvolvido um sistema web e mobile (Android) com tecnologias atuais, tais como: PHP, HTML, CSS, banco de dados MySQL e interface baseada em *Material Design* (GOOGLE, 2015), novo padrão de design desenvolvido pela Google. O sistema tem a função de automatizar todo o processo desde a geração de boleto e processamento da presença ao desenvolvimento do certificado e envio ao participante via e-mail.

### 2. METODOLOGIA

Para facilitar o projeto e desenvolvimento do sistema, o mesmo foi dividido em três camadas com funcionalidades distintas, nomeadas Aplicação, Middleware e Hardware, cujo diagrama de blocos é ilustrado na FIGURA 1.

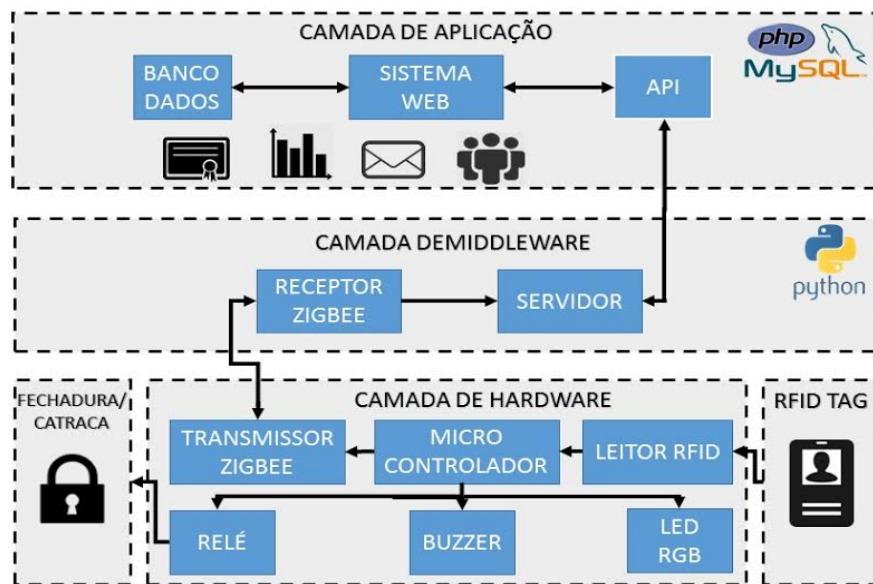


FIGURA 1. Diagrama de blocos do sistema

A **Camada de Aplicação** é a responsável pelos serviços fornecidos pelo sistema, consiste em todo código executado no servidor web, incluindo banco de dados, código do sistema web e API. Foi empregada uma metodologia incremental de desenvolvimento, onde primeiramente foi feita uma análise de requisitos. Decidido os requisitos do sistema, foi então realizada a modelagem UML (Linguagem de Modelagem Unificada) do sistema utilizando os principais diagramas, como: casos de uso e sequência.

Após completada a fase de modelagem, iniciou-se a fase de codificação, foram realizados um total de vinte incrementos de duração de em torno de uma semana cada, onde cada incremento era referente a implementação e teste de uma nova funcionalidade do sistema.

A **Camada de Hardware**, consiste no conjunto de componentes eletrônicos responsáveis pela aquisição dos dados contidos nos crachás RFID. Após concluída as fases de modelagem, desenvolvimento e testes da camada de aplicação, iniciou-se a fase do projeto do hardware, desenvolvimento e testes da placa eletrônica. Então, com o auxílio do software de CAD Eagle, que é uma poderosa aplicação para projeto de circuito impresso e esquemas eletrônicos, foi projetado todo o circuito e realizado testes de consistência do sistema eletrônico.

Enfim, após o período de testes em ferramenta de CAD, iniciou-se a programação do microcontrolador, configuração dos módulos Zigbees, montagem e simulação do protótipo em protoboard.

A **Camada de Middleware** é responsável pelo meio de comunicação entre a camada de Hardware e Aplicação. Nela está inserido o dispositivo receptor Zigbee e software do servidor local. Para o seu funcionamento, é necessário um servidor com sistema operacional Ubuntu conectado à internet, executando um software em linguagem Python, que lê periodicamente entrada USB onde se encontra o dispositivo receptor Zigbee e comunica-se com a API do Sistema Web, além de salvar os dados em um banco local, caso não haja acesso à internet.

### 3. RESULTADOS E DISCUSSÃO

Até o momento, foi desenvolvido um protótipo operacional de todo o sistema. A FIGURA 2. ilustra a tela inicial da aplicação web e tela de informações de uma palestra da aplicação mobile. Na tela do sistema web, no canto superior direito, o usuário pode visualizar informações sobre todas palestras cadastradas no evento e no canto inferior esquerdo, pode visualizar notícias da Computação UFPel. Já na tela do aplicativo mobile (Android), pode ser visualizada informações sobre a palestra, tais como: público presente, horário, link para o currículo Lattes, e-mail do palestrante e número de *likes*. Além disto, uma das dificuldades em eventos é o sorteio de brindes de uma forma eficiente, onde haja uma maior possibilidade do participante sorteado estar presente naquele momento. Por isso, ao ser clicado no botão do canto inferior esquerdo, é sorteado um participante presente da palestra.

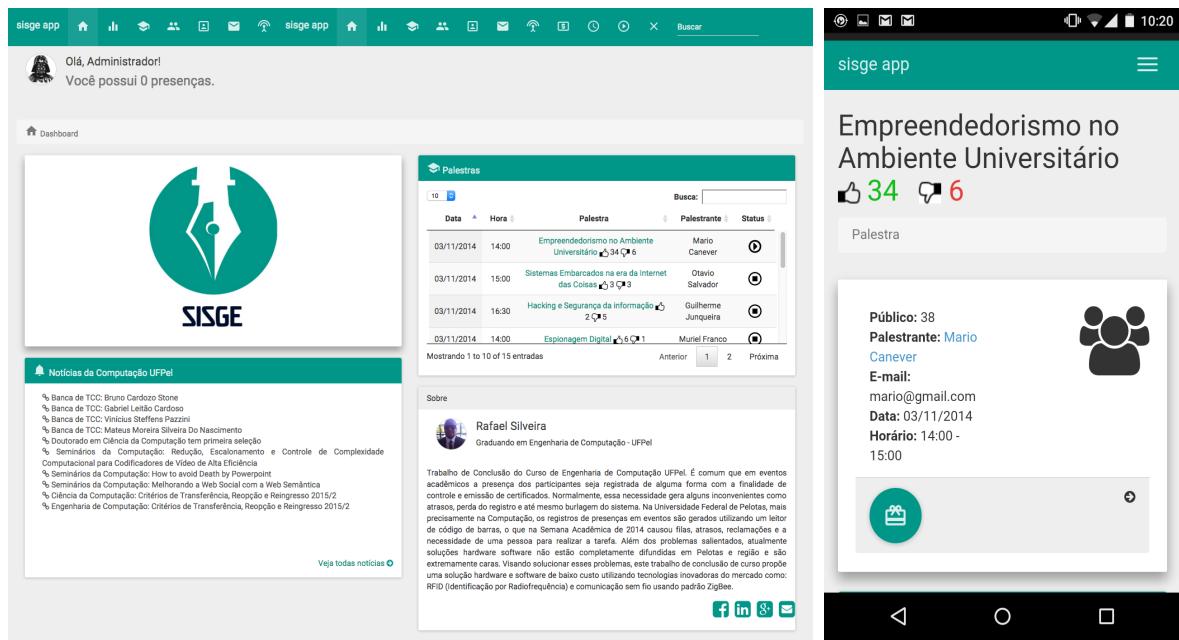


FIGURA 2. Telas do sistema web e mobile (Android)

A FIGURA 3, apresenta a imagem do protótipo de placa eletrônica desenvolvida, que conta com um microcontrolador Atmega 328 PU para processar os dados oriundos da leitura de um cartão (*RFID-Tag*) utilizando um leitor RFID modelo MFRC522. Os dados de identificação são enviados via transmissor ZigBee até um receptor localizado no servidor local, que faz a conexão com a API online, onde após autenticar-se na mesma, obtém acesso ao banco de dados MySQL contendo informações dos participantes previamente cadastrados no evento. Após a comunicação com a API, são retornadas informações como nome do participante, número de presenças e permissões de acesso. Caso o participante não esteja cadastrado no evento, é retornado um aviso. Do ponto de vista de interface com o usuário, é utilizado um display LCD modelo Nokia 5110 para exibir as informações de presenças e nome do participante. Um buzzer emitindo avisos sonoros para usuários com deficiência visual e led RGB que alterna entre as cores vermelha e verde para a melhor compreensão de usuários com deficiência auditiva.

Por fim, um relé eletromecânico é o responsável por permitir o acesso, o qual, eventualmente, poderá ser ligado a uma fechadura eletrônica ou catraca onde será acionado pelo sistema permitindo a entrada do participante.

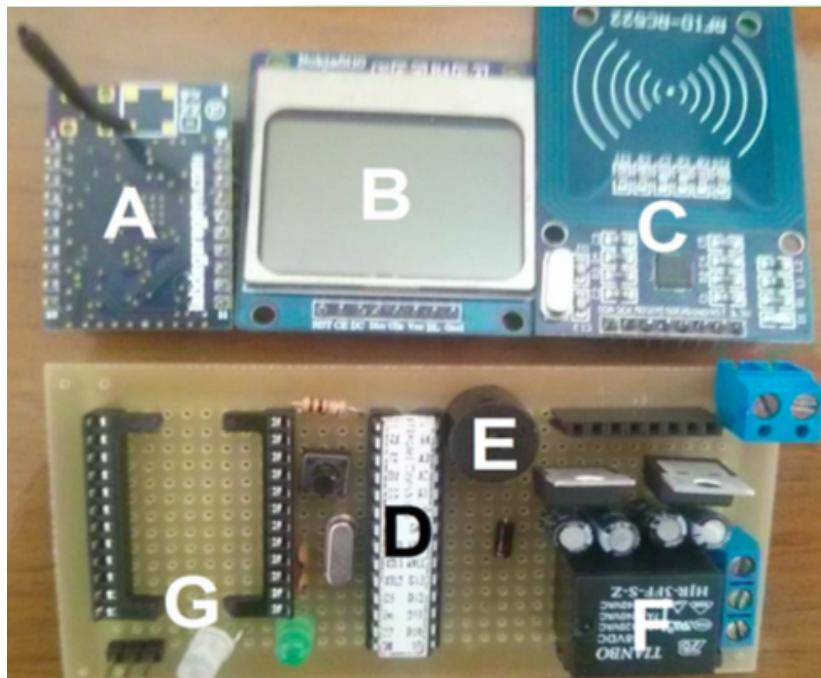


FIGURA 3. A) Transmissor Zigbee; B) Display LCD; C) Leitor RFID; D) Microcontrolador; E) Buzzer; F) Relé; G) Led RGB.

#### 4. CONCLUSÕES

Este trabalho apresentou o Desenvolvimento de um Sistema Automatizado para Gerenciamento de Eventos Acadêmicos. O protótipo do sistema encontra-se em operação e em fase de testes e aperfeiçoamento. Como trabalho futuro, visando a difusão e a ampla utilização pela comunidade acadêmica, está planejado o uso do sistema na Semana Acadêmica da Computação UFPel (SACOMP). Por fim, pretende-se que o sistema seja disponibilizado de forma gratuita, atendendo às necessidades dos cursos da UFPel, facilitando e qualificando o gerenciamento dos eventos da Universidade.

#### 5. REFERÊNCIAS BIBLIOGRÁFICAS

FINKENZELLER, Klaus. **RFID Handbook: Radio-frequency identification fundamentals and applications**. Wiley, 1999.

FALUDI, Robert. **Building wireless sensor networks: with ZigBee, XBee, arduino, and processing**. O'Reilly Media, Inc., 2010.

GOOGLE. **Material Design**. Acessado em 01 jul. 2015. Online. Disponível em: <https://www.google.com/design/spec/material-design/introduction.html>