

Sistema para gerenciamento de eventos

André Nogueira Brandão Guilherme Costa Zanelato
Matheus Pinheiro dos Santos Diego Silva de Carvalho
Breno Bernard Nicolau de França

Relatório Técnico - IC-PFG-17-09

Projeto Final de Graduação

2017 - Dezembro

UNIVERSIDADE ESTADUAL DE CAMPINAS
INSTITUTO DE COMPUTAÇÃO

The contents of this report are the sole responsibility of the authors.
O conteúdo deste relatório é de única responsabilidade dos autores.

Sistema para gerenciamento de eventos

André Nogueira Brandão Guilherme Costa Zanelato
Matheus Pinheiro dos Santos Diego Silva de Carvalho
Breno Bernard Nicolau de França*

Resumo

O problema de gerenciar eventos é de grande relevância para a comunidade acadêmica uma vez que existem diversos tipos de atividades e palestras com necessidades específicas que ocorrem todos os anos nesse meio. Nesse contexto, este trabalho visa desenvolver uma aplicação Web que seja capaz de administrar tais eventos de forma efetiva tanto para os usuários como para seus administradores. Para esse projeto foram utilizadas técnicas e práticas modernas de desenvolvimento ágil com validação das principais partes interessadas.

Sumário

1	Introdução	2
2	Objetivo	2
3	Justificativa	2
4	Metodologia	2
5	Arquitetura	3
5.1	Cliente	3
5.2	Servidor	3
6	Funcionalidades	4
7	Guia de acesso	5
7.1	Usuário	5
7.2	Administrador	5
8	Guia de desenvolvimento	5
8.1	Cliente	5
8.2	Servidor	6
8.3	Instalação	7

*Instituto de Computação, Universidade Estadual de Campinas, 13081-970 Campinas, SP.

1 Introdução

Atualmente, o Instituto de Computação (IC) da UNICAMP realiza eventos acadêmicos de pequeno porte com a finalidade de divulgar os trabalhos desenvolvidos e atender aos interesses dos alunos de graduação e pós-graduação.

A organização destes eventos envolve professores, funcionários e alunos da unidade em esforço conjunto. Entretanto, mesmo se tratando de eventos de baixa escala, o esforço de organização por parte dos envolvidos é considerável e, hoje, não há disponível nenhuma ferramenta de apoio no que diz respeito às características dos eventos realizados. Ainda, não fica nenhum registro digital organizado do evento que apoie na organização de eventos futuros, sobretudo para membros inexperientes da organização. Assim, as pessoas envolvidas na organização ficam sobrecarregadas, sem suporte tecnológico e automatizado para as tarefas de organização (planejamento, divulgação, realização e fechamento) dos eventos, e sem um histórico estruturado de fácil consulta para os eventos anteriores. Sendo assim, o presente trabalho apresenta um sistema capaz de auxiliar em todo o gerenciamento desse processo.

2 Objetivo

Este projeto final de graduação teve por objetivo desenvolver e implantar um sistema para gestão de eventos do IC da UNICAMP. Ele é capaz de gerenciar os eventos e atividades cadastradas além de seus administradores, usuários e palestrantes por meio da plataforma web. O sistema também tem potencial para eventual transferência da tecnologia ou de ser utilizado por outras unidades da universidade e até externas à UNICAMP.

3 Justificativa

Este projeto é justificado, além da necessidade do produto final, pela possibilidade de consolidar as habilidades e conhecimentos em computação adquiridos pelos alunos do curso no desenvolvimento de um projeto passando por todas as etapas do ciclo de vida do software.

Entendemos que esta oportunidade é diferenciada das disciplinas do curso, tendo em vista seu caráter agregador e interdisciplinar com outras áreas do curso. Ainda, é interessante que os alunos tenham uma vivência completa em um projeto real tendo em vista sua formação como profissional para o mercado de trabalho, utilizando tecnologias e metodologias que compõem o estado da prática.

4 Metodologia

De forma a minimizar os riscos e maximizar o feedback para as partes interessadas incluindo a equipe do projeto, foi utilizado um desenvolvimento iterativo e incremental, em ciclos curtos (semanais), com entregas a cada duas semanas que foram avaliadas pelo Prof. Breno.

Os principais stakeholders foram as professoras Esther Colombini, Ariadne Rizzoni, Juliana Borin e alunos que administram a semana da computação de onde foram extraídos requisitos técnicos e funcionais.

Como ferramenta para o gerenciamento do projeto foi adotado o Trello onde foi possível criar as tarefas a partir das histórias, classificá-las e ter um controle sobre sua execução.

O modelo de desenvolvimento utilizou princípios e práticas de métodos ágeis, amplamente utilizados na prática pelas empresas e explorados cientificamente. Em relação às práticas, houve a utilização de:

- Automação de build e testes
- Controle de Versão
- Integração Contínua
- Iterações curtas
- Planejamento de releases

A quantidade de releases (liberações) necessárias se deu em função do escopo do sistema de gestão de eventos, que foi identificado com os stakeholders junto com o Prof. Breno que atuou como facilitador, intervindo em questões que necessitavam de esclarecimentos.

5 Arquitetura

O sistema pode ser dividido em duas frentes, cliente e servidor, que se comunicam por meio de chamadas em uma API RESTful.

5.1 Cliente

Foi desenvolvido como uma aplicação de página única a partir de um template (ng2-admin). O Angular em sua versão 4 é o framework base que se utiliza uma arquitetura baseada em componentes e NPM é o seu gerenciador de pacotes. Além dele estão presentes frameworks auxiliares como Karma, Jasmine e Webpack.

5.2 Servidor

Foi desenvolvido utilizando-se a linguagem Java e o Maven como ferramenta para integração e compilação do projeto além do Spring Boot. Como sistema de gerenciamento de banco de dados foi utilizado o MySQL com mapeamento e manipulação pelo Hibernate.

O design pattern DAO com uma camada para serviços é utilizado para separar o acesso aos dados no banco com os serviços oferecidos pela API.

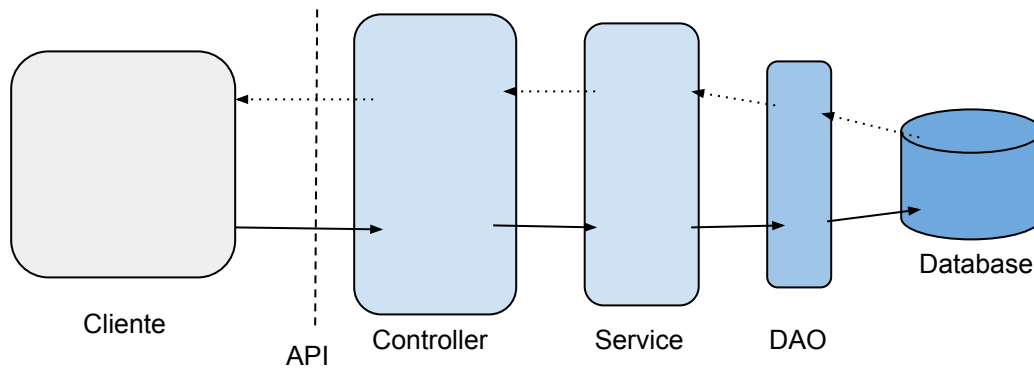


Figura 1: Modelo de organização utilizado no servidor

6 Funcionalidades

Por meio de acesso via browser o administrador é capaz de logar no sistema e realizar as seguintes ações:

- Gerenciar todos os eventos e atividades;
- Criar um novo administrador a partir de e-mail e senha fornecidos;
- Alterar sua senha a partir da senha antiga e nova senha;
- Criar evento com campos de nome, descrição, local, data de realização, período de inscrição, preço de inscrição (se não for gratuito), quantidade de vagas e se há submissão de trabalhos;
- Alterar informações dos eventos;
- Adicionar atividades aos eventos com campos de nome, descrição, local, data, período de realização, período de inscrições opcional, preço (se não for gratuito), quantidade de vagas, chair opcional e palestrantes;
- Alterar as informações de atividades dos eventos;
- Listar inscritos no evento e em cada atividade do evento;
- Adicionar palestrantes com campos de nome completo, e-mail, empresa ou faculdade e descrição;
- Listar e gerenciar os palestrantes de cada evento;
- Listar e gerenciar os trabalhos que foram submetidos no evento;
- Gerenciar a frequência dos usuários cadastrados nas atividades marcando presença e saída;
- Gerar link para o bidding em atividades e gerenciar o bidding;

O usuário comum pode realizar as seguintes ações via browser:

- Verificar os eventos disponíveis além de informações como descrição, programação, datas

de realização, inscrição e detalhes sobre suas atividades;

Registrar-se no sistema por meio de cadastro em um evento com inscrição aberta fornecendo seu nome completo, telefone, e-mail, número de documento e senha;

Cadastrar-se em mais de um evento ou atividade (quando necessária inscrição), após ter feito o cadastro no sistema por meio do cadastro inicial em um evento;

Submeter trabalhos a partir de upload de arquivo e campos descritivos de nome do trabalho, orientador, tipo de aluno, área da computação, tipo da apresentação;

Gerenciar seus trabalhos que foram submetidos após fazer login no sistema;

7 Guia de acesso

7.1 Usuário

Ao acessar o site será exibida a página inicial com a lista de eventos e botão de login. Nela é possível ver datas, valores e se as inscrições estão abertas para os eventos. Ao clicar em um evento, o usuário vai para a tela de detalhes do evento que contém todas as informações relativas ao mesmo além das suas atividades. Se o período de inscrição estiver aberto, o usuário poderá se inscrever no evento e se cadastrar no sistema. Após isso ele poderá submeter trabalhos e se inscrever em atividades que necessitam de cadastro.

7.2 Administrador

Para acessar como administrador do sistema, é necessário realizar o login através da página de login de administrador.

Com isso o administrador é redirecionado para a tela de gerenciamento de eventos onde é possível ver os detalhes e alterar os eventos. Na tela de detalhes ele é capaz de realizar as ações de gerenciamento do evento como listar e adicionar palestrantes, listar trabalhos, listar inscritos e adicionar ou alterar atividades assim como gerenciá-las com controle de frequência dos inscritos e bidding.

Além disso o administrador tem acesso a um menu na lateral esquerda com as funcionalidades de ver o painel de gerenciamento, criar evento, criar administrador e alterar senha.

8 Guia de desenvolvimento

O repositório com todo o código desenvolvido encontra-se em:
<https://gitlab.ic.unicamp.br/ec011/events-system>

8.1 Cliente

Recomenda-se uma IDE com suporte ao Angular e Typescript.

1. Instalar o Node.js

2. Pela linha de comando ir ao diretório principal do cliente:

```
cd admin/ng2-admin/
```

3. Fazer o download das dependências através do comando:

```
npm install
```

4. (Opcional) Fazer o build da aplicação com:

```
npm build
```

5. Fazer o build e rodar a aplicação com:

```
npm start
```

Todos os comandos disponíveis estão no arquivo `package.json`. Após realizar os passos acima pode-se acessar a página no endereço:

```
localhost:4200
```

8.2 Servidor

1. Instalar o pacote Java (SDK)
2. Instalar o Maven
3. Instalar o MySQL
4. Criar uma database com nome `event_system` ou outro (nesse caso alterar nas configurações)
5. Criar um arquivo chamado `application.yml` (este arquivo deve ser excluído do versionamento) no diretório raiz com a seguinte informação (substituindo os valores):

```
spring.datasource.url: jdbc:mysql://localhost:3306/event_system
spring.datasource.username: username
spring.datasource.password: password
DRIVER-CLASS-NAME: com.mysql.jdbc.Driver
HIBERNATE-DIALECT: org.hibernate.dialect.MySQL5Dialect
server.port: 8090
```

6. Fazer o build da aplicação com o comando:

```
mvn clean install
```

ou

```
mvn clean install -DskipTests=true (pular os testes)
```

7. Rodar a aplicação com o comando:

```
mvn spring-boot:run
```

ou

se em uma IDE pode-se rodar a classe `EventManagerApplication`.

Após realizados esses passos, a API pode ser acessada pelo endereço:
localhost:8090/

8.3 Instalação

Para realizar o deployment da aplicação em produção deve-se executar o comando de instalação do maven com o perfil de produção. Utilize o comando:

```
mvn clean install -P prod
```

Após isso, será gerado um arquivo .jar contendo o código compilado capaz de rodar a API do servidor que serve o cliente com conteúdo estático pronto para ser usado. Se as configurações do banco de dados já foram definidas no servidor de produção, basta enviar o arquivo para lá e executá-lo com o comando:

```
java $JAVA_OPTS -jar target/EventManager-0.0.1-SNAPSHOT.jar
```

Se não foram, após ter acesso a uma instância de banco de dados no ambiente de produção, existem 2 opções para configuração, pois nesse ambiente não existe um arquivo application.yml como em desenvolvimento para guardar as informações de acesso ao banco.

- Definir as configurações como variáveis de ambiente no sistema
- Definir diretamente adicionando os parametros por meio da linha de comando ao executar o arquivo com o comando:

```
java $JAVA_OPTS -jar target/EventManager-0.0.1-SNAPSHOT.jar -Dspring.datasource.url=url  
-Dspring.datasource.username=username -Dspring.datasource.password=password
```

No cliente as variáveis de ambiente são configuradas de uma forma diferente. Por padrão o Angular irá usar as configurações em:

```
admin/ng2-admin/src/environments/environment.ts
```

mas no ambiente de produção será usado:

```
admin/ng2-admin/src/environments/environment.prod.ts
```

Referências

- [1] COHN, Mike. User stories applied: For agile software development. Addison-Wesley Professional, 2004.
- [2] DUVALL, Paul M.; MATYAS, Steve; GLOVER, Andrew. Continuous integration: improving software quality and reducing risk. Pearson Education, 2007.

- [3] FOWLER, Martin. Patterns of enterprise application architecture. Addison-Wesley Longman Publishing Co., Inc., 2002.
- [4] SCHWABER, Ken; BEEDLE, Mike. Agile software development with Scrum. Upper Saddle River: Prentice Hall, 2002.
- [5] Node.js: <https://nodejs.org/en/>
- [6] Angular: <https://angular.io>
- [7] NG2-Admin: <http://akveo.github.io/ng2-admin/>
- [8] Spring-Boot: <https://projects.spring.io/spring-boot/>
- [9] Maven: <https://maven.apache.org>