

## **DECISÕES ARQUITETURAIS**

***Sistema de gerenciamento de eventos***

*UFRBeventos*

Laércio Leal, Lanna Vitória

Marina Souza, Nathan Bastos

Orlando Eduardo

## INTRODUÇÃO

Este documento especifica decisões arquiteturais do *Sistema de Gerenciamento de Eventos - UFRBeventos*, proporcionando aos desenvolvedores informações indispensáveis para a construção e implementação do projeto, bem como, para o desenvolvimento dos testes necessários para a ratificação do sistema.

## DESCRIÇÃO GERAL DO SISTEMA

### Abrangência

O *sistema de gerenciamento de eventos - UFRBeventos* tem como seu propósito estabelecer o controle das operações de gerenciamento dos eventos promovidos pela Universidade, tais operações são: inscrição em evento, CRUD de evento, submissão de resumos, submissão de anais e emissão de certificado. Cada operação de gerenciamento recebe dados fornecidos pelo usuário e com base nesses, realizam operações e geram respostas correspondentes.

### Usuários

O presente sistema dispõe de 04 (quatro) tipos diferentes de usuários, estes são:

- Comum: Caracterizado por usuários aos quais não possuem vínculo empregatício ou permissões concedidas pela universidade. Estes podem ou não ser estudantes.
- Orientador: Caracterizado por usuários aos quais são professores/pesquisadores atuantes na universidade.
- Revisor: Caracterizado por usuários aos quais foram encarregados pela universidade as atividades de revisão de resumo e anais de eventos.
- Administrador: Caracterizado por usuários aos quais receberam permissões da universidade para atuarem como responsáveis pelo sistema. Este usuário possui permissão para realizar todas as funções disponíveis. Tais usuários podem ou não ser funcionários do setor administrativo da universidade.

Para realizar o controle de tipos de usuários foi estabelecida o seguinte padrão:

A classe de usuários deve receber um atributo chamado de *accountIdentification*, este deve ser utilizado como identificador de tipo de usuário antes da realização de qualquer operação. A seguir são listados os seus valores e respectivos significados:

- **accountIdentification = 0** representa usuário **administrador**;
- **accountIdentification = 1** representa usuário **revisor**;
- **accountIdentification = 2** representa usuário **orientador**;
- **accountIdentification = 3** representa usuário **comum**;

## TECNOLOGIAS

- **NodeJs:**

Foi decidido que será utilizada para o desenvolvimento back-end por mostrar uma boa performance e aceitação da tecnologia em diversos projetos do mesmo tipo. Além disso, dispõe de uma vasta biblioteca e comunidade o que auxilia no processo de desenvolvimento, melhorando assim a qualidade e diminuindo o tempo de produção.

- **Flutter:**

Foi escolhida esta tecnologia para o desenvolvimento do front-end do software, pois é sabido que o framework é aceito por diferentes sistemas operacionais sem necessitar de muitas mudanças em seu código. Além disso, apresenta um bom desempenho nas suas aplicações, o que o torna a melhor opção no momento.

- **PostgreSQL:**

Foi escolhido como o sistema de banco de dados a ser utilizado, pois permite a utilização de estruturas de dados, tipagens e métodos criados pelos desenvolvedores, desta forma permite que o desenvolvimento e a manutenção do código sejam mais fáceis. Além disso, permite criar funções específicas que otimizam o processamento de informações utilizando os Stored Procedures em um sistema modular.

## REQUISITOS NÃO FUNCIONAIS

### [RNF001] Portabilidade

O sistema deverá apresentar compatibilidade com diferentes sistemas operacionais mobile, estes são: Android, iOS.

**Prioridade:**      ☒ Essencial                      ☐ Importante                      ☐ Desejável

### [RNF002] Alta disponibilidade

O sistema deverá estar disponível em 100% do tempo, de maneira que quando necessário realizar a suspensão temporária dos seus serviços para manutenção ou atualizações, seja definido o momento que gere menor impacto aos usuários do sistema.

**Prioridade:**      ☒ Essencial                      ☐ Importante                      ☐ Desejável

**Obs.:** Será utilizada uma redundância de servidores controlados por um gerenciador de carga, no qual fará a alternância entre os servidores caso algum deles apresente falhas.

### [RNF003] Segurança de armazenamento dos dados

Os dados gerados através do presente sistema deverão ser salvos em uma base de dados segura, e deve-se tentar ao máximo bloquear o acesso de pessoas não autorizadas.

**Prioridade:**      ☒ Essencial                      ☐ Importante                      ☐ Desejável

**Obs.:** Será utilizado a autenticação JWT para consultas do usuário ao servidor, com criptografia MD5 e Base64 das credenciais para solicitação do token e protocolo https para comunicação.

### [RNF004] Confiabilidade durante execução

O sistema deverá ser capaz de lidar com diferentes exceções, se recuperando de comportamentos inesperados sem que haja perda de dados previamente armazenados, nem a necessidade de reiniciar o sistema ou realizar ações paliativas para a restauração do comportamento normal do sistema.

**Prioridade:**      ☒ Essencial                      ☐ Importante                      ☐ Desejável

### [RNF005] Segurança dos dados

O sistema de autenticação de usuário deverá ser criptografado, de modo que apenas usuários com credenciais válidas possuam acesso ao mesmo. Garantindo que os dados armazenados através do sistema estejam devidamente seguros.

**Prioridade:**      ☒ Essencial                      ☐ Importante                      ☐ Desejável

**Obs.:** Será necessário enviar o token de autenticação a cada requisição onde o servidor fará a validação e enviará uma resposta com os dados caso o token seja válido ou retornará uma exceção de usuário não autenticado.

### [RNF006] Escalabilidade

O sistema deve permitir o acesso simultâneo de diferentes usuários sem a perda de qualidade e informações durante a execução. A quantidade prevista de usuários para o sistema é de pelo menos 150 mil.

**Prioridade:**      ☒ Essencial                      ☐ Importante                      ☐ Desejável

**Obs.:** O sistema deve ser desenvolvido utilizando o conceito de processamento dos dados de maneira assíncrona para funções/operações/atividades as quais possam apresentar melhor desempenho, garantindo assim que seja possível atender um maior número de usuários ao mesmo tempo sem que o tráfego de dados seja congestionado.

### [RNF007] Usabilidade

O sistema deverá apresentar uma interface simplificada, de forma que todas as funcionalidades sejam de fácil acesso, proporcionando aos usuários do sistema agradável experiência.

**Prioridade:**      ☒ Essencial                      ☐ Importante                      ☐ Desejável

### [RNF008] Manutenibilidade

O sistema deverá possuir um código que facilite a manutenção do código ao longo do tempo e facilitando a escalabilidade do mesmo.

**Prioridade:**      ☒ Essencial                      ☐ Importante                      ☐ Desejável

## **PADRÕES DE PROJETO**

O sistema será modular e terá seu back-end baseado em microsserviços, facilitando a manutenibilidade e permitindo que a manutenção seja feita em pequenos pedaços sem necessidade de interromper todo o sistema para pequenas atualizações.

O back-end seguirá o padrão de Model-Controller em seus módulos, tendo em vista que esse padrão é amplamente usado e têm demonstrado sucesso nos projetos que utilizam NodeJs.

No aplicativo Flutter será utilizado os conceitos de SOLID, Clean Code e Clean Architecture, com uso extensivo de testes automatizados, para garantir qualidade do código e facilitar a leitura de código para a equipe de manutenção, esses conceitos já vem sendo utilizados e amplamente recomendados pelos desenvolvedores mais experientes nesta tecnologia.

Para mantermos a qualidade do projeto ficou acordado entre os desenvolvedores que será utilizado o package MobX para fazer a gerência de estados das Views que necessitarem de mudanças de estados e o package Flutter Modular para injeções de dependências e navegação entre telas, ambos já foram amplamente testados pela comunidade e apresentam uma excelente qualidade, atendendo nossas demandas com confiabilidade.