

GamaHub

Documento de Arquitetura

Versão 0.1

Histórico de Revisão

Data	Versão	Descrição	Autor(es)
03/11/2023	0.1	Definições e alinhamentos com outros documentos	Rodrigo, Manoel, Mateus, João

Autores:

Matrícula	Nome	Descrição do papel assumido na equipe	% de contribuição ao trabalho (*)¹
170161951	Eduardo Rodrigues Levenhagem		16.5%
211063022	Rodrigo Wendrel Ferreira	Scrum Master	16.5%
190058048	Lara Giuliana Lima dos Santos		16.5%
200023535	Manoel Castro Moura Filho		16.5%
200024825	Mateus Santos Negrini		16.5%
211061940	João Pedro Ferreira Alves		16.5%

¹ (*) – para cada integrante da equipe, considere sua participação tanto no Documento de Arquitetura, quando nos demais documentos já entregues pela equipe (Visão do produto e do projeto; Declaração de escopo) e atribua um, percentual. A soma dos percentuais de todos os integrantes deve fechar em 100%)

Sumário

1 Introdução	5
1.1 Propósito	5
1.2 Escopo	5
2 Representação Arquitetural	6
2.1 Definições	6
2.2 Justifique sua escolha.	6
2.3 Detalhamento	6
2.4 Metas e restrições arquiteturais	6
2.5 Visão de Casos de uso (escopo do produto)	7
2.6 Visão lógica	7
2.7 Visão de Implementação	11
2.8 Visão de Implantação	11
2.9 Restrições adicionais	11
3 Bibliografia	12
4 Anexos	13

1 Introdução

1.1 Propósito

Providenciar uma plataforma para aqueles interessados em participar de eventos ou grupos no FGA possam se encontrar e se comunicar, além de poderem cadastrar seus próprios eventos.

1.2 Escopo

O software se trata de uma aplicação web com o objetivo de permitir a integração dos alunos da Faculdade Gama por meio de atividades sociais que são anunciadas via aplicação.

2 Representação Arquitetural

2.1 Definições

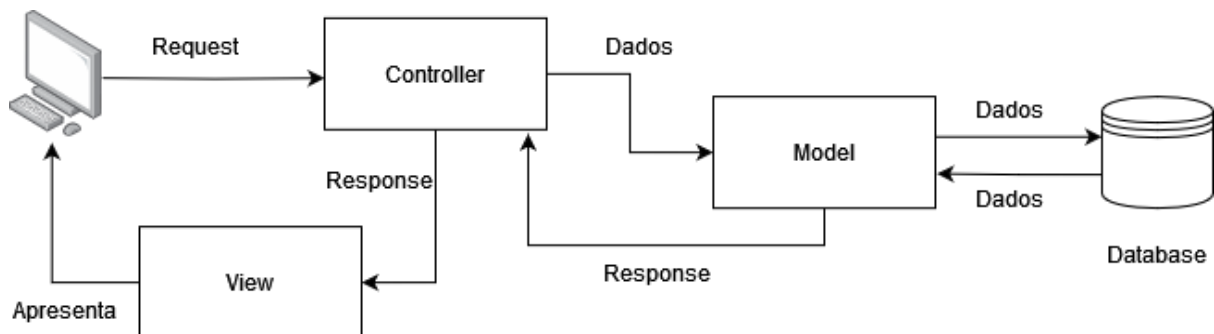
O sistema seguirá uma arquitetura MVC, onde a aplicação é dividida em três camadas: Model, View e Controller. A camada Model (Modelo) é responsável pela manipulação de dados, a camada View é a camada por onde o usuário irá interagir com a aplicação, por fim, a camada Controller é responsável por conectar a Model e a View, repassando as requisições da View para a Model, e repassando os resultados da Model para a View.

2.2 Justifique sua escolha.

Escolhemos o MVC devido a sua modularização permitindo o desenvolvimento simultâneo de diversas partes do software além de oferecer flexibilidade para evoluir a aplicação já que, as alterações na Model não afetam a View e vice-versa, permitindo assim, uma manutenção mais simplificada.

2.3 Detalhamento

Imagem 1: Diagrama de funcionamento da arquitetura MVC



2.4 Metas e restrições arquiteturais

As metas e restrições são definidas para guiar o time de desenvolvimento para entregar um produto de qualidade dentro do escopo. Nas tabelas abaixo estão registradas as metas e restrições assumidas pela equipe no projeto.

Tabela 1: Metas

Metas	Descrição
Atuação	capacidade de 100 usuários simultâneos.
Segurança e confiabilidade	Verificar se um usuário já foi cadastrado.
Segurança e confiabilidade	Não permitir que um usuário faça login em uma conta que não lhe pertence.
Segurança e confiabilidade	Sistema de autenticação de e-mail.
Segurança e confiabilidade	Permitir a exclusão de posts somente por parte do autor ou do administradores.

Manutenção	Cada módulo deve ser independente de modo que seja possível alterar um sem necessariamente alterar toda a aplicação.
Manutenção	O site deve estar disponível para uso 24h por dia.

Tabela 2: Restrições

Restrições	Descrição
Responsividade	A aplicação deve ser adaptável e oferecer uma experiência consistente em todas as variedades de telas de smartphones.
Acessibilidade	A aplicação deve ter utilização intuitiva e simples.
Idioma	A aplicação deve ser desenvolvida para o idioma português do Brasil.
Plataforma	A aplicação será utilizada em navegadores web modernos e atualizados.
Internet	O usuário deverá estar conectado para utilizar a aplicação.

2.5 Visão de Casos de uso (escopo do produto)

A aplicação visa facilitar a criação e divulgação de eventos na Faculdade do Gama, assim gerando maior interação dentro do campus. A aplicação web foi proposta para utilização por três perfis diferentes, o visitante, que acessa o site sem login, o usuário, que utiliza o site logado e o administrador, que possui poderes de curadoria em todo o site. Os visitantes podem visualizar os eventos disponíveis, os usuários podem criar e interagir com eventos, os administradores podem fazer tudo o que os usuários podem, além de banir usuários e deletar eventos ou comentários.

2.6 Visão lógica

O sistema é subdividido nos seguintes módulos: eventos, usuários e comentários. A imagem 2 representa em um diagrama os módulos do software. As imagens 3 e 4 representam os diagramas de estados dos perfis de uso da aplicação.

Imagem 2: Diagrama de Módulos



Imagem 3: Diagrama de Estados(Usuário / Visitante)

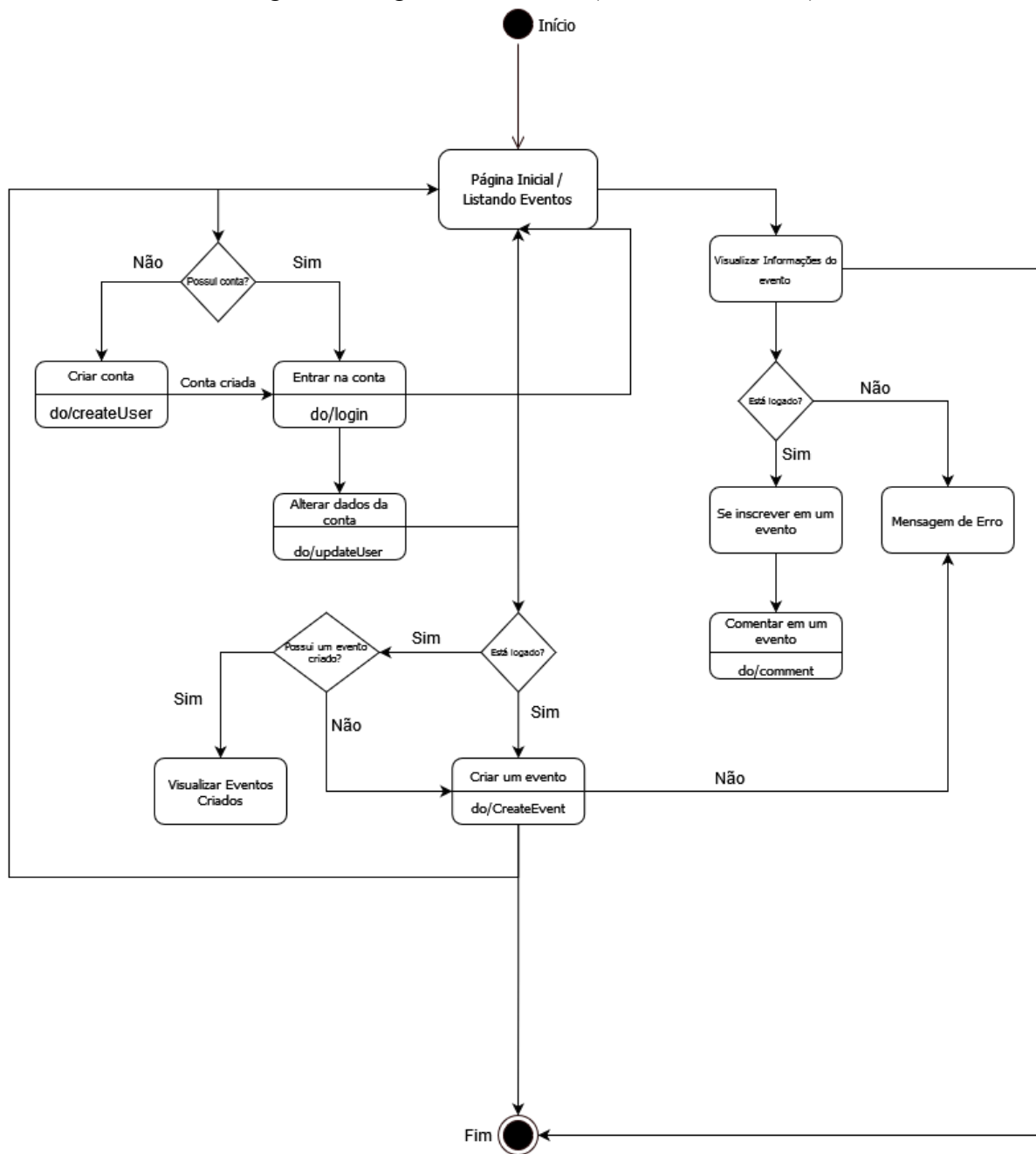


Imagem 4: Diagrama de Estados(Administrador)

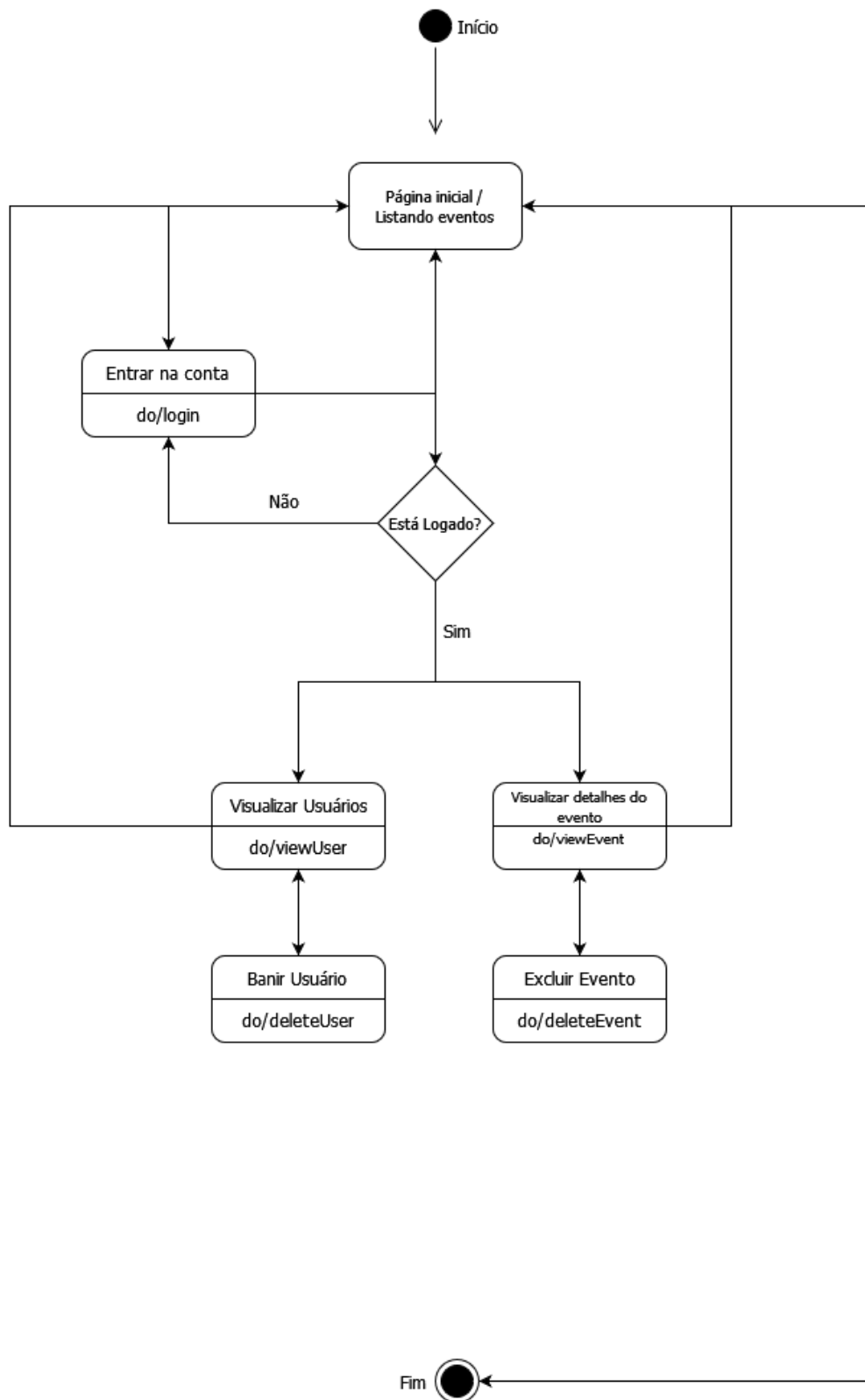
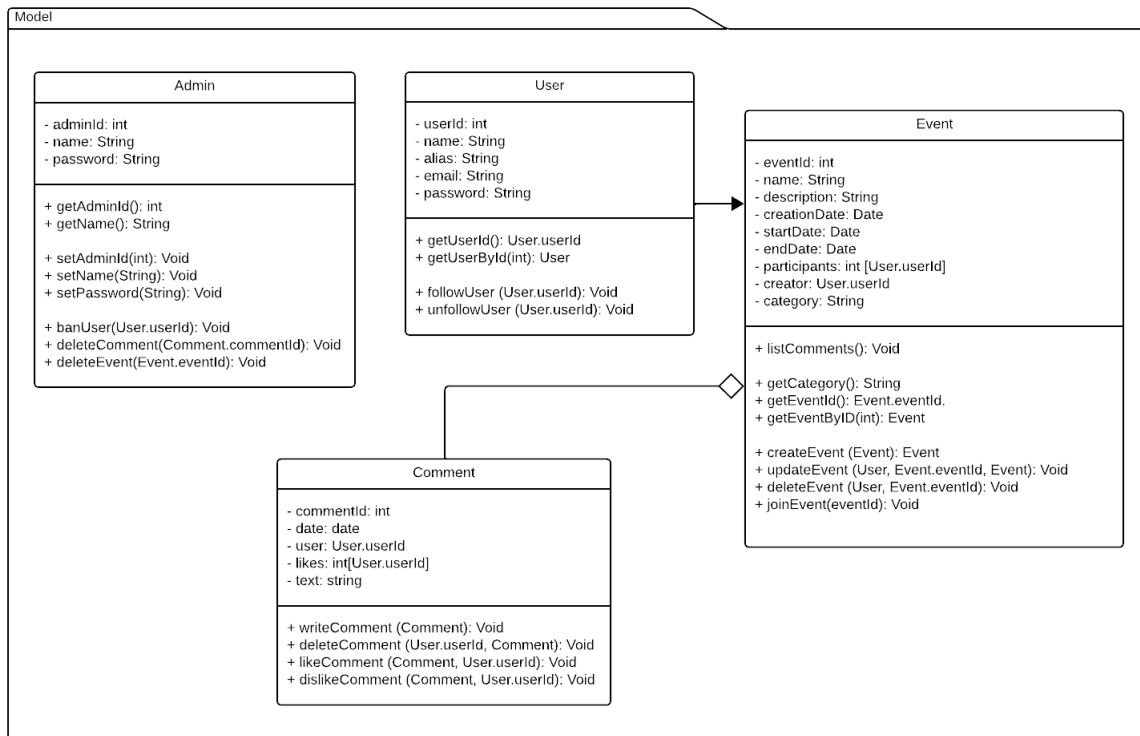


Imagem 5: Diagrama de classes

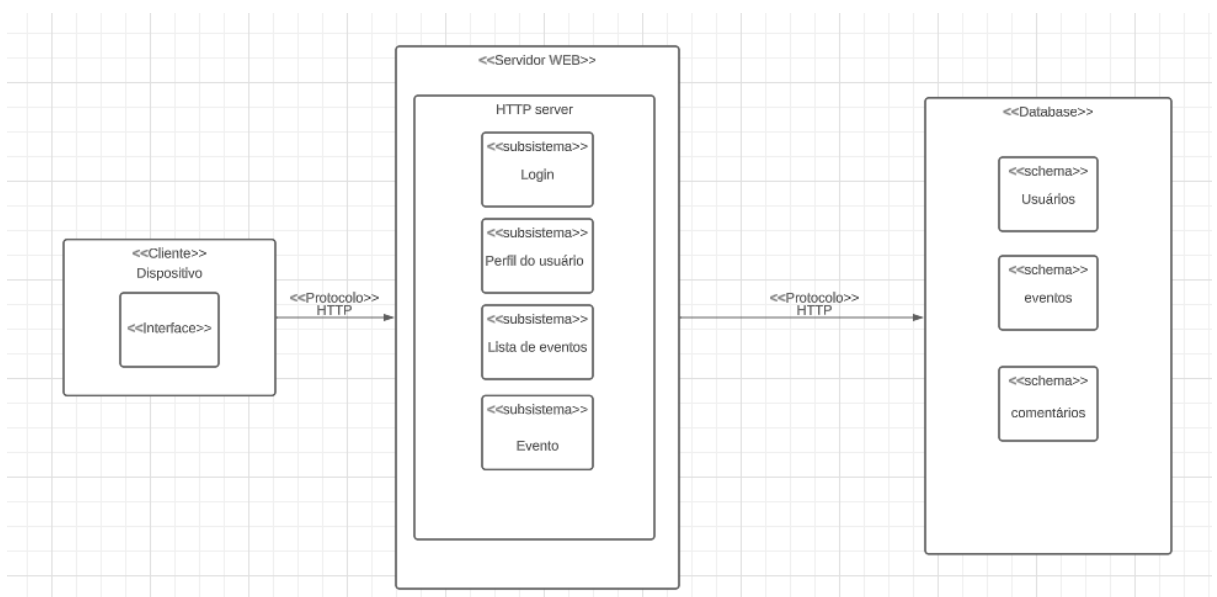


2.7 Visão de Implementação

Tabela 3 - Visão de Implementação

Visão de Implementação - Lógica de negócios e Regras de Negócio
Administradores podem excluir perfis de usuário, postagens no fórum ou eventos que violem os Termos de Uso da aplicação *(Tabela de Regras Gerais).
Cada perfil de usuário terá um login com senha, tendo seu e-mail verificado; Usuários sem um perfil criado poderão apenas visualizar, mas não interagir com os dados da aplicação.
Cada usuário será capaz de fazer postagens no fórum, além de poder cadastrar eventos e endossar comentários com curtidas
O perfil de visitante não terá permissão para fazer postagens nem qualquer tipo de interações, apenas visualizar

2.8 Visão de Implantação



- Front-end React: será implementado o front-end usando React, que será servido por meio de um servidor web Node.js. Será distribuído em servidores na nuvem
- Back-end Node.js: Nosso servidor Node.js atuará como a camada back-end
- Banco de Dados: O MongoDB, por conta de sua simplicidade de uso e aprendizado, será nosso banco de dados principal para armazenar e gerenciar os dados da aplicação web. Utilizaremos um cluster MongoDB gerenciado em um servidor na nuvem para garantir alta disponibilidade

2.9 Restrições adicionais

É necessário o uso de um email da instituição (UnB) para criação de conta e utilização da aplicação.

3 Bibliografia

HIRAMA, Kechi. Engenharia de Software: Qualidade e Produtividade com Tecnologia. Rio de Janeiro: GEN LTC, 2011. 232 p.

KOSCIANSKI, André; SOARES, Michel dos Santos. Qualidade de Software - 2ª edição: Aprenda as metodologias e técnicas. São Paulo: Novatec Editora, 2007. 395 p.

4 Anexos

Tabela de Regras de uso:

Regras Gerais e Diretrizes:

Levando em conta que as atividades serão realizadas utilizando o espaço público da UnB - FGA, algumas orientações são necessárias para garantir que este espaço de interação e convívio cumpra sua função da melhor maneira possível.

- Respeito: O respeito pelos colegas jogadores é fundamental. Qualquer forma de assédio, bullying ou conduta antidesportiva não é tolerada.
- Jogo Justo: Os participantes devem aderir às regras dos jogos e jogar de forma justa.
- Presença: Se você se inscrever para um evento, por favor, compareça ou avise o organizador se não puder comparecer.
- Cuidado com o Equipamento: Trate os equipamentos de jogo e recursos com cuidado. Qualquer dano deve ser reportado imediatamente.
- Rodízio de Turnos: Siga um sistema justo de rodízio de turnos para jogos de tabuleiro e assegure oportunidades iguais para todos os jogadores.
- Pontualidade: Seja pontual para os eventos para evitar atrasos. Aqueles que chegarem atrasados podem ter que esperar pelo próximo jogo.
- Limpeza: Mantenha a área de jogo limpa e organizada. Limpe após jogar e respeite o ambiente do campus.
- Nível de Habilidade: Seja considerado com o nível de habilidade dos outros e ofereça ajuda a iniciantes quando necessário.
- Trapaça: Trapacear ou tentar explorar falhas nas regras é estritamente proibido.
- Controle de Ruído: Mantenha os níveis de ruído razoáveis para evitar perturbar os outros nas proximidades.
- Feedback: O feedback construtivo é encorajado e deve ser compartilhado por meio do sistema de feedback designado