PUC GOIÁS

PROJETO INTEGRADOR MÓDULO 5

DOCUMENTAÇÃO DO SOFTWARE

EVENTOSFERA

Alunos

Marcos Vinicius Pereira Botelho de Souza - 20231012000410

Pedro Paulo Botelho Gama - 20231012000592

Daniel Duarte Mendonça - 20231012000495

Carlos Gabriel Caetano - 20231012000819

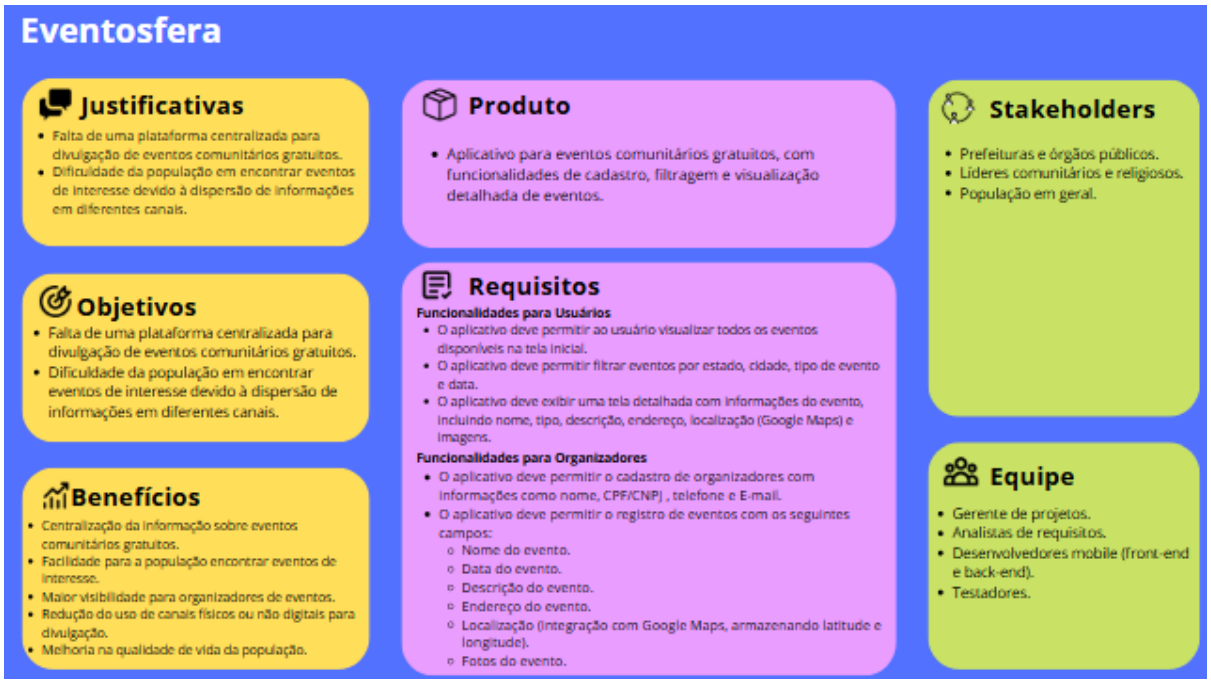
Índice

1. Introdução

1.1 Objetivo

- Nome do Produto: **Eventosfera**
- Este documento tem como objetivo descrever os requisitos do sistema **Eventosfera**, um aplicativo móvel **visualização e gerenciamento de eventos**.
- O **público-alvo** inclui:
    - **Usuários finais** que desejam encontrar eventos próximos.
    - **Organizadores** que precisam cadastrar e gerenciar eventos.
    - **Desenvolvedores e testadores** responsáveis pela implementação.

2. Definição do Escopo



Legenda para cada campo **Project Model Canvas**:

- **Justificativas**
    - Falta de uma plataforma centralizada para divulgação de eventos comunitários gratuitos.

- Dificuldade da população em encontrar eventos de interesse devido à dispersão de informações em diferentes canais.

- **Objetivos**
  - Falta de uma plataforma centralizada para divulgação de eventos comunitários gratuitos.
  - Dificuldade da população em encontrar eventos de interesse devido à dispersão de informações em diferentes canais.

- **Benefícios**
  - Centralização da informação sobre eventos comunitários gratuitos.
  - Facilidade para a população encontrar eventos de interesse.
  - Maior visibilidade para organizadores de eventos.
  - Redução do uso de canais físicos ou não digitais para divulgação.
  - Melhoria na qualidade de vida da população.

- **Produto**
  - Aplicativo para eventos comunitários gratuitos, com funcionalidades de cadastro, filtragem e visualização detalhada de eventos.

- **Requisitos**
  - **Funcionalidades para Usuários**
    - O aplicativo deve permitir ao usuário visualizar todos os eventos disponíveis na tela inicial.
    - O aplicativo deve permitir filtrar eventos por estado, cidade, tipo de evento e data.
    - O aplicativo deve exibir uma tela detalhada com informações do evento, incluindo nome, tipo, descrição, endereço, localização (Google Maps) e imagens.

- **Funcionalidades para Organizadores**
  - O aplicativo deve permitir o cadastro de organizadores com informações como nome, CPF/CNPJ, telefone e E-mail.
  - O aplicativo deve permitir o registro de eventos com os seguintes campos:
    - Nome do evento.

        - Data do evento.
        - Descrição do evento.
        - Endereço do evento.
        - Localização (integração com Google Maps, armazenando latitude e longitude).
        - Fotos do evento.

- **Stakeholders**
    - **Prefeituras e órgãos públicos.**
    - Líderes comunitários e religiosos.
    - População em geral.

- **Equipe**
    - Gerente de projetos.
    - Analistas de requisitos.
    - Desenvolvedores mobile (front-end e back-end).
    - Testadores.

3. Especificação de Requisitos

3.1 Produto

Eventosfera é um aplicativo móvel que permite:

- **Para usuários:**
    - Visualizar eventos em uma lista organizada.
    - Filtrar eventos por localização, tipo e data.
    - Ver detalhes completos, incluindo mapa de localização.
- **Para organizadores:**
    - Cadastrar-se como organizador.
    - Criar, editar e excluir eventos.
    - Gerenciar informações como data, local e descrição.
- **Fora do escopo:**

- Venda de ingressos ou integração com sistemas de pagamento.
- Suporte para iOS (versão inicial apenas para Android).

3.2 Descrição Geral

3.2.1 Requisitos Funcionais

Para Usuários

Código	Descrição	Prioridade
RF01	Exibir lista de eventos na tela inicial com nome, data, local e imagem.	Obrigatório
RF02	Permitir filtros por **estado, cidade, tipo de evento** e **data**.	Desejável
RF03	Mostrar detalhes do evento, incluindo **mapa (Google Maps)**.	Obrigatório

Para Organizadores

Código	Descrição	Prioridade
RF05	Cadastro de organizadores (CPF/CNPJ, e-mail, senha).	Obrigatório
RF06	Cadastrar eventos com **nome, data, local, descrição e imagens**.	Obrigatório
RF07	Editar ou cancelar eventos já cadastrados.	Obrigatório

3.2.2 Requisitos de Interface

Interfaces do Usuário

- **Tela Inicial:** Lista de todos os eventos ativos, contendo uma imagem e com as seguintes informações do evento:

    - Nome do Evento
    - Endereço completo (rua, número, complemento, cep, bairro, cidade, estado)
    - Tipo de evento
    - Nome do Organizador
    - Data do evento
- **Tela de Filtros:** Dropdowns para selecionar:
    - Estado
    - Cidade
    - Tipo de evento
    - Data.
- **Tela de Detalhes:** Exibe todas as informações do evento:
    - Nome do evento
    - Descrição do evento
    - Tipo do evento
    - Data do evento
    - Horário do evento
    - Mapa com localização do evento
    - Galeria de imagens.
- **Telas de Cadastro e Login:** Para organizadores.

3.2.3 Interfaces Externas

- **Google Maps API** (para exibição de localização).
- **Redes Sociais** (compartilhamento via WhatsApp).

3.2.4 Requisitos Não Funcionais

Código	Descrição	Prioridade
RNF01	O aplicativo deve ser **fácil de usar**, mesmo para pessoas com pouca experiência tecnológica.	Obrigatório
RNF02	Deve funcionar em **Android 8.0 (Oreo) ou superior**.	Obrigatório
RNF03	Backend desenvolvido em **Spring Boot + Hibernate**.	Obrigatório

3.2.5 Características dos Usuários

- **Usuários comuns:** Pessoas que buscam eventos.
- **Organizadores:** Promotores de eventos, com conhecimento básico de tecnologia.

3.2.6 Restrições

- **Plataforma:** Apenas Android (versão inicial).
- **Tecnologias:** Frontend **Android** nativo (**Java**) **empty view activity**, Backend em **Java (Spring Boot)**, banco de dados relacional (MySql).
- **Idioma:** português brasileiro.

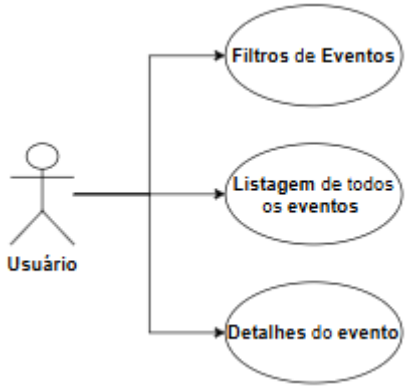
3.2.7 Suposições e Dependências

- **Suposições:**
    - Organizadores fornecerão informações corretas sobre eventos.
- **Dependências:**

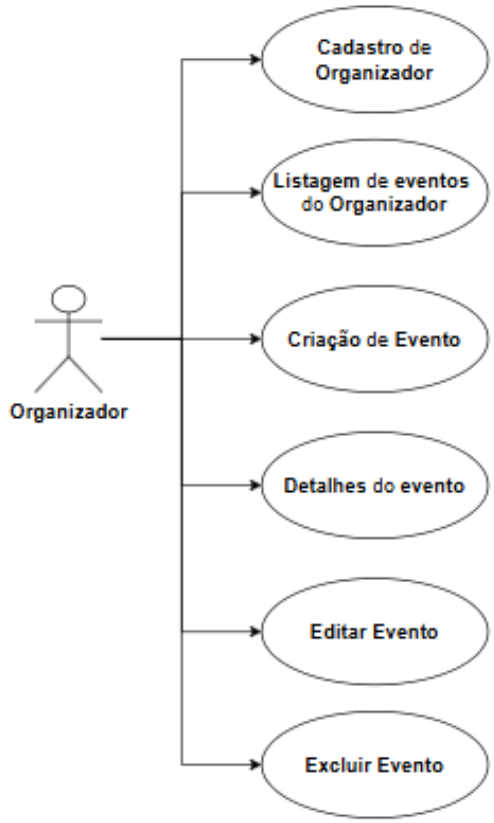
- Disponibilidade da **API do Google Maps**.
- Funcionamento estável do **servidor de backend**.

3.3 Diagrama de casos de Uso

Area do Publica:



Area do Organizador:

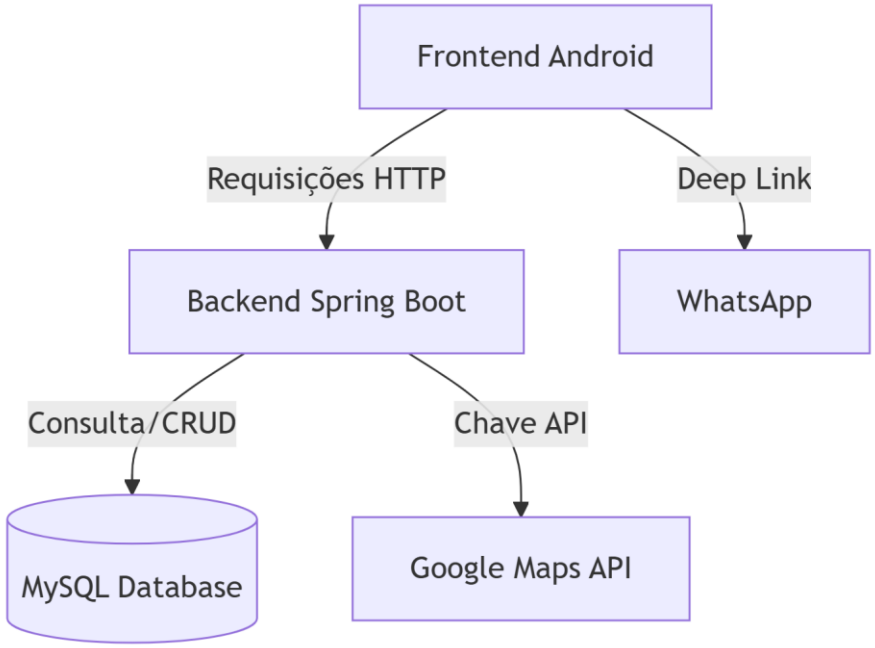


4. Arquitetura de Software

4.1 Tecnologias e Frameworks Utilizados

Componente	Tecnologia/Framework	Detalhes
Frontend	Android Nativo (Java)	Views em XML tradicional
Backend	Spring Boot + Hibernate	Padrão MVC, REST API com JSON, sem autenticação avançada (apenas senha)
Banco de Dados	MySQL	organizer, event, event_image
Integrações	Google Maps API (estático)	Exibição de localização via latitude/longitude
	Deep Link (WhatsApp)	Compartilhamento via intent

4.2 Diagrama de componentes



Fluxo Principal:

- Frontend (Android) faz chamadas REST para o backend.
- Backend processa as requisições (ex: listar eventos) e interage com o banco de dados.
- Para localizações, o backend envia coordenadas (latitude/longitude) ao frontend, que as exibe via Google Maps API.

4.3 Diagrama de camadas (Backend MVC)

Controller: Recebe requisições HTTP e retorna respostas JSON.

Service: Lógica de negócio (ex: validação de eventos).

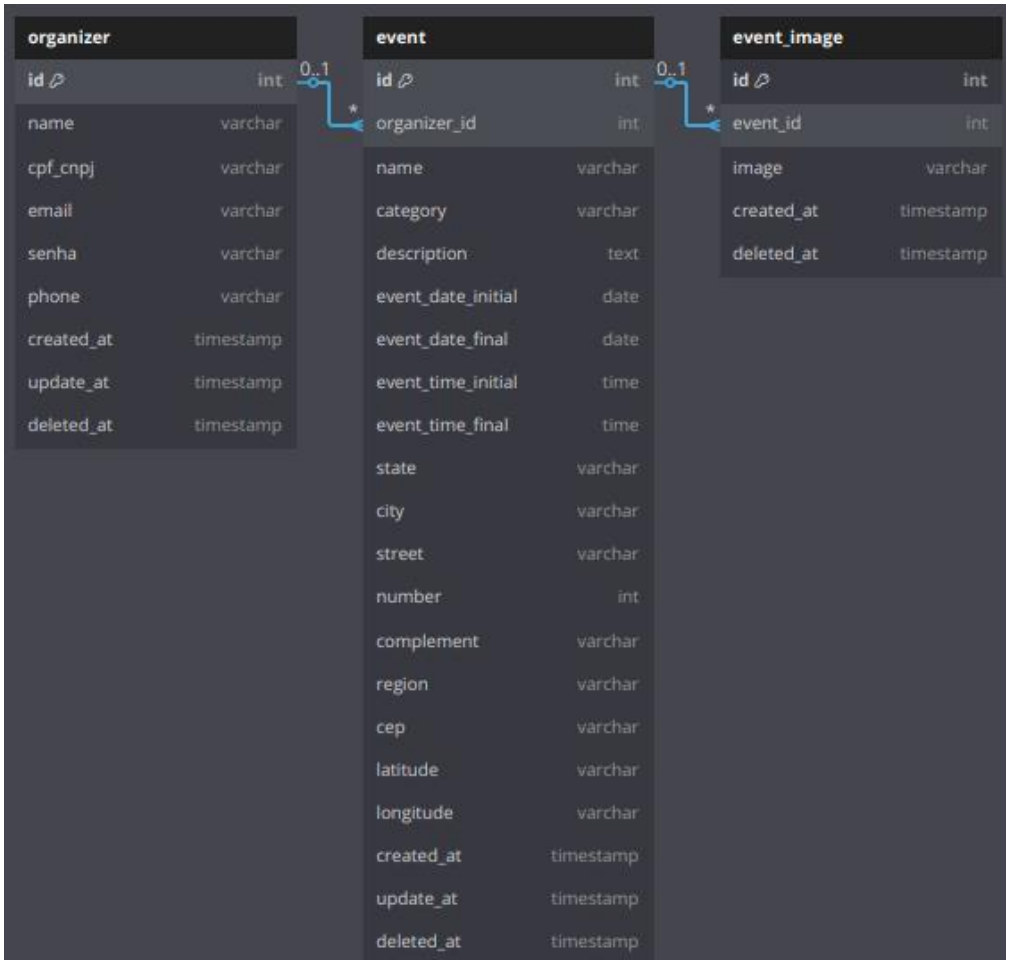
Repository: Acesso ao banco de dados via Hibernate.

4.4 Padrões Adotados

Área	Padrão	Aplicação
Backend	MVC	Separação em Controller-Service-Repository.
Frontend	Tradicional (Java)	Sem padrões
Comunicação	REST API	JSON para requests/responses

5. Projeto do Banco de Dados

5.1 Diagrama Entidade-Relacionamento (DER)



5.2 Descrição das tabelas, chaves e relacionamentos

Tabela **organizer**

Descrição: Armazena dados dos organizadores de eventos.

Nome	Tipo	Descrição	Restrições
id	INT	Chave primária (auto-incremento).	PRIMARY KEY, NOT NULL
name	VARCHAR	Nome do organizador.	NOT NULL
cpf_cnpj	VARCHAR	CPF ou CNPJ (sem criptografia).	NOT NULL, UNIQUE
email	VARCHAR	E-mail de contato.	NOT NULL, UNIQUE
senha	VARCHAR	Senha (texto plano).	NOT NULL
phone	VARCHAR	Telefone para contato.	
created_at	TIMESTAMP	Data de criação do registro.	NOT NULL, DEFAULT CURRENT_TIMESTAMP
updated_at	TIMESTAMP	Data de atualização.	ON UPDATE CURRENT_TIMESTAMP
deleted_at	TIMESTAMP	Data de exclusão (soft delete).	NULL

Relacionamentos:

- 1:N com event (um organizador pode criar vários eventos).

Tabela **event**

Descrição: Contém informações sobre os eventos cadastrados.

Campos:

Nome	Tipo	Descrição	Restrições
id	INT	Chave primária.	PRIMARY KEY, NOT NULL
organizer_id	INT	Chave estrangeira para organizer.	FOREIGN KEY, NOT NULL
name	VARCHAR	Nome do evento.	NOT NULL
category	VARCHAR	Tipo/categoria do evento.	NOT NULL
description	TEXT	Descrição detalhada.	NULL permitido
event_date_initial	DATE	Data de início.	NOT NULL
event_date_final	DATE	Data de término.	NULL permitido
event_time_initial	TIME	Horário de início.	NOT NULL
event_time_final	TIME	Horário de término.	NULL permitido
state	VARCHAR	Estado do evento.	NOT NULL
city	VARCHAR	Cidade do evento.	NOT NULL
street	VARCHAR	Logradouro.	NOT NULL
number	INT	Número do local.	NOT NULL
complement	VARCHAR	Complemento do endereço.	NULL permitido
region	VARCHAR	Bairro.	NOT NULL
cep	VARCHAR	CEP.	NOT NULL
latitude	VARCHAR	Latitude para Google Maps.	NOT NULL
longitude	VARCHAR	Longitude para Google Maps.	NOT NULL
created_at	TIMESTAMP	Data de criação.	DEFAULT CURRENT_TIMESTAMP

updated_at	TIMESTAMP	Data de atualização.	ON UPDATE CURRENT_TIMESTAMP
deleted_at	TIMESTAMP	Soft delete.	NULL

Chaves e Relacionamentos:

- FOREIGN KEY (organizer_id) REFERENCES organizer(id): Vincula o evento ao organizador.
- 1:N com event_image (um evento pode ter várias imagens).

Tabela **event_image**

Descrição: Armazena imagens associadas a eventos.

Campos:

Nome	Tipo	Descrição	Restrições
id	INT	Chave primária.	PRIMARY KEY, NOT NULL
event_id	INT	Chave estrangeira para event.	FOREIGN KEY, NOT NULL
image	VARCHAR	URL ou caminho da imagem.	NOT NULL
created_at	TIMESTAMP	Data de upload.	DEFAULT CURRENT_TIMESTAMP
deleted_at	TIMESTAMP	Soft delete.	NULL

Relacionamentos:

- FOREIGN KEY (event_id) REFERENCES event(id): Cada imagem pertence a um único evento.

6. Descrição do Software Desenvolvido

O **Eventosfera** foi desenvolvido como uma primeira versão funcional (MVP), atendendo aos requisitos essenciais descritos no documento. O aplicativo está operacional para Android com as seguintes características:

- **Estado atual**: Versão inicial concluída, com funcionalidades básicas validadas em ambiente de desenvolvimento.

6.1 Principais desafios técnicos:

- Implementação da **API do Google Maps no frontend**, utilizando o crédito gratuito fornecido pela Google para exibição estática de mapas.
- Estruturação do **soft delete** para eventos e organizadores.

6.2 Apresentação do Aplicativo e Integrações

6.2.1 Frontend (Android Nativo - Java/Kotlin)

- **Telas principais**:
    - **Usuário Público**:
        - **Tela Inicial (Home)**: Exibe cards de eventos com informações básicas (nome, tipo, endereço, organizador e imagem).
        - **Filtros**: Ocultáveis, permitindo busca por nome, estado, cidade e tipo de evento.

        - **Detalhes do Evento**: Mostra todas as informações, mapa estático (via Google Maps API) e galeria de imagens.
    - **Organizador**:
        - **Login/Cadastro**: Campos básicos (email e senha em texto plano – a ser atualizado).
        - **Área Restrita**: Listagem dos eventos do organizador com opções de edição, exclusão (soft delete) e criação de novos eventos.

6.2.2 Backend (Spring Boot + Hibernate)

- **Funcionalidades implementadas**:
    - **CRUD** de eventos e organizadores.
    - Armazenamento das **coordenadas (latitude/longitude)** para uso no frontend.
    - **API REST** para comunicação com o aplicativo Android.

6.2.3 Integrações Externas

- **Google Maps API (Frontend)**:
    - Exibição estática de mapas baseada nas coordenadas armazenadas no banco de dados.
    - Uso do plano gratuito (limite de crédito em dólares para testes).
- **Futuras integrações**:
    - **ViaCEP** para autocompletar endereços no cadastro de eventos.
    - **WhatsApp** (compartilhamento de eventos).

6.3 Fluxo Principal de Uso

- **Usuário Público**:
  - Acessa a **tela inicial** → Aplica filtros (opcional) → Seleciona um evento → Visualiza **detalhes** (mapa, galeria, informações completas).
- **Organizador**:
  - Realiza **login** → Acessa a **área restrita** → Gerencia eventos (cria, edita ou exclui).

6.4 Limitações Conhecidas e Melhorias Futuras

6.4.1 Limitações Atuais

- **Segurança**:
  - Falta de **tokenização (JWT)** e **hash para senhas**.
- **Funcionalidades Pendentes**:
  - **Cancelamento de eventos** (além do soft delete).
  - **Marcador preciso no mapa** (para ajuste manual da localização).
  - **Cadastro de usuários públicos** com favoritos.
- **Validações**:
  - **CPF/CNPJ** não validados no cadastro de organizadores.

6.4.2 Melhorias Planejadas

- **Prioritárias**:
  - Implementar **JWT** e **criptografia de senhas**.
  - Adicionar **compartilhamento via WhatsApp**.
  - **Integração com ViaCEP** para autopreenchimento de endereços.

- **Em Estudo**:
    - Restrição de cadastro apenas para **CNPJ** (para maior controle).
    - **Notificações push** para avisos sobre eventos.