**PROJETO.....**

**Requisitos da disciplina Modelagem de Software e Arquitetura de Sistemas**

**INTEGRANTES DO PROJETO e RA’S**

Adriana Carmem - 11413

São Paulo

2025

Sumário

[**1 INTRODUÇÃO 3**](#_heading=h.lnx34lreujwa)

[**2. DOCUMENTO DE ABERTURA DO PROJETOS 3**](#_heading=h.326qbybq2ym9)

[**3. REQUISITOS DE SISTEMA 4**](#_heading=h.753xxg74xnie)

[3.1 REQUISITOS FUNCIONAIS DE SOFTWARE 4](#_heading=h.q7cgzaizk13k)

[3.2 REQUISITOS NÃO FUNCIONAIS DE SOFTWARE 5](#_heading=h.el7v60llhxfm)

[**4. CASOS DE USO 6**](#_heading=h.6z28qvg41xp4)

[**5. DIAGRAMA DE CLASSE 6**](#_heading=h.5rl2sx532ntr)

[**6. ARQUITETURA DO SISTEMA 6**](#_heading=h.1ck6x8tvewzg)

[6.1. Visão Geral do Sistema 6](#_heading=h.61lqcr1rf4aa)

[6.2. Arquitetura Geral 6](#_heading=h.6aspq58xs9za)

[6.3. Componentes Arquiteturais e Camadas 6](#_heading=h.jww6khxsdd1g)

[6.4. Diagrama da Arquitetura 7](#_heading=h.x4oxh51pher6)

[6.5. Organização de Pastas e Módulos 7](#_heading=h.wze94yf24rbk)

[6.6. Integrações Externas (se houver) 8](#_heading=h.gnl1rzgej0zg)

[6.7. Ambiente e Implantação 8](#_heading=h.xh4zcbrep0cg)

[**7. REFERÊNCIAS BIBLIOGRÁFICAS 8**](#_heading=h.a5a44a3325z6)

# 1 INTRODUÇÃO

Tendo.......

# 2. DOCUMENTO DE ABERTURA DO PROJETOS

**Prefácio**

Deve definir os possíveis leitores do documento e descrever seu histórico de versões, incluindo uma justificativa para a criação de uma nova versão e um resumo das mudanças feitas em cada versão.

**Introdução**

Deve descrever a necessidade para o sistema. Deve descrever brevemente as funções do sistema e explicar como ele vai funcionar com outros sistemas. Também deve descrever como o sistema atende aos objetivos globais de negócio ou estratégicos da organização que encomendou o software.

**Glossário**

Deve definir os termos técnicos usados no documento. Você não deve fazer suposições sobre a experiência ou o conhecimento do leitor.

**Definição de requisitos de usuário**

Deve descrever os serviços fornecidos ao usuário. Os requisitos não funcionais de sistema também devem ser descritos nessa seção. Essa descrição pode usar a linguagem natural, diagramas ou outras notações compreensíveis para os clientes. Normas de produto e processos que devem ser seguidos devem ser especificados.

**Arquitetura do sistema**

Deve apresentar uma visão geral em alto nível da arquitetura do sistema previsto, mostrando a distribuição de funções entre os módulos do sistema. Componentes de arquitetura que são reusados devem ser destacados.

**Especificação de requisitos do sistema**

Deve descrever em detalhes os requisitos funcionais e não funcionais. Se necessário, também podem ser adicionados mais detalhes aos requisitos não funcionais. Interfaces com outros sistemas podem ser definidas.

**Modelos do sistema**

Pode incluir modelos gráficos do sistema que mostram os relacionamentos entre os componentes do sistema, o sistema e seu ambiente. Exemplos de possíveis modelos são modelos de objetos, modelos de fluxo de dados ou modelos semânticos de dados.

**Evolução do sistema**

Deve descrever os pressupostos fundamentais em que o sistema se baseia, bem como quaisquer mudanças previstas, em decorrência da evolução de hardware, de mudanças nas necessidades do usuário etc. Essa seção é útil para projetistas de sistema, pois pode ajudá-los a evitar decisões capazes de restringir possíveis mudanças futuras no sistema.

**Apêndices**

Deve fornecer informações detalhadas e específicas relacionadas à aplicação em desenvolvimento, além de descrições de hardware e banco de dados, por exemplo. Os requisitos de hardware definem as configurações mínimas ideais para o sistema. Requisitos de banco de dados definem a organização lógica dos dados usados pelo sistema e os relacionamentos entre esses dados.

# 3. REQUISITOS DE SISTEMA

## 3.1 REQUISITOS FUNCIONAIS DE SOFTWARE

Necessários 6 requisitos

| **RFS01** | |
| --- | --- |
| **Função** |  |
| **Descrição** |  |
| **Entradas** |  |
| **Fonte** |  |
| **Saídas** |  |
| **Ação** |  |
|  |  |
| **RFS02** | |
| **Função** |  |
| **Descrição** |  |
| **Entradas** |  |
| **Fonte** |  |
| **Saídas** |  |
| **Ação** |  |

## 3.2 REQUISITOS NÃO FUNCIONAIS DE SOFTWARE

Necessários 6 requisitos

| **RFS01** | |
| --- | --- |
| **Função** |  |
| **Descrição** |  |
| **Entradas** |  |
| **Fonte** |  |
| **Saídas** |  |
| **Ação** |  |
|  |  |
| **RFS02** | |
| **Função** |  |
| **Descrição** |  |
| **Entradas** |  |
| **Fonte** |  |
| **Saídas** |  |
| **Ação** |  |

# 4. CASOS DE USO

Apresentar 3 casos de uso do sistema

# 5. DIAGRAMA DE CLASSE

# 6. ARQUITETURA DO SISTEMA

Para a arquitetura do sistema, considere o template abaixo:

**Template (esquema de arquitetura)** para orientar a elaboração de um **Projeto de Arquitetura de Sistema Web**, cobrindo desde a camada de apresentação até a persistência de dados, com foco em modularidade, escalabilidade e eficiência.

### 6.1. Visão Geral do Sistema

* **Nome do Sistema**: Instituto Criativo - Dashboard ONG.
* **Objetivo do Sistema**: (Breve descrição da finalidade do sistema).
* **Público-Alvo**: Beneficiários e colaboradores.
* **Principais Funcionalidades**: Registrar e organizar usuários (beneficiários, colaboradores, doadores…), registro e acompanhamento de eventos, tais como, capacidade de participantes, registro e acompanhamento de projetos, registro e acompanhamento de doações, assim como, módulo de exportação dos dados via excel.

### 6.2. Arquitetura Geral

O projeto foi desenvolvido no modelo MVC.

**Exemplo**: Arquitetura em Camadas (Layered Architecture) com separação clara entre apresentação, lógica de negócios e persistência.

### 6.3. Componentes Arquiteturais e Camadas

**Responsável pela lógica de negócios e acesso a dados:**

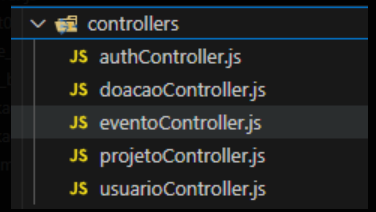
* **Modelos de banco de dados Sequelize**
  + **Usuário – usuários (voluntários, doadores, admin)**
  + **Donation.js – doações realizadas**
  + **Event.js – eventos da ONG**

**Responsável pela interface com o usuário (utilizando React + HTML5/CSS3):**

* **Componentes React (em src/components/):**
  + **Menu – cabeçalho com navegação**
  + **Footer.js – rodapé com contatos e redes sociais**
  + **Private Route - autenticação**

**Controla a lógica entre modelo e visualização (usando Node.js com Express):**

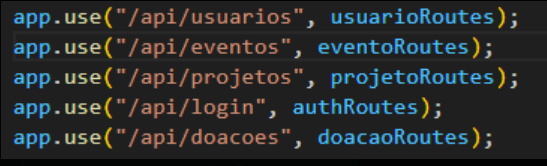
* **Controllers em** controllers/:



### Frontend (React + HTML5/CSS3/JS)

* Páginas públicas:  
  + Página inicial
  + Página de eventos
  + Página de notícias
  + Página de doações
  + Página de projeto
* Área do usuário:  
  + Login/cadastro
  + Dashboard do usuário (visualizar doações feitas, eventos inscritos)
* Área do administrador:  
  + Gerenciar eventos,projetos, doações, usuários

### Backend (Node.js + Express + Bibliotecas de Integração)



* Middleware:  
  + Autenticação
  + Validação de dados (ex: express-validator)
  + Manipulação de erros
* Integrações:

Upload de imagens (Multer)

### Banco de Dados (MySQL)

* Tabelas:

usuários, projetos, eventos e doações.

### Bibliotecas e Ferramentas Comuns

* Frontend:  
  + React Router
  + Axios (requisições à API)
* Backend:  
  + Express.js
  + Sequelize (ORM para MySQL)

#### 6.3.1. Camada de Apresentação (Frontend)

* **Tecnologias**: HTML5/CSS3/JavaScript/MySQL/React/Node.js/bibliotecas de integração.
* **Responsabilidades**:
  + Interface com o usuário;
  + Validação inicial de dados;
  + Comunicação com a API via HTTP/JSON;

#### 6.3.2. Camada de Aplicação (Backend/API)

* **Tecnologias**: Node.js.
* **Responsabilidades**:
  + Implementação da lógica de negócio;
  + Autenticação/autorização;
  + Comunicação com a camada de dados;

#### 6.3.3. Camada de Domínio (Negócio)

* **Descrição**:
  + Contém as regras de negócio e os modelos de domínio (administrador e usuários);
  + Independente das demais camadas.
* **Componentes**:
  + Entidades;
  + Casos de uso;
  + Regras de validação;

#### 6.3.4. Camada de Persistência (Banco de Dados)

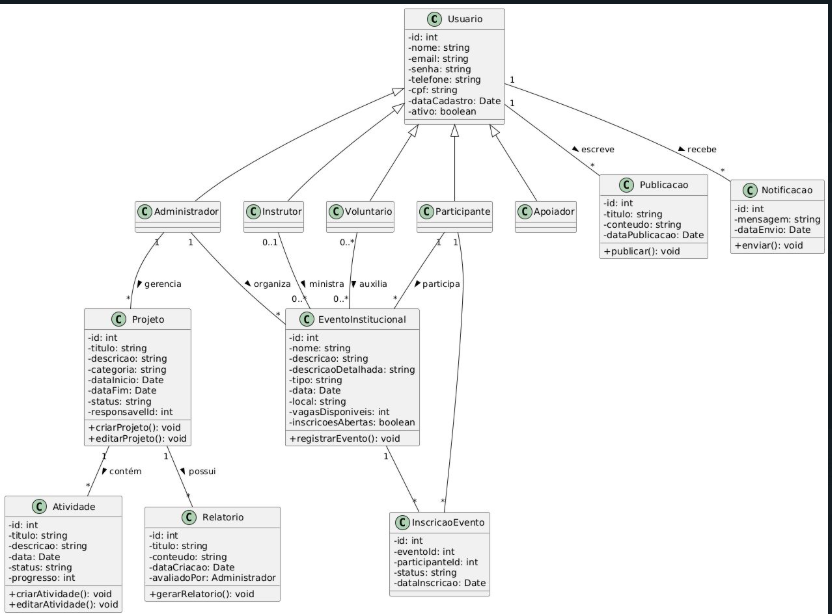
* **Tecnologias**: MySQL.
* **Responsabilidades**:
  + Armazenamento e recuperação de dados
  + Scripts de criação e migração de banco

### 6.4. Diagrama da Arquitetura

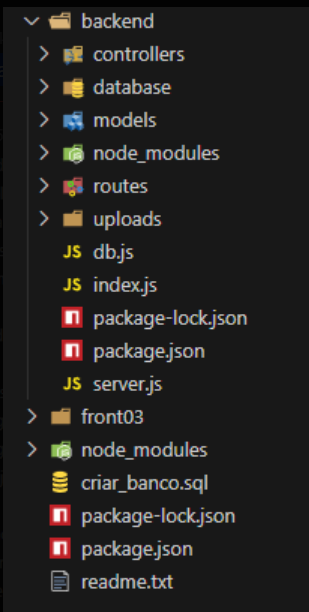
Inserir um **diagrama estrutural** (ex.: em draw.io, Lucidchart ou outro) mostrando:

* Fluxo entre frontend, backend, banco de dados;
* Componentes principais;
* APIs e serviços envolvidos;
* Gateways, autenticação;

Exemplo textual de uma arquitetura de sistema web.



### 6.5. Organização de Pastas e Módulos



### 6.6. Integrações Externas (se houver)

Não tem.

### 6.7. Ambiente e Implantação

* **Ambientes**: desenvolvimento, homologação, produção;
* **CI/CD**: GitHub;
* **Hospedagem**: Azure;

# 7. REFERÊNCIAS BIBLIOGRÁFICAS

SOMMERVILLE, I. **Engenharia de Software.** 11ª Edição. São Paulo: Pearson Addison-Wesley, 2017.

**Silva, Leandro Nunes de Castro.** *Engenharia de Software: Uma Abordagem Profissional*. São Paulo: Pearson, 2013.

**Freeman, Eric; Robson, Elisabeth.** *Use a Cabeça! Padrões de Projetos*. Alta Books, 2007.