FECAP

**PROJETO.....**

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

São Paulo

2024

**INTEGRANTES DO PROJETO e RA’S**

Caio dos Santos Gomes - 24016876

**Sumário**

[**1 INTRODUÇÃO 3**](#_heading=h.30j0zll)

[**2. DOCUMENTO DE ABERTURA DO PROJETOS 3**](#_heading=h.1fob9te)

[**3. REQUISITOS DE SISTEMA 4**](#_heading=h.2et92p0)

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

[3.2 REQUISITOS NÃO FUNCIONAIS DE SOFTWARE 5](#_heading=h.3dy6vkm)

[**4. CASOS DE USO 5**](#_heading=h.1t3h5sf)

[**5. ARQUITETURA DO SISTEMA 6**](#_heading=h.4d34og8)

[**6. REFERÊNCIAS BIBLIOGRÁFICAS 6**](#_heading=h.2s8eyo1)

# 1 INTRODUÇÃO

Tendo.......

# 2. DOCUMENTO DE ABERTURA DO PROJETOS

**Prefácio**

Este documento destina-se a:

* **Desenvolvedores**: Para implementação e manutenção do sistema.
* **Gerentes de Projeto**: Para acompanhamento do escopo e progresso.
* **Administradores do Sistema**: Para suporte técnico e operacional.
* **Stakeholders**: Para compreensão das funcionalidades e objetivos do sistema.

Primeira versão, criada no dia 24/02/2025, código e implementação de desing.

**Introdução**

Instruções de Entrega: Documento de abertura do projeto, Contextualização e

objetivo da solução, Identificação de

Requisitos funcionais e não funcionais.

Descrição: Identificar 6 requisitos

funcionais e 6 não funcionais do sistema.

**Glossário**

Administrador: Usuário responsável por gerenciar os eventos e informações do site  
Conjunto de rotinas e padrões de programação para acesso a um aplicativo ou plataforma. Back-end: Parte do sistema responsável pela lógica de negócios, banco de dados e comunicação com o servidor. Banco de Dados: Estrutura que armazena informações utilizadas pelo sistema.

Ocasião beneficente cadastrada no banco de dados, com informações como data, horário, local e descrição.

Painel de Controle: Interface exclusiva para administradores gerenciarem eventos e informações.

**Definição de requisitos de usuário**

**Para Administradores:**

Login e autenticação segura e criação de novos eventos com as seguintes informações:

* Nome do evento
* Data e horário
* Descrição
* Endereço completo
* Imagem representativa (opcional)
* Contato do organizador
* Edição e exclusão de eventos existentes.
* Painel de controle para gerenciar todos os eventos.

**Para Sub Administradores:**

* Validação de ingressos
* Lista de nomes cadastrados para o evento
* Gerar novos ingressos

**Para Usuários:**

* Visualização do evento
* Abrir fotos em Fullscreen
* Criar login com email e CPF
* Gerar ingresso para o evento selecionado

**Arquitetura do sistema**

O sistema é dividido nos seguintes módulos principais:

**Frontend (Cliente)** – Responsável pela interface com o usuário, onde visitantes visualizam eventos e geram ingressos, enquanto administradores gerenciam os eventos.

**Componentes reutilizáveis:**

* Formulários de login e cadastro.
* Componentes de exibição de eventos e ingressos.
* Módulo de leitura de QR code para validação de ingressos.

**Backend (Servidor)** – Autenticação e comunicação com o banco de dados.

**Componentes reutilizáveis:**

* Serviço de autenticação compartilhado entre usuários.
* Módulo de validação de ingressos.

**Banco de Dados** – Responsável pelo armazenamento de usuários, eventos e ingressos.

* Estruturas otimizadas para consultas eficientes.
* Relacionamentos entre eventos e participantes.

**Fluxo de Dados Simplificado**

1. Visitantes acessam o site e visualizam os eventos.
2. Clientes criam conta e geram ingressos.
3. Administradores gerenciam eventos e usuários.
4. Subadministradores validam ingressos.
5. O banco de dados mantém todas as informações atualizadas.

**Especificação de requisitos do sistema**

**Requisitos Funcionais**

**1.1 Cadastro e Acesso de Usuários**

* O sistema deve permitir que os clientes criem uma conta utilizando nome, e-mail ou CPF e data de nascimento.
* Administradores e subadministradores devem ter acessos diferenciados com permissões específicas.

**1.2 Gerenciamento de Eventos**

* O administrador deve poder criar, editar e excluir eventos.
* Cada evento deve conter título, descrição, data, horário, local e imagens.
* Os eventos devem ser listados para os visitantes do site.

**1.3 Geração e Validação de Ingressos**

* O cliente deve poder gerar um ingresso ao se inscrever em um evento.
* O ingresso será usado principalmente para capturar estatísticas e perfil de clientes participantes do evento
* O ingresso ira ter um identificador único (como código numérico ou QR Code).
* O subadministrador deve poder validar ingressos no local do evento através do sistema, a parte de validação aparecera apenas para os administradores

**1.4 Exibição de Eventos**

* O sistema deve listar eventos futuros disponíveis para inscrição.
* O visitante pode visualizar detalhes do evento sem necessidade de login.
* O login só será exido quando se optar por gerar um ingresso

**2. Requisitos Não Funcionais**

**2.1 Desempenho e Escalabilidade**

* O sistema deve ser capaz de suportar até 500 usuários simultâneos sem degradação de desempenho.
* As consultas ao banco de dados devem ser otimizadas para rápida resposta.

**2.2 Usabilidade**

* A interface deve ser intuitiva e acessível em dispositivos móveis e desktops.
* O tempo de resposta das páginas principais deve ser inferior a 3 segundos.

**2.3 Disponibilidade e Confiabilidade**

* O sistema deve estar disponível pelo menos 99% do tempo.
* Deve haver backup automático dos dados essenciais diariamente.

**2.4 Segurança**

* O sistema deve exigir senhas fortes para usuários administradores.
* As informações do usuário devem ser protegidas contra acessos não autorizados.

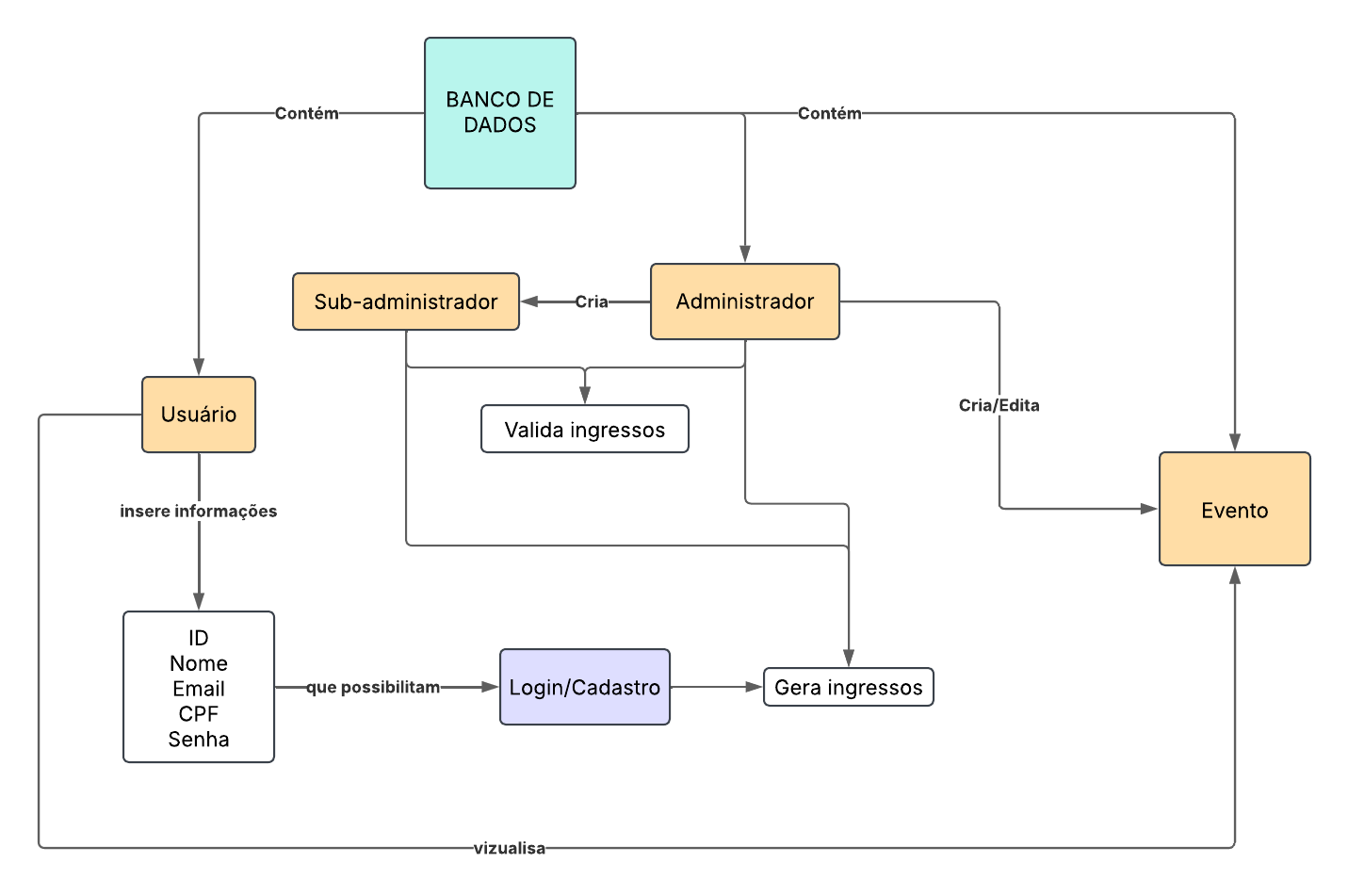
**3. Interfaces com Outros Sistemas**

O sistema pode integrar-se a outras plataformas para ampliar suas funcionalidades:

* **Utilização de Câmera no Mobile:** Para validação de ingressos.
* **Serviço de QR Code**: Para gerar e validar ingressos com mais segurança.
* **Ter como objetivo integrar as redes sociais do Instituto Criativo (META)**

**Modelos do sistema**

O administrador terá acesso a tudo, tanto na edição visual como na de dados. Todos os participantes do sistema podem visualizar.



**Evolução do sistema**

**Pressupostos Fundamentais**

O sistema será projetado com base nos seguintes pressupostos:

* O site terá um **ciclo de vida curto**, sendo utilizado para eventos específicos.
* O sistema funcionará majoritariamente **online**, mas algumas funcionalidades, como a validação de ingressos, podem ser otimizadas para **uso offline**.( Armazenando em disco os ingressos gerados junto com suas informações)
* O banco de dados armazenará apenas **informações essenciais**, evitando sobrecarga e garantindo eficiência.
* A autenticação de usuários será feita de forma **simples e segura**, ( login e senha)

pode evoluir para suportar múltiplos eventos simultâneos, integração com pagamentos, aplicativo mobile e ferramentas de análise. Para garantir flexibilidade, a arquitetura é modular, com banco de dados escalável e APIs padronizadas. No futuro, melhorias na segurança, automação de relatórios e integração com redes sociais podem ser implementadas sem grandes reestruturações.

**Apêndices**

O sistema mínimo para rodar a aplicação é definido pela necessidade de garantir o funcionamento adequado das funcionalidades essenciais, sem sobrecarregar os recursos do ambiente. De acordo com as características da aplicação, as configurações mínimas e ideais para o sistema são as seguintes:

* **Configuração Mínima**:  
  O sistema pode ser executado em um ambiente básico com os seguintes requisitos:
  + **Processador:** Intel Core i3 ou equivalente
  + **Memória RAM:** 4 GB
  + **Armazenamento:** 20-40 GB em HD ou SSD
  + **Sistema Operacional:** Windows 10 ou Linux
* **Configuração Ideal**:  
  Para um desempenho mais eficiente e para suportar um maior número de usuários ou processos simultâneos, a configuração ideal seria:
  + **Processador:** Intel Core i7 ou equivalente
  + **Memória RAM:** 8 GB ou superior
  + **Armazenamento:** 100 GB SSD
  + **Sistema Operacional:** Windows 10 ou Linux (versões de servidor para ambientes maiores)

Esses requisitos são baseados na análise do funcionamento da aplicação e dos recursos necessários para garantir uma operação fluida, sem gargalos de desempenho.

#### **Requisitos de Banco de Dados**

O sistema será apoiado por um banco de dados relacional, projetado para gerenciar dados de forma eficiente e organizada. A escolha do banco de dados e a estrutura dos dados são determinadas de acordo com as necessidades do projeto:

* **Banco de Dados:**  
  O banco de dados a ser utilizado será **MySQL** ou **PostgreSQL**, escolhidos devido à sua robustez e flexibilidade para lidar com operações de inserção, consulta e atualização de dados.
* **Estrutura Lógica dos Dados:**  
  A organização dos dados será feita da seguinte forma:
  + **Tabelas principais:**
    - **Usuários:** id (PK), nome, email, senha
    - **Eventos:** id (PK), nome\_evento, data\_evento, local
    - **Participações:** id (PK), id\_usuario (FK), id\_evento (FK)
  + **Relacionamentos:**
    - **1:N (um para muitos):** Um usuário pode participar de vários eventos.
    - **N:M (muitos para muitos):** Um evento pode ter muitos participantes, e um participante pode se inscrever em vários eventos.
* **Considerações de Normalização:**  
  O banco de dados estará normalizado até a **2ª Forma Normal (2NF)**, garantindo que redundâncias sejam eliminadas e os dados sejam organizados de forma eficiente.

# 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** |  |

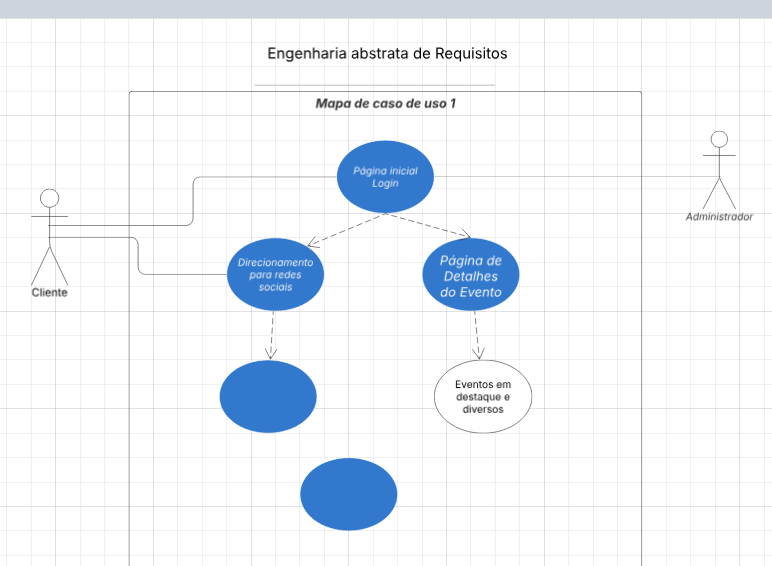
## 3.2 REQUISITOS NÃO FUNCIONAIS DE SOFTWARE

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. ARQUITETURA DO SISTEMA

# 6. REFERÊNCIAS BIBLIOGRÁFICAS

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