

**PONTIFÍCIA UNIVERSIDADE CATÓLICA DE MINAS GERAIS**

**Bacharelado em Sistemas de Informação**

**RELATÓRIO FINAL**

**Trabalho Integrado: Aplicações para Sustentabilidade**  
**+ Projeto de Software**

**CTC Team**

**Autores:**

Ana Beatriz Costa

Grazielle Sorrentino

Gustavo Anselmo

Karina Oliveira

Nicole Marie

# SUMÁRIO

<b>1. Apresentação</b>	<b>3</b>
1.1 Cliente	3
1.2 Problema	3
1.3 Objetivo do trabalho	4
1.4 Definições e Abreviaturas	4
<b>2. Requisitos</b>	<b>6</b>
2.1. Lista de Atores	6
2.2. Lista de Funcionalidades	7
2.3. Requisitos Funcionais	8
2.4. Requisitos Não Funcionais	9
2.5. Descrição Resumida dos Casos de Uso ou Histórias de Usuários	9
2.7. Restrições Arquiteturais	12
2.8. Mecanismos Arquiteturais	13
<b>3. Modelagem e diagramas arquiteturais</b>	<b>14</b>
3.1. Nível 1: Diagrama de Contexto	15
3.2. Nível 2: Diagrama de Contêiner	16
3.3. Nível 3: Diagrama de Componentes	17
3.4. Nível 4: Código	17
<b>APÊNDICE A - DETALHAMENTO DOS CASOS DE USO:</b>	<b>19</b>
<b>APÊNDICE B - DIAGRAMAS DE SEQUÊNCIA PARA OS CASOS DE USO 1 E 2</b>	<b>21</b>
<b>APÊNDICE C - MODELAGEM DE ESTADOS DA CLASSE AGENDA</b>	<b>22</b>
<b>APÊNDICE D - MODELAGEM DE ATIVIDADES</b>	<b>23</b>

## **1. Apresentação**

As Organizações da Sociedade Civil (OSC) são entidades sem fins lucrativos que desempenham um papel essencial na estrutura social, atuando onde o setor privado e o governo muitas vezes não alcançam. Essas organizações promovem ações em áreas como assistência social, saúde, educação e defesa dos direitos humanos, contribuindo diretamente para o bem-estar coletivo. Seus recursos, em grande parte, provêm de subvenções governamentais, além de doações realizadas por pessoas físicas e jurídicas (SEBRAE, 2022)

### **1.1 Cliente**

O **Círculo de Trabalhadores Cristãos de Belo Horizonte (CTC)**, denominado cliente ou parceiro, é uma **OSC sem fins lucrativos** que promove atividades com o objetivo de atuar em causas de caráter social. A instituição realiza ações como bazares, eventos gastronômicos, aluguel de lojas para afiliados, entre outras iniciativas destinadas a arrecadar fundos. Esses recursos são revertidos em eventos culturais e de lazer para a população atendida, garantindo o acesso a um dos direitos sociais fundamentais, que possibilitam o desenvolvimento pessoal e coletivo.

Entre as atividades realizadas em parceria com outras associações, destacam-se **sessões de cinema, aulas de dança, apresentações artísticas e o projeto Academia da Cidade**. O público-alvo principal é formado por pessoas em situação de vulnerabilidade social, mas atualmente a instituição atende também a uma diversidade maior da população local, de diferentes perfis socioeconômicos, com ou sem acesso a meios tecnológicos.

### **1.2 Problema**

O **Círculo de Trabalhadores Cristãos de Belo Horizonte (CTC)**, relata que todo o cotidiano da OSC é planejado, utilizando cadernos, documentos escritos, planilhas simples, bem como utilização de mensagens via whatsapp ou instagram. Também relatam que não possuem tanta visibilidade quanto gostariam, apesar de desenvolverem várias ações interessantes, as pessoas precisam ir presencialmente para se inteirar de tudo que acontece no círculo. Problemas como coleta de assinaturas de reuniões, organização de salas para projetos que atuam dentro dos espaços do círculo, dificuldade em cálculos para relatórios mensais, falta de um sistema mais otimizado para organização, planejamento,

movimentações financeiras, divulgação de atividades, controle de pessoas que participam dos eventos e um melhor controle dos atendimentos e locatários.

### **1.3 Objetivo do trabalho**

Nosso objetivo é atuar em cima dos principais problemas relatados, site de divulgação e organização da OSC. A ideia é desenvolver um sistema web que auxiliará na divulgação dos trabalhos feitos pela OSC, bem como dos principais eventos, montando uma home page que ajude nessa divulgação, com imagens, textos, calendário de eventos. Outras funções da plataforma atuariam no auxílio organizacional, um sistema para cálculos de movimentações financeiras que gere um relatório final pronto para envio, sistema de auxílio para definição de locatários, organização de espaço e horários para os diferentes eventos/organizações que utilizam do espaço do círculo. Todo o sistema, tem como foco principal, auxiliar o CTC BH nas principais atividades do cotidiano e na divulgação de todo o cronograma.

### **1.4 Definições e Abreviaturas**

A seguir, são apresentadas as definições, siglas e abreviações utilizadas neste documento e no desenvolvimento do projeto, com o objetivo de garantir a clareza e o pleno entendimento das informações técnicas e institucionais.

1. **CTC – Centro dos Trabalhadores Cristãos de Belo Horizonte:** Organização responsável pela iniciativa do projeto, voltada para ações sociais, educacionais e de apoio à comunidade.
2. **ODS – Objetivos de Desenvolvimento Sustentável:** Conjunto de metas globais estabelecidas pela ONU para promover o desenvolvimento sustentável em diversas áreas, como erradicação da pobreza, educação de qualidade e igualdade de gênero.
3. **WP – WordPress:** Sistema de gerenciamento de conteúdo (CMS) de código aberto, utilizado para criação e administração de sites, blogs e lojas virtuais, com ampla personalização por meio de temas e plugins.
4. **OSC – Organização da Sociedade Civil:** Instituições privadas sem fins lucrativos que atuam em diversas áreas do interesse público, como assistência social, educação e meio ambiente.
5. **ONG – Organização Não Governamental:** Entidade do terceiro setor, sem fins lucrativos e independente do governo, que atua em causas sociais, ambientais, educacionais, entre outras.
6. **C# – C Sharp:** Linguagem de programação orientada a objetos desenvolvida pela Microsoft, amplamente utilizada em aplicações .NET.

7. **SQL – Structured Query Language:** Linguagem padrão para gerenciamento e manipulação de bancos de dados relacionais.
8. **NoSQL – Not Only SQL:** Categoria de bancos de dados que não seguem o modelo relacional tradicional, sendo otimizados para escalabilidade, flexibilidade e desempenho em grandes volumes de dados. Exemplos: MongoDB, Cassandra, Redis.
9. **SGBDs – Sistemas de Gerenciamento de Banco de Dados:** Conjunto de programas que permite criar, gerenciar e manipular bancos de dados. Exemplos: MySQL, PostgreSQL, Oracle, SQL Server.
10. **IIS – Internet Information Services:** Servidor web da Microsoft utilizado para hospedar sites e aplicações web, incluindo serviços baseados em ASP.NET.
11. **API – Application Programming Interface (Interface de Programação de Aplicações):** Conjunto de rotinas, protocolos e ferramentas que permite a comunicação entre diferentes sistemas ou componentes de software.
12. **RF – Requisito Funcional:** Funcionalidade ou comportamento esperado de um sistema, descrevendo o que ele deve fazer a partir da perspectiva do usuário.
13. **RNF – Requisito Não Funcional:** Característica ou restrição de qualidade que o sistema deve atender, como desempenho, segurança, escalabilidade ou usabilidade.
14. **UC – Caso de Uso (Use Case):** Descrição de uma interação entre um ator (usuário ou sistema externo) e o sistema, com o objetivo de atingir um resultado específico de valor para o ator.
15. **ORM – Object-Relational Mapping (Mapeamento Objeto-Relacional):** Técnica de programação que permite a conversão de dados entre sistemas orientados a objetos e bancos de dados relacionais. Exemplos: Entity Framework, Hibernate.
16. **JS – JavaScript:** Linguagem de programação leve, interpretada e orientada a objetos, amplamente utilizada no desenvolvimento de interfaces web dinâmicas.
17. **Modelo C4 – Modelo de Arquitetura de Software em 4 Níveis:** Abordagem de modelagem visual que descreve a arquitetura de sistemas de software em quatro níveis: Contexto, Contêiner, Componente e Código. Ajuda na comunicação clara entre equipes técnicas e não técnicas.

## 2. Requisitos

Antes do desenvolvimento da aplicação, é importante definir seus requisitos, funcionais e não funcionais, que descrevem o que o sistema deve fazer e seu comportamento respectivamente. A seguir serão apresentados os requisitos que orientarão a construção da solução e garantirão que elas atendam as necessidades dos stakeholders.

### 2.1. Lista de Atores

Figura 1: História de usuário - Paulo



**Paulo**

**Sobre**  
Funcionário público, 37 anos que ama participar de eventos voltados para caridade

**História**  
Paulo é uma pessoa que gosta de participar dos eventos promovidos pela CTC como visitante e voluntário, como cinemas, alas de dança, apresentações artísticas e brechós.

**Dores**  
Dificuldade de encontrar os eventos feitos pela instituição por meio da internet.  
Não conseguir se cadastrar como voluntário de forma remota.

**Motivações**

- Participar dos eventos propostos pela instituição.
- Ajudar os necessitados nos eventos benfeiteiros.

**Necessidades**

- Conseguir se cadastrar como voluntário sem ser de forma remota.
- Ter maior visibilidade dos eventos da instituição.

Fonte: Autoria Própria

Figura 2: História de usuário - Shirley



**Shirley**

**Sobre**  
• Idade: 68 anos  
• Localização: Bairro residencial da cidade  
• Escolaridade: Ensino Fundamental completo  
• Ocupação: Aposentada  
• Renda: R\$ 1.500/mês

**História**  
Shirley é aposentada, mora sozinha e depende de programas sociais e da OSC para ter acesso a recursos como atendimento social, atividades comunitárias de lazer, saúde e atividades físicas. Tem rotina tranquila, mas limitada em mobilidade e acesso a tecnologias.

**Dores**  
• Esquecer datas de eventos importantes.  
• Ter dificuldade de acesso às atividades devido à mobilidade reduzida.  
• Não receber informações claras ou atualizadas.

**Motivações**

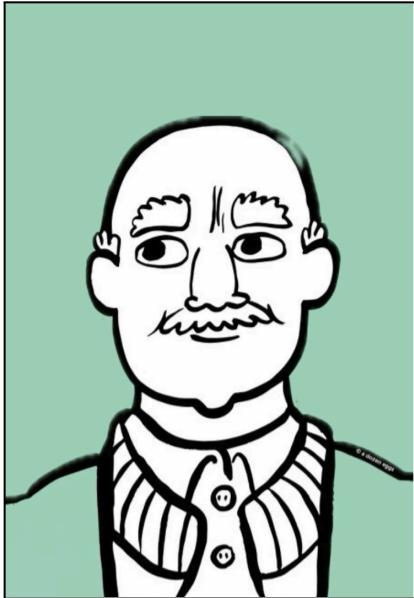
- Necessidade de apoio social e econômico.
- Busca de convívio social e segurança.
- Receber lembretes e notificações que facilitem a participação.

**Necessidades**

- Participar de eventos para ter acesso a serviços de apoio.
- Sentir-se acolhida e incluída na comunidade.
- Mantener contato social e atividades que tragam bem-estar.

Fonte: Autoria própria

**Figura 3: História de usuário - Junior**



# Junior

**Sobre**  
Gestor Do Círculo Trabalhador Cristão, 49 anos, que tem a obrigação de gerenciar os dados da OSC.

**História**  
Júnior é um dos voluntários do CTC que cuida dos eventos da instituição e registro de pessoas relacionadas a eles.

**Dores**  
Dificuldade de gerenciar e salvar os dados dos voluntários e colaboradores nos eventos da instituição

**Motivações**

- Dar visibilidade às pessoas que são "invisíveis" para sociedade.
- Ter melhor organização dos enevnts da intituição.

**Necessidades**

- Dar visibilidade para os voluntários/collaboradores que participam dos eventos.
- Facilitar o preenchimento das listas de presença dos eventos

Fonte: Autoria própria

**Figura 4: História de usuário - Marta**



# Marta

**Sobre**

- Idade: 42 anos
- Localização: Centro da cidade
- Escolaridade: Ensino Superior completo em Administração ou Contabilidade
- Ocupação: Gestora Financeira da OSC como tarefa extra

**História**  
Marta é responsável pelo controle financeiro da OSC, incluindo relatórios contábeis, declarações e organização dos recursos. Tem experiência em gestão de finanças e trabalha de forma metódica, mas ainda não tão digitalizada

**Dores**

- Baixa digitalização e integração: excesso de planilhas e documentos físicos gera lentidão e risco de erro.
- Informações dispersas: dificuldade de centralizar dados financeiros.
- Inconsistências: falta de automação aumenta a chance de dados incorretos.
- Pressão de prazos: coleta manual torna o fechamento contábil mais demorado.

**Motivações**

- Garantir transparência e conformidade legal.
- Aumentar eficiência e reduzir retrabalho.
- Ter dados precisos para decisões e reconhecimento profissional.

**Necessidades**

- Sistema integrado com relatórios automáticos.
- Alertas de prazos e rastreabilidade financeira.
- Ferramenta simples e intuitiva.

Fonte: Autoria própria

## 2.2. Lista de Funcionalidades

Os requisitos aqui apresentados correspondem às funcionalidades solicitadas pelo parceiro/cliente na visão de negócios, tais como:

- Gerenciamento de usuários do sistema
- Gerenciamento dos voluntários
- Criação de relatórios financeiros
- Agendamento das atividades internas que serão realizadas no local
- Criação de relatório dos participantes dos eventos
- Notificação no sistema quando um evento estivesse faltando 1 dia para acontecer
- Gerenciamento dos aluguéis
- Gerenciamento dos beneficiários

### 2.3. Requisitos Funcionais

ID	Descrição Resumida	Dificuldade (B/M/A)*	Prioridade (B/M/A)*
RF01	O sistema deve permitir que o usuário realize login	B	A
RF02	O sistema deve permitir realizar o gerenciamento de usuários (criar / atualizar / visualizar / deletar)	M	A
RF 03	O sistema deve permitir realizar gerenciamento de voluntários (criar / atualizar / visualizar / deletar)	M	M
RF 04	O sistema deve permitir a criação de relatórios financeiros	M	M
RF 05	O sistema deve permitir realizar o agendamento de atividades no local	M	M
RF 06	O sistema deve permitir a criação de um relatório de participação dos usuários nos eventos	M	A
RF 07	O sistema deve permitir realizar o gerenciamento de eventos (criar / atualizar / visualizar / deletar)	M	M
RF 08	O sistema deve permitir realizar o gerenciamento de alugueis (criar / atualizar / visualizar / deletar)	M	M
RF 09	O sistema deve exibir um lembrete na tela principal quando um evento estiver próximo (faltando 1 dia)	B	B
RF 10	O sistema deve permitir o gerenciamento dos beneficiários do sistema (criar / atualizar / visualizar / deletar)	M	A
RF 11	O sistema deve permitir que os funcionários registrem o histórico de atendimentos de cada beneficiário.	M	B
RF 12	O sistema deve permitir o envio de notificações por e-mail aos usuários sobre eventos agendados.	A	B

\*B=Baixa, M=Média, A=Alta.

## 2.4. Requisitos Não Funcionais

ID	Descrição	Prioridade B/M/A
RNF01	O sistema deve ser acessível pelos principais navegadores web de forma responsiva.	A
RNF02	O sistema deve manter a integridade dos dados dos beneficiários e voluntários	A
RNF03	O sistema deve atender às WCAG, garantindo a acessibilidade de todas as interfaces.	M
RNF04	O sistema deve ser modularizado e manter padrões de código limpo, para que a manutenção posterior seja possível	M
RNF05	O sistema deve ser compatível com máquinas com hardware de no mínimo: processador de 1-2GHz, 100 MB de espaço de disco e 4GB de RAM.	B
RNF06	O sistema deve dar feedbacks com mensagens claras para o usuário para operações de sucesso ou falha	M
RNF07	O sistema deve garantir que nenhuma informação cadastrada seja perdida, mesmo em caso de falha do sistema ou desconexão	A

\*B=Baixa, M=Média, A=Alta.

## 2.5. Descrição Resumida dos Casos de Uso ou Histórias de Usuários

Nome do Caso de Uso	Descrição	Atores	Prioridade	Requisitos Associados	Fluxo Principal
Login Seguro	Permite autenticação de usuários com verificação de credenciais e bloqueio após tentativas excessivas.	Usuário, Administrador	Alta	RNF01, RNF07, RNF08, RNF15	1. Usuário acessa tela de login. 2. Informa credenciais. 3. Sistema valida dados e aplica criptografia. 4. Concede ou nega acesso, exibindo mensagem clara.
Gerenciar Usuários	Cadastrar, editar, excluir e definir papéis (RBAC) para usuários internos.	Administrador	Alta	RNF01, RNF02, RNF09, RNF15	1. Administrador acessa painel. 2. Inclui ou altera dados de um usuário. 3. Sistema salva e registra log da operação.
Gerenciar Voluntários	Permite registro, atualização e consulta de voluntários, garantindo integridade dos dados.	Administrador, Voluntário	Alta	RNF01, RNF02, RNF09, RNF15	1. Administrador abre módulo de voluntários. 2. Insere ou edita informações. 3. Sistema confirma operação com feedback claro.
Gerenciar Eventos	Criação, edição e divulgação de eventos com atualização em tempo real.	Administrador, Visitante	Média	RNF01, RNF03, RNF13	1. Administrador cria evento. 2. Sistema publica no calendário e atualiza para visitantes sem recarregar a página.
Sistema de Agendamento	Permite que usuários reservem espaços e horários, controlando uso simultâneo.	Administrador, Locatário	Média	RNF01, RNF07, RNF11, RNF15	1. Locatário solicita reserva. 2. Sistema verifica disponibilidade. 3. Administrador aprova e agenda.
Gerar Relatório Financeiro	Registrar entradas/saídas e gerar relatórios em PDF/CSV em até 5 s.	Gestor Financeiro	Alta	RNF01, RNF07, RNF10, RNF12, RNF15	1. Gestor insere transações. 2. Sistema calcula saldo. 3. Gera relatório exportável.
Relatório de Participação	Geração de relatórios de presença em eventos, exportável em PDF/CSV.	Administrador	Média	RNF01, RNF07, RNF12	1. Administrador seleciona evento. 2. Sistema compila dados e exporta arquivo.
Gerenciar Aluguéis	Cadastro de contratos de locação, controle de pagamentos e logs de auditoria.	Administrador, Locatário	Alta	RNF01, RNF07, RNF14, RNF15	1. Locatário solicita locação. 2. Administrador aprova. 3. Sistema registra todas as alterações.

## 2.6. Histórias de Usuário

<b>EU, COMO... PAPEL</b>	<b>QUERO/PRECISO... FUNCIONALIDADE</b>	<b>PARA... MOTIVO/VALOR</b>
Funcionário do CTC	quero realizar login no sistema com meu usuário e senha	para ter acesso seguro às funcionalidades internas da ONG.  ..
Funcionário do CTC	quero cadastrar, editar e remover usuários do sistema	para manter o controle de quem pode acessar o sistema
Funcionário do CTC	quero criar, atualizar, visualizar e deletar cadastros de voluntários	para organizar a equipe de apoio do CTC.
Funcionário do CTC	quero gerar relatórios financeiros	para enviar para a empresa de contabilidade parceira.
Funcionário do CTC	quero gerenciar as atividades internas	para organização do CTC
Funcionário do CTC	quero criar, atualizar, visualizar e deletar eventos	para organizar e controlar todas as atividades promovidas pela ONG.
Funcionário do CTC	quero criar, atualizar, visualizar e deletar registros de aluguéis	para controlar o uso do espaço de aluguel

<b>EU, COMO... PAPEL</b>	<b>QUERO/PRECISO... FUNCIONALIDADE</b>	<b>PARA... MOTIVO/VALOR</b>
Funcionário do CTC	quero ver um lembrete na tela principal quando um evento estiver próximo (faltando 1 dia)	para não esquecer de preparar e organizar a atividade.
Funcionário do CTC	quero criar, atualizar, visualizar e deletar registros de beneficiários	para manter o controle das pessoas que recebem ajuda do CTC
Funcionário do CTC	quero registrar o histórico de atendimentos de cada beneficiário	para acompanhar o progresso e as necessidades de cada pessoa.
Funcionário do CTC	quero enviar notificações por e-mail aos usuários sobre eventos agendados	para garantir que todos estejam informados e participem das atividades.

## 2.7. Restrições Arquiteturais

A seguir estão as principais **restrições arquiteturais** aplicáveis ao projeto. Essas restrições não são requisitos funcionais, mas limitam/condicionam as escolhas tecnológicas e de implementação, sendo que cada item traz um impacto na arquitetura do projeto:

- **Aplicação web responsiva**

O sistema será entregue como aplicação web acessível por navegadores modernos (desktop e mobile). Impacto: interface responsiva e testes básicos cross-browser.

- **Banco de dados relacional**

Dados principais (usuários, voluntários, beneficiários, eventos, alugueis, lançamentos financeiros, histórico de atendimentos) devem ficar em um banco relacional (PostgreSQL/MySQL) para garantir integridade e facilitar relatórios.

- **Autenticação e controle de papéis**

Mecanismo central de login com suporte a roles (Admin, Funcionário, Voluntário) e hashing seguro de senhas. Impacto: autorização por rota/ação e registro de operações sensíveis.

- **Execução de tarefas agendada**

Lógica de verificação/alerta “1 dia antes” e geração de relatórios deve rodar de forma assíncrona/agendada (cron, worker leve ou serviço gerenciado).

- **Envio de e-mail por provedor externo**

Notificações por e-mail (lembrete de evento a 1 dia, confirmações) via provedor SMTP ou API com filas/retries básicas.

- **Comunicação segura (HTTPS) e backups**

Transporte via HTTPS/TLS obrigatório; política básica de backup periódico dos dados e procedimentos mínimos de restauração.

- **Operar em infraestrutura de baixo custo**

Projetar para rodar bem em ambientes com recursos modestos (otimização de consultas, front leve), preferindo soluções open-source quando possível.

## 2.8. Mecanismos Arquiteturais

Esta seção apresenta uma visão geral dos principais mecanismos que compõem a arquitetura dos softwares que serão produzidos para o CTC, considerando três perspectivas: análise, design e implementação.

Na fase de **análise**, destacam-se os aspectos essenciais que estruturam o sistema, como persistência de dados, integração com sistemas legados, geração de logs, ambiente de front-end, tratamento de exceções, definição do formato dos testes e estratégias de distribuição e implantação (deploy).

Durante o **design**, são definidos os padrões tecnológicos que orientarão cada um desses mecanismos, garantindo consistência, escalabilidade e manutenção adequada do sistema.

Por fim, na etapa de **implementação**, são escolhidos os produtos e ferramentas específicas que darão suporte à solução, permitindo que os componentes projetados sejam efetivamente construídos e operem conforme planejado.

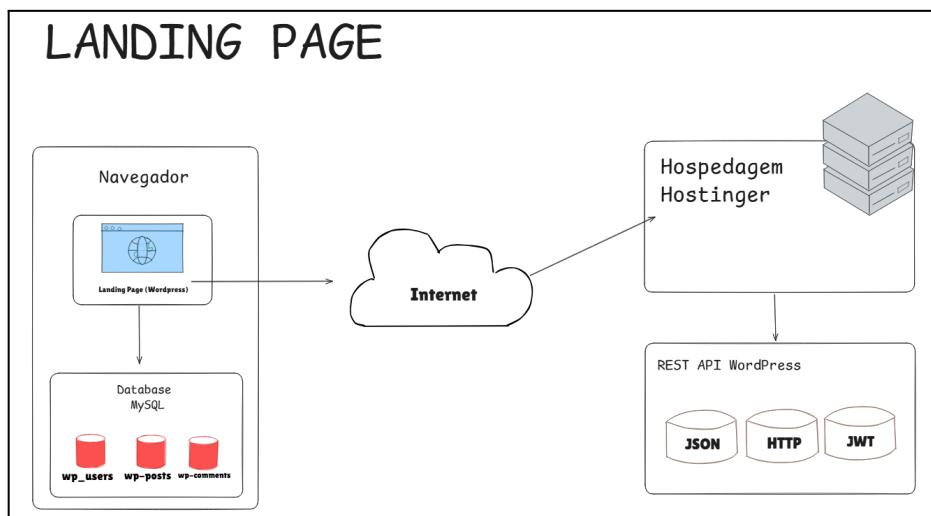
Análise	Design	Implementação
Persistência	ORM	Entity Framework + SQL Server
Front end	Biblioteca JS	React, JavaScript, HTML, CSS
Back end	Framework .NET	C# com ASP.NET Core
Integração	API REST / Web Services	ASP.NET Web API + JSON
Teste de Software	Testes unitários	Jest (para React), xUnit (para C#)
Deploy	CI/CD Pipeline	Vercel (Front-end) + IIS (Back-end)
Logs	Logging Estruturado	Serilog
Tratamento de Exceções	Middleware de exceções	ILogger .NET
Servidor de Aplicação	Servidor Web	IIS (Windows Server)

### 3. Modelagem e diagramas arquiteturais

Como visão geral das duas aplicações, temos:

**LANDING PAGE:** É um website simples (Landing Page) construído com WordPress que tem uma arquitetura monolítica e desacoplada (devido ao uso de API REST), e será armazenado na Hostinger. O navegador acessa o site via Internet, que utiliza um banco de dados MySQL para gerenciar dados como usuários e posts. A comunicação utiliza a REST API do WordPress com formatos JSON, HTTP e JWT.

**Figura 5: Visão geral da Landing Page**

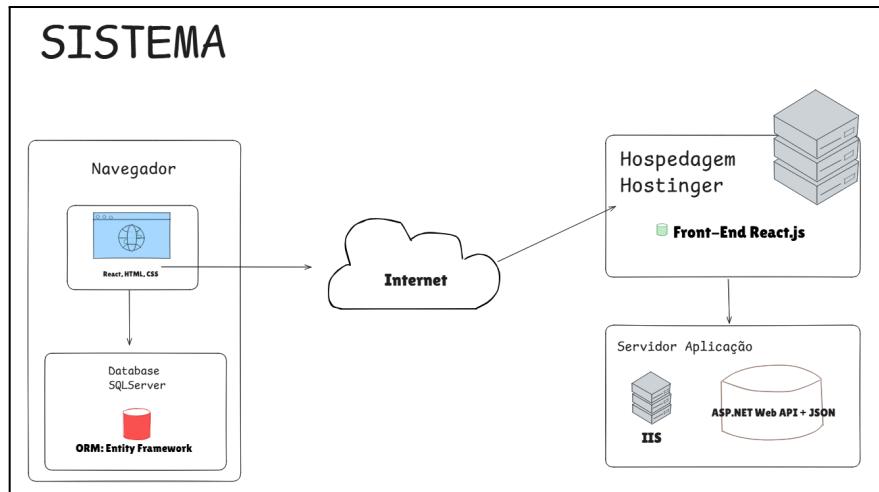


Fonte: Autoria própria

**SISTEMA:** É uma aplicação mais complexa com uma arquitetura Arquitetura de Três Camadas (3-Tier Architecture).

O Front-End (React.js) é hospedado na Hostinger e o usuário ao acessar via navegador. O Front-End se comunica com o Servidor de Aplicação (.NET Web API), que é hospedado no IIS e usa JSON. O servidor acessa um banco de dados SQL Server, utilizando o Entity Framework (ORM).

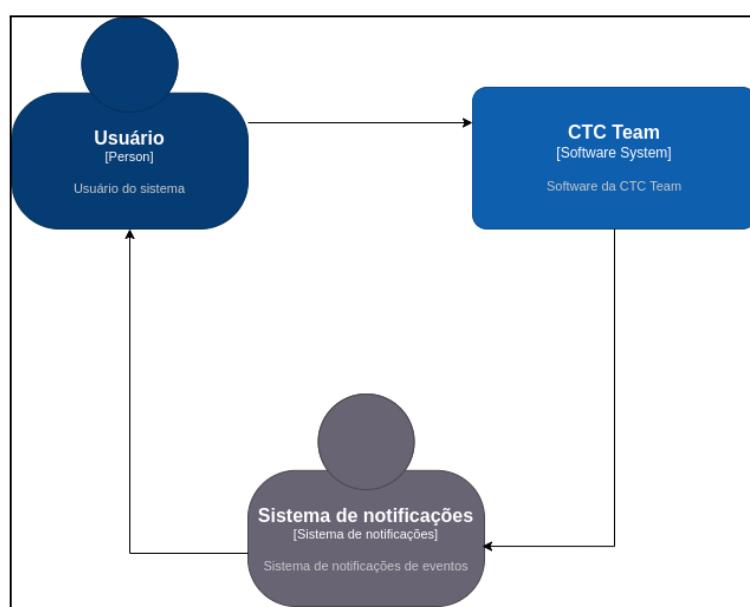
**Figura 6: Visão geral da Landing Page**



Fonte: Autoria própria

### 3.1. Nível 1: Diagrama de Contexto

**Figura 7: Visão geral da Landing Page**

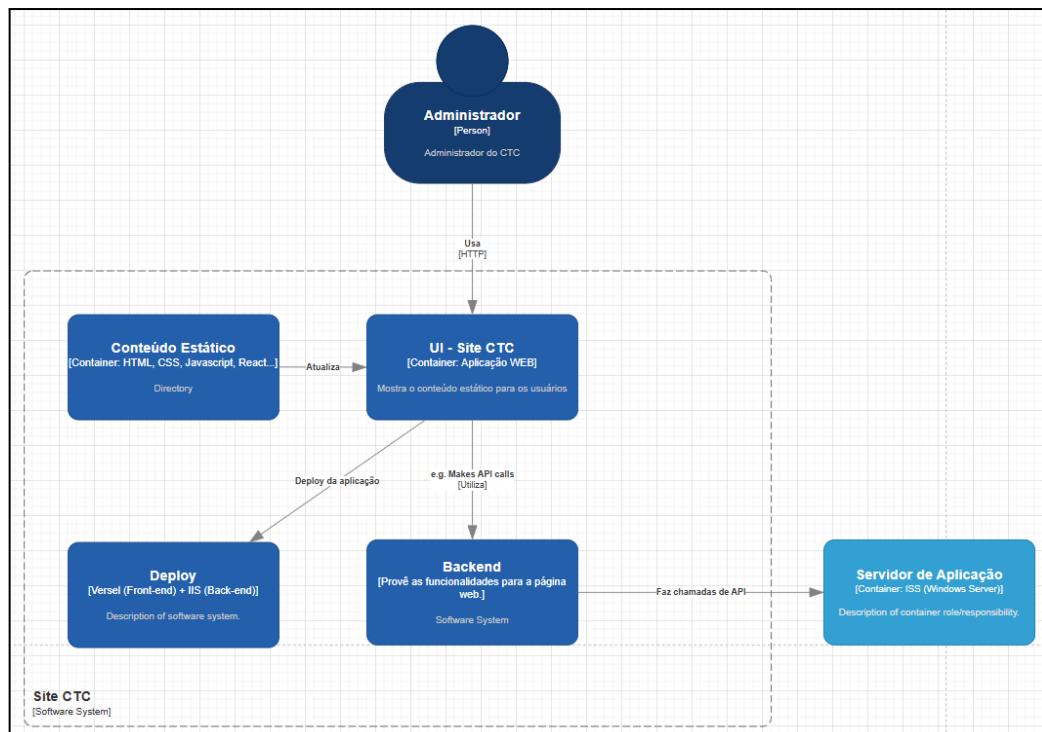


Fonte: Autoria própria

### 3.2. Nível 2: Diagrama de Contêiner

O diagrama de contexto apresenta o Usuário, interagindo com o sistema principal denominado CTC Team. Esse sistema é responsável pelas funções de organização de eventos, emissão de relatórios e gerenciamento de usuários, constituindo o núcleo das operações do CTC.

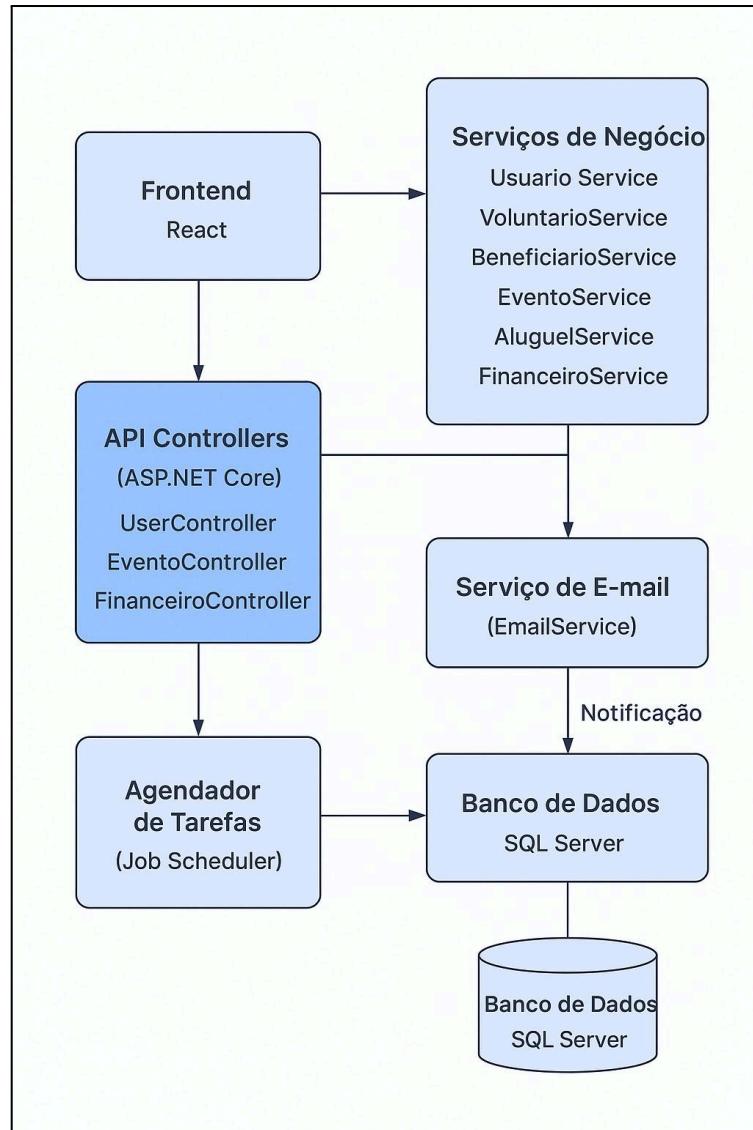
**Figura 8: Visão geral da Landing Page**



Fonte: Autoria própria

### 3.3. Nível 3: Diagrama de Componentes

Figura 9: Visão geral da Landing Page

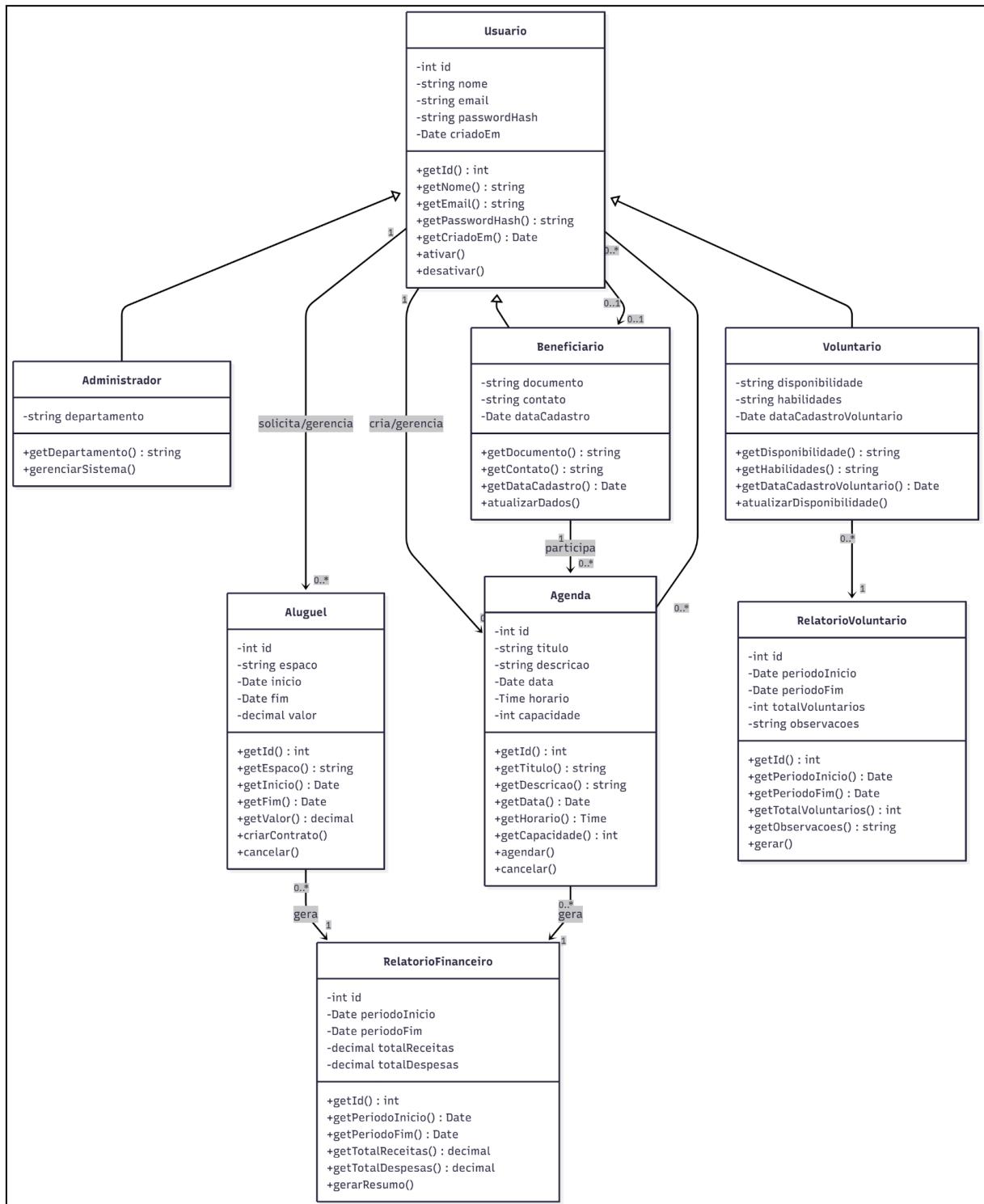


Fonte: Autoria própria

### 3.4. Nível 4: Código

O diagrama de classes modela as entidades principais do sistema. **Usuário** é a superclasse (atributos protegidos: id, nome, email, passwordHash, criadoEm) e é especializada em **Administrador**, **Voluntário** e **Beneficiário** para comportamentos e dados específicos; **Agenda** e **Aluguel** representam, respectivamente, eventos/atendimentos e reservas de espaço, que podem ser criados e gerenciados por usuários, com relação N:N entre Agenda e Usuario para participantes; **RelatorioVoluntario** e **RelatorioFinanceiro** agregam métricas de voluntariado e fluxos financeiros (sumarizando agendas e aluguéis por período). A modelagem privilegia encapsulamento (atributos privados/protected com getters públicos) e facilita persistência relacional.

**Figura 10: Diagrama de classes**



Fonte: Autoria própria

## **APÊNDICE A - DETALHAMENTO DOS CASOS DE USO:**

### **Caso de uso 1:**

**Título:** Criar Evento

**Atores:** Administrador, Visitante

**Pré-condição:** O usuário deve estar logado.

**Pós-condição:** O evento é criado e uma mensagem de sucesso é exibida na tela.

#### **Fluxo Principal:**

1. O ator seleciona a opção “**Criar evento**”;
2. Informa a data do evento;
3. Adiciona a descrição do evento;
4. Salva o evento;
5. O evento é publicado no calendário.

#### **Fluxo Alternativo:**

3. Evento recorrente: O usuário opta por criar um evento recorrente.
  - O sistema solicita a periodicidade (diária, semanal ou mensal);
  - O ator confirma e o sistema cria todos os eventos correspondentes;
  - O fluxo retorna ao passo 5.

#### **Fluxos de Exceção:**

**2a. Data inválida:** O sistema exibe a mensagem “*Data selecionada é inválida. Por favor, digite novamente.*”

2a.1. Quando o usuário informa uma data válida, o fluxo retorna ao passo 3.

## **Caso de uso 2:**

**Título:** Realizar Aluguel de Espaço

**Atores:** Administrador, Locatário

**Pré-condição:** O usuário deve estar logado.

**Pós-condição:** O aluguel é criado e uma mensagem de sucesso é exibida na tela.

### **Fluxo Principal:**

1. O ator seleciona a opção “**Solicitar Alocação**”;
2. Seleciona o espaço desejado;
3. Informa o dia desejado;
4. Envia a solicitação e aguarda a aprovação;
5. O aluguel é concluído com sucesso.

### **Fluxo Alternativo:**

3. Aluguel por período: O usuário opta por alugar o espaço por mais de um dia.
  - O sistema solicita a data de início e de término do aluguel;
  - O ator informa as duas datas;
  - O sistema verifica a disponibilidade para todo o período;
  - Se todas as datas estiverem livres, o fluxo retorna ao passo 4.

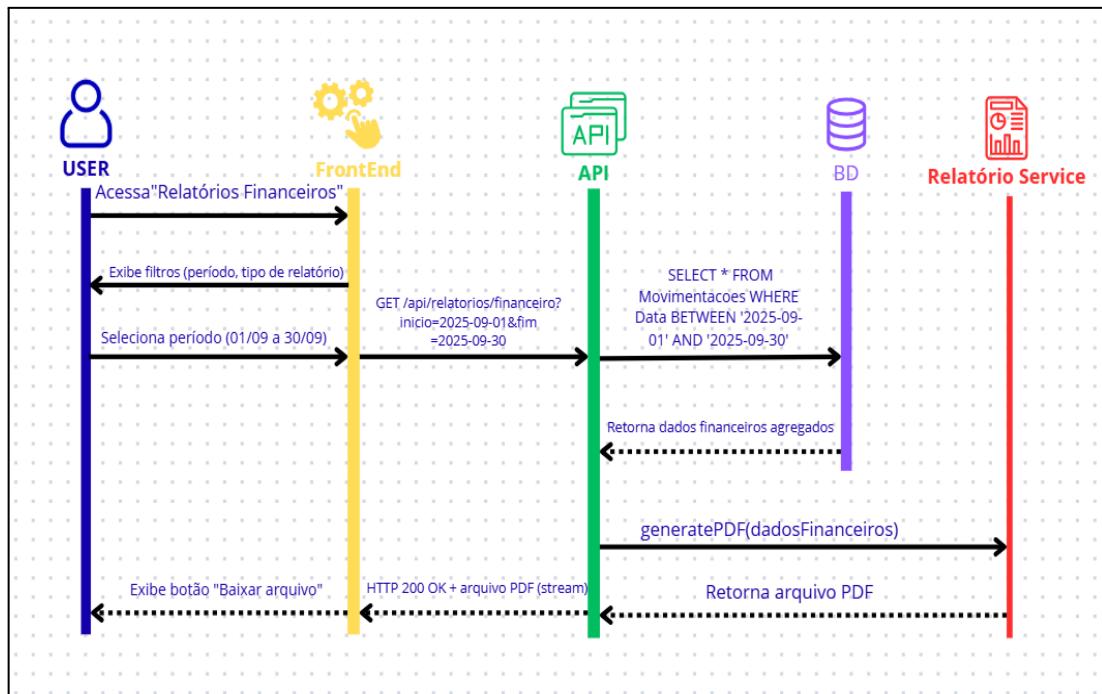
### **Fluxos de Exceção:**

**2a. Espaço indisponível:** O sistema exibe a mensagem “*O espaço selecionado já está alugado. Por favor, escolha outro.*”

2a.1. Quando o usuário escolhe outro espaço, o fluxo retorna ao passo 3.

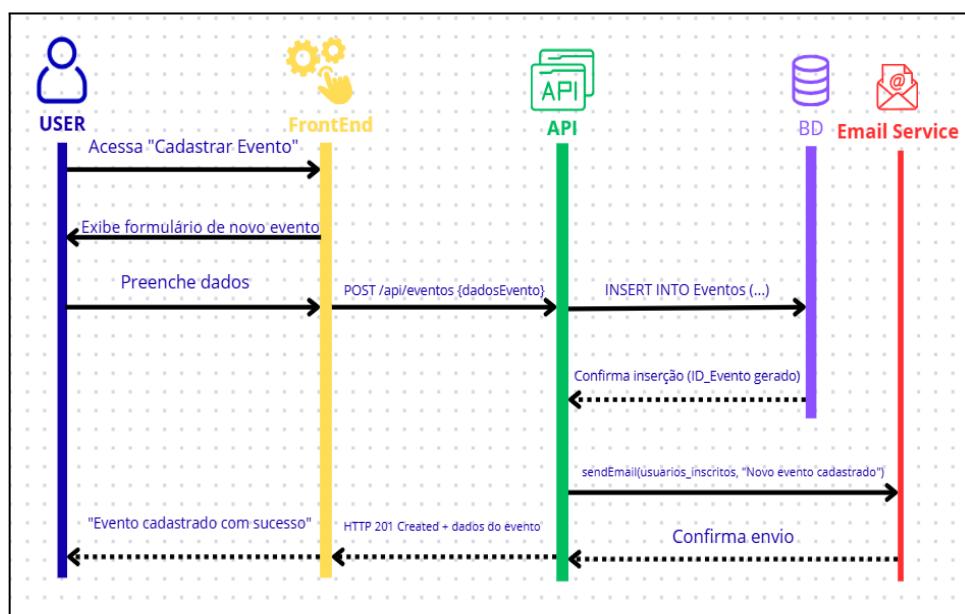
## APÊNDICE B - DIAGRAMAS DE SEQUÊNCIA PARA OS CASOS DE USO 1 E 2

**Figura 11: Diagrama de Sequência do caso de uso 1 - Gerenciar Eventos**



Fonte: Autoria própria

**Figura 12: Diagrama de Sequência do caso de uso 1 - Gerenciar Eventos**



Fonte: Autoria própria

## APÊNDICE C - MODELAGEM DE ESTADOS DA CLASSE AGENDA

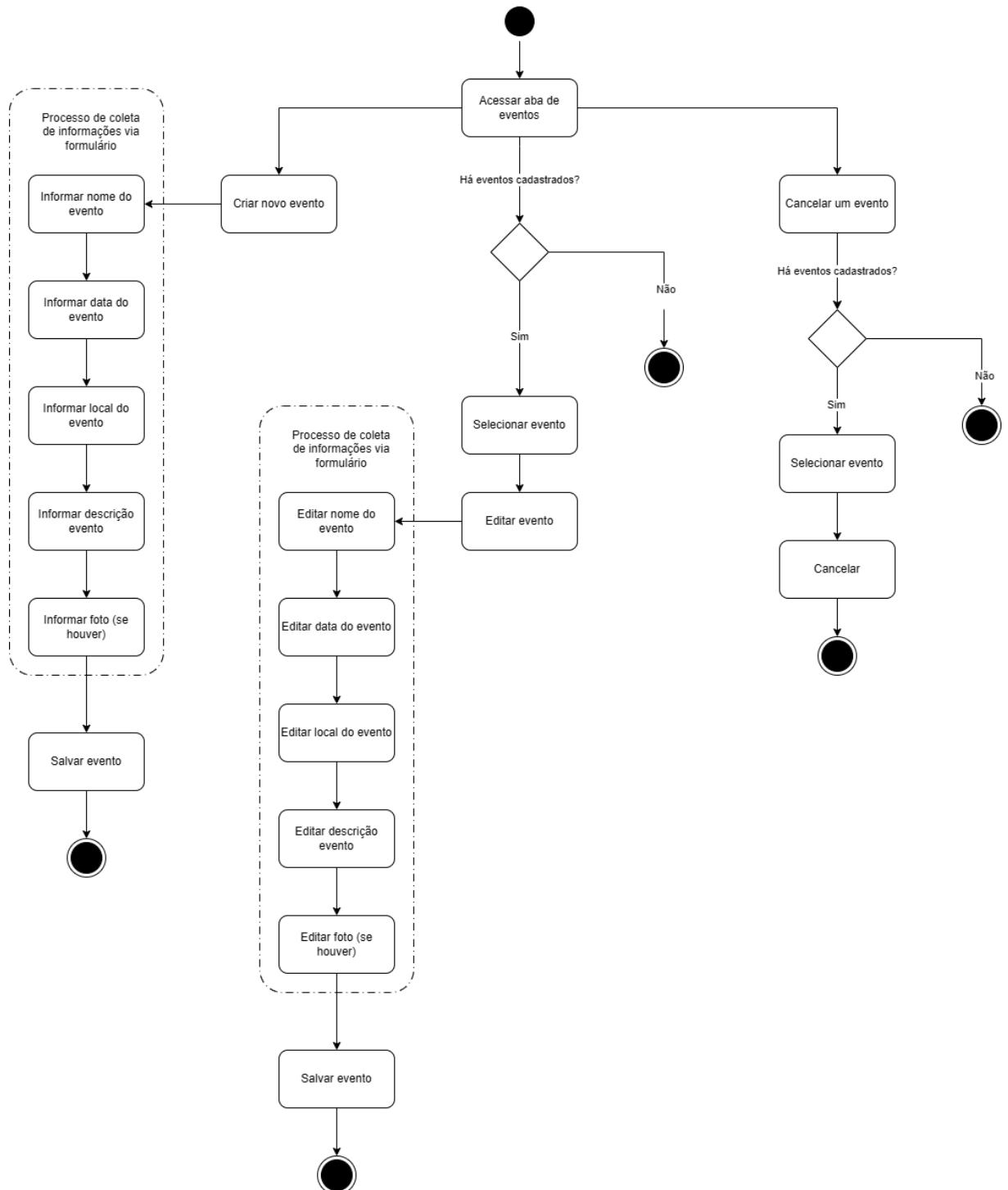
Figura 13: Diagrama de Estados - Classe Agenda



Fonte: Autoria própria

## APÊNDICE D - MODELAGEM DE ATIVIDADES

**Figura 14: Diagrama de Sequência do caso de uso 1 - Gerenciar Eventos**



Fonte: Autoria própria