FECAP

PROJETO SABOR SOLIDÁRIO

Requisitos da disciplina Modelagem de Software e Arquitetura de Sistemas

São Paulo

2024

INTEGRANTES DO PROJETO e RA'S

Bruna Cristina Lira Curralo	RA: 24025837
Deivid Gomes de Oliveira	RA: 24025839
Rafaela Coelho Bastos	RA: 24026076
Isabela Nunes Zeferino	RA: 24026460

Sumário

1. INTRODUÇÃO

O projeto Sabor Solidário tem como objetivo ser um centro de informações sobre ONGs que atuam na distribuição de alimentos, conectando doadores a organizações que combatem a fome no Brasil.

Além disso, busca conscientizar sobre o desperdício de alimentos e promover um futuro sem fome.

2. DOCUMENTO DE ABERTURA DO PROJETOS

➢ Prefácio

O documento será utilizado por desenvolvedores, gestores de ONGs e doadores interessados em compreender o funcionamento do sistema **Sabor Solidário**. Ele será atualizado conforme novas funcionalidades e melhorias forem implementadas no sistema. O histórico de versões contém as seguintes alterações:

- **Versão 1.0**: Primeira versão do sistema, incluindo funcionalidades básicas como cadastro, listagem e remoção de ONGs.
- **Versão 1.1**: Melhorias na responsividade do frontend para dispositivos móveis e otimização de consultas ao banco de dados SQLite.
- **Versão 1.2**: Inclusão de funcionalidade de filtro por estado na lista de ONGs.
- **Versão 2.0**: Adição de um calendário interativo para o agendamento de doações, permitindo que doadores marquem datas e horários para entrega.
- **Versão 2.1**: Implementação de relatórios para ONGs, permitindo acompanhar o volume de doações recebidas por mês.
- **Versão 3.0**: Integração com APIs de geolocalização para que os doadores possam localizar ONGs mais próximas automaticamente.

➢ Introdução

O sistema **Sabor Solidário** foi desenvolvido para atender à necessidade de centralizar informações sobre ONGs que trabalham com distribuição de alimentos. Ele permite que doadores localizem ONGs por estado e acessem informações detalhadas para facilitar as doações.
O sistema está alinhado aos objetivos estratégicos de combate à fome, promovendo a redistribuição de alimentos e conectando doadores a instituições de forma eficiente.

➢ Glossário

ONG: Organização não governamental, sem fins lucrativos, que realiza ações sociais.

CNPJ: Cadastro Nacional da Pessoa Jurídica, usado para identificar ONGs no sistema.

Responsividade: Capacidade do site de se adaptar a diferentes tamanhos de tela e dispositivos.

Backend: Parte do sistema que processa e armazena os dados.

Frontend: Interface gráfica visível para o usuário.

- **Definição de requisitos de usuário**

Os usuários terão acesso às seguintes funcionalidades:

1. Cadastro de ONGs para disponibilizar informações relevantes como nome, endereço, e horário de funcionamento.
2. Filtro por estado para localizar ONGs próximas ao doador.
3. Remoção de ONGs, caso parem de operar.
4. Lista de ONGs acessível e atualizada em tempo real.

Usuário	Ação	Resultado Esperado
ONG	Cadastro	ONG visível na lista de ONGs.
Doador	Filtrar ONGs por estado	Exibição de ONGs localizadas no estado escolhido.
ONG	Remover ONG	Exclusão da ONG da lista e do banco de dados.

Requisitos não funcionais:

1. Responsividade para diferentes dispositivos.
2. Tempo de resposta inferior a 2 segundos.
3. Conformidade com normas de segurança de dados (como HTTPS).

- **Arquitetura do sistema**

O sistema **Sabor Solidário** segue uma arquitetura baseada em três camadas:

1. **Frontend**: Desenvolvido utilizando React, uma biblioteca JavaScript moderna para a criação de interfaces dinâmicas e responsivas.
2. **Backend**: Implementado em Node.js com o framework Express, responsável pela lógica de negócios, manipulação de rotas e conexão com o banco de dados.
3. **Banco de Dados**: SQLite, um banco de dados leve, utilizado para o armazenamento de informações relacionadas às ONGs cadastradas, com suporte total a operações CRUD (Create, Read, Update, Delete).

A comunicação entre o frontend e o backend ocorre via APIs RESTful, garantindo uma troca eficiente de dados.

- **Especificação de requisitos do sistema**

Requisitos Funcionais:

1. Cadastro de ONGs com validação de campos.
2. Listagem e filtragem de ONGs por estado, com informações detalhadas.
3. Atualização e remoção de ONGs cadastradas.
4. Login e autenticação para as ONGs gerenciarem suas informações.

Requisitos Não Funcionais:

1. O frontend desenvolvido em React deve ser responsivo e garantir uma experiência de usuário fluida.
2. O backend deve assegurar a validação de dados e a integridade das operações CRUD no banco SQLite.
3. O sistema deve oferecer tempo de resposta inferior a 2 segundos em consultas e ações.
4. A segurança dos dados transmitidos entre frontend e backend deve ser garantida, utilizando HTTPS e boas práticas de segurança.

Modelos do sistema

Diagrama de Classe

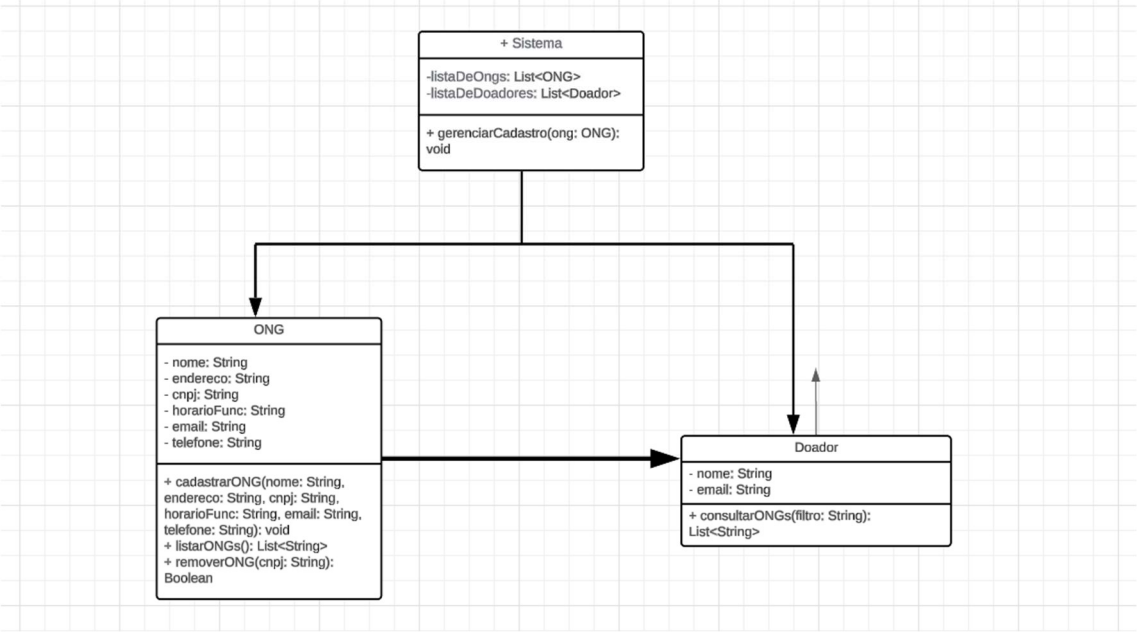
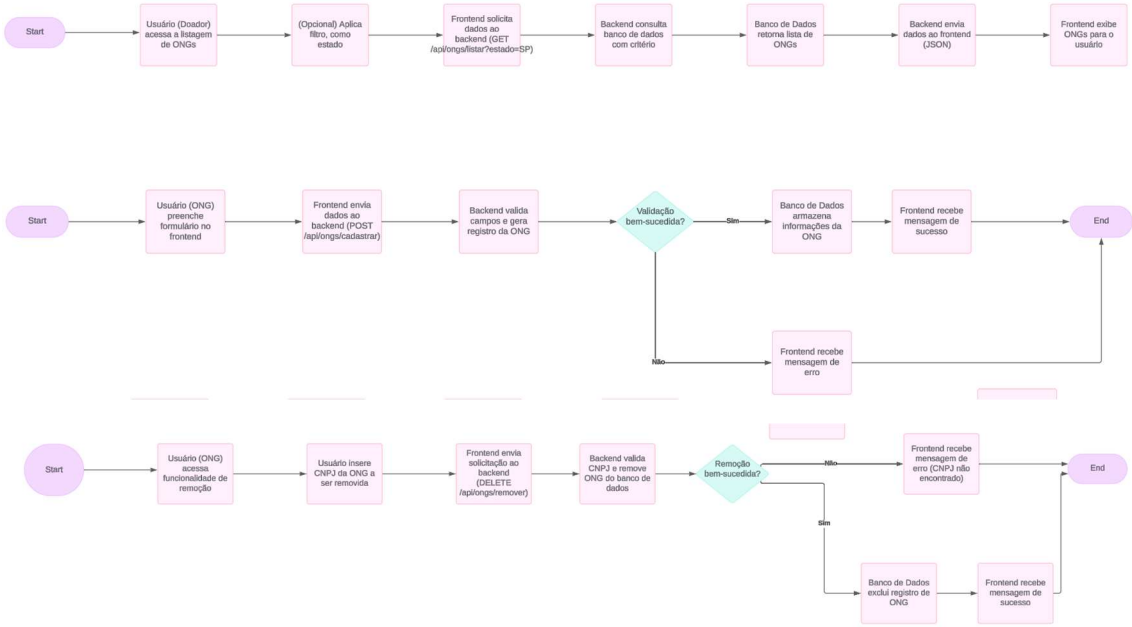
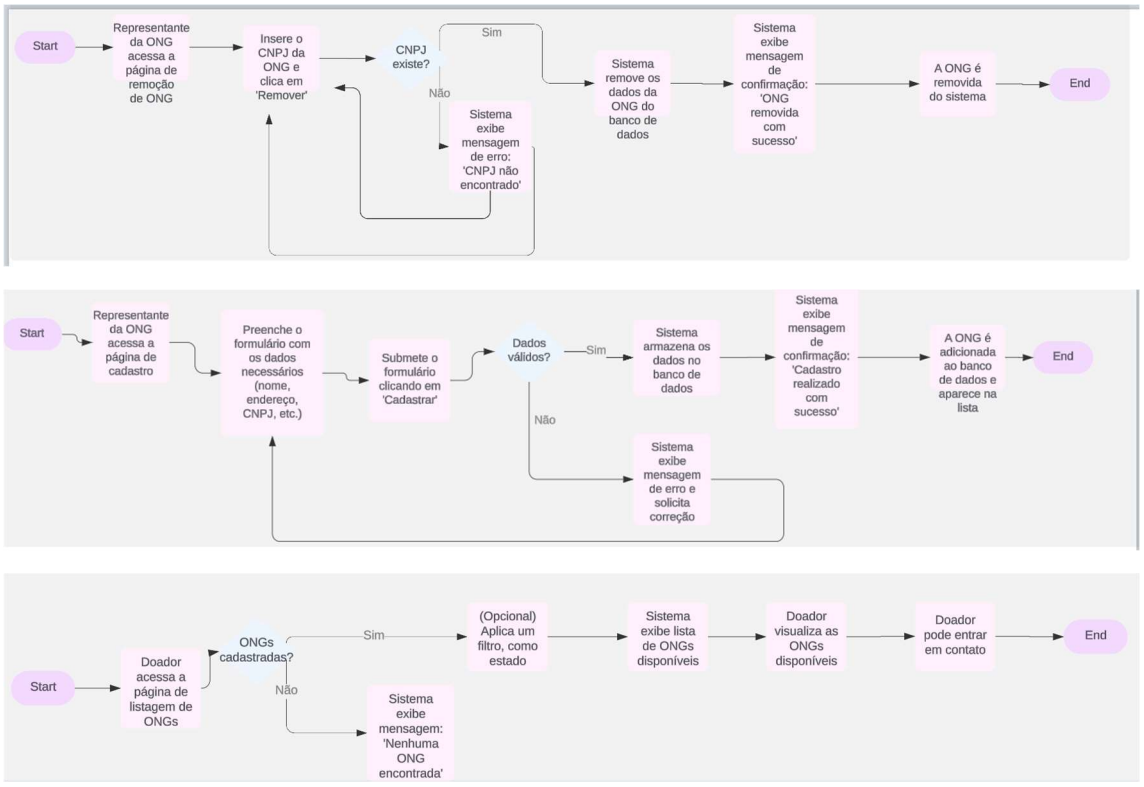


Diagrama de Fluxo de Dados



Casos de Uso



- **Evolução do sistema**

As tecnologias utilizadas (React, Node.js, Express e SQLite) permitem escalabilidade e integração futura com:

1. Um banco de dados mais robusto, como PostgreSQL ou MySQL, caso o número de ONGs cadastradas cresça significativamente.
2. Desenvolvimento de um aplicativo mobile utilizando React Native para facilitar o acesso de doadores e ONGs.
3. Integração com serviços de mapas e geolocalização para melhorar a experiência de busca por ONGs próximas.

- **Apêndices**

Hardware: Requisitos mínimos para servidores que rodem o backend incluem 4 GB de RAM e processador Dual-Core.

Banco de Dados: Estruturado com tabelas para ONGs, doadores e logs de atividades.

3. REQUISITOS DE SISTEMA

3.1 REQUISITOS FUNCIONAIS DE SOFTWARE

Necessários 6 requisitos

RFS01 - Cadastro de ONG	
Função	Permitir que uma ONG cadastre suas informações no sistema.
Descrição	O sistema deve oferecer um formulário para que as ONGs registrem seus dados, como nome, endereço, CNPJ, horário de funcionamento, email e telefone.
Entradas	Nome, endereço, CNPJ, horário de funcionamento, email e telefone.
Fonte	Representante da ONG.
Saídas	Mensagem de confirmação de cadastro.
Ação	Registrar a ONG no banco de dados.

RFS02 - Listagem de ONGs

Função	Exibir uma lista de ONGs cadastradas no sistema.
Descrição	O sistema deve listar todas as ONGs cadastradas, com opção de filtro por estado.
Entradas	Estado (filtro, opcional).
Fonte	Banco de Dados.
Saídas	Lista de ONGs com detalhes (nome, endereço, telefone, etc.).
Ação	Retornar as ONGs disponíveis com base nos critérios fornecidos.
RFS03 - Remoção de ONG	
Função	Permitir que uma ONG remova seu cadastro.
Descrição	O sistema deve remover os dados de uma ONG com base no CNPJ fornecido.
Entradas	CNPJ da ONG.
Fonte	Representante da ONG.
Saídas	Mensagem de confirmação de remoção ou erro.
Ação	Excluir os dados da ONG do banco de dados.

RFS04 - Login	
Função	Autenticar usuários que desejam acessar funcionalidades restritas.
Descrição	O sistema deve permitir login de ONGs para gerenciar seus dados.
Entradas	Email e senha.
Fonte	Representante da ONG.
Saídas	Acesso à área de gerenciamento ou mensagem de erro.
Ação	Validar credenciais e autorizar acesso.
RFS05 - Validação de Dados	

Função	Garantir que os dados inseridos no cadastro de ONG sejam válidos.
Descrição	O sistema deve validar formatos de CNPJ, email e telefone no momento do cadastro.
Entradas	Dados do formulário de cadastro.
Fonte	Formulário preenchido pelo usuário.
Saídas	Mensagem de erro em caso de dados inválidos.
Ação	Bloquear cadastro com dados inválidos.

RFS06 - Filtragem de ONGs por Estado	
Função	Permitir que os doadores filtrem a lista de ONGs cadastradas por estado.
Descrição	O sistema deve oferecer uma funcionalidade que permita aos doadores aplicarem filtros para visualizar ONGs em um estado específico.
Entradas	Estado selecionado pelo doador.
Fonte	Usuário (doador) no frontend.
Saídas	Lista de ONGs filtrada pelo estado selecionado.
Ação	Realizar consulta no banco de dados para retornar apenas ONGs localizadas no estado escolhido.

3.2 REQUISITOS NÃO FUNCIONAIS DE SOFTWARE

Necessários 6 requisitos

RFS01 - Responsividade	
Função	Garantir que o sistema funcione adequadamente em diferentes dispositivos.
Descrição	O sistema deve ajustar automaticamente seu layout e funcionalidades para dispositivos com diferentes tamanhos de tela, como celulares, tablets e desktops.
Entradas	Nenhuma.
Fonte	Requisitos técnicos de usabilidade.
Saídas	Interface visual adequada ao dispositivo do usuário.
Ação	Adotar técnicas de design responsivo e frameworks como Bootstrap no frontend.

RFS02 - Tempo de Resposta	
Função	Manter o tempo de resposta adequado para as operações do sistema.
Descrição	O sistema deve responder a requisições do usuário (listagem, cadastro, remoção) em até 2 segundos.
Entradas	Requisições de ações do frontend.
Fonte	Requisitos de desempenho.
Saídas	Resposta em tempo adequado (ex.: mensagem de sucesso ou erro).
Ação	Otimizar consultas ao banco de dados e minimizar o tamanho das respostas do backend.

RFS03 - Segurança	
Função	Proteger os dados do sistema e dos usuários.
Descrição	As informações de login e cadastro devem ser armazenadas de forma segura, utilizando criptografia para senhas e HTTPS para comunicação entre frontend e backend.
Entradas	Dados sensíveis do usuário (senhas, CNPJs, etc.).
Fonte	Normas de segurança da informação.
Saídas	Dados protegidos contra acessos não autorizados.
Ação	Implementar criptografia e certificados SSL/TLS.

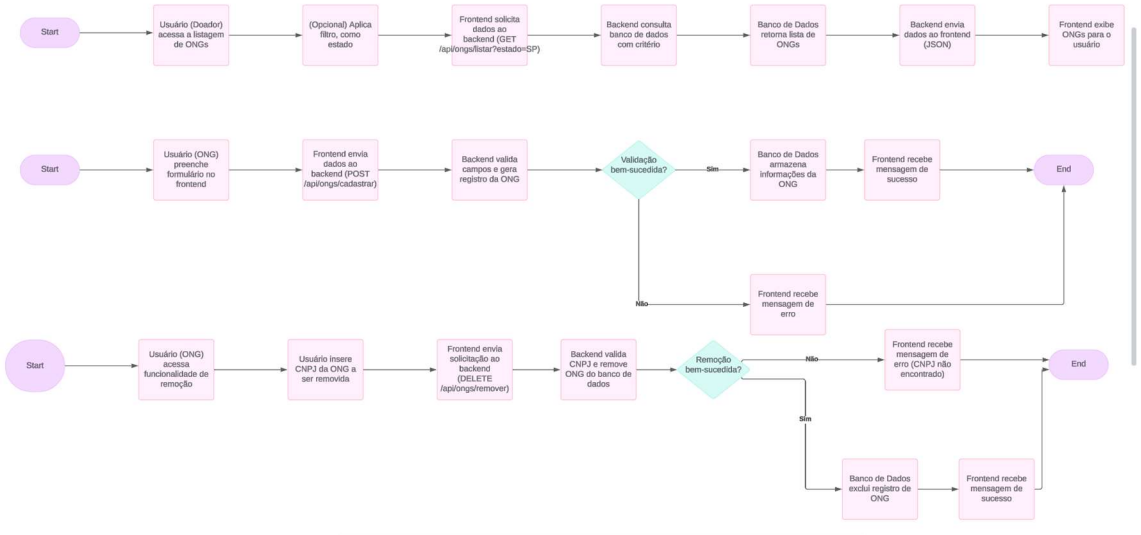
RFS04 - Disponibilidade	
Função	Garantir que o sistema esteja acessível durante a maior parte do tempo.
Descrição	O sistema deve estar disponível para os usuários pelo menos 99% do tempo, com tempo de inatividade mínimo.
Entradas	Nenhuma (característica de infraestrutura).
Fonte	Requisitos de confiabilidade.
Saídas	Sistema disponível para interação com usuários.
Ação	Monitorar uptime do servidor e implementar estratégias de backup e redundância.

RFS05 - Escalabilidade	
Função	Permitir que o sistema suporte um aumento no número de usuários e funcionalidades.
Descrição	O sistema deve ser projetado para permitir adições futuras, como novos módulos ou integração com APIs externas.
Entradas	Solicitações de novos recursos ou aumento de tráfego.
Fonte	Requisitos de manutenção e crescimento.
Saídas	Sistema funcional mesmo com aumento de carga.
Ação	Planejar arquitetura do sistema modular e utilizar servidores escaláveis.

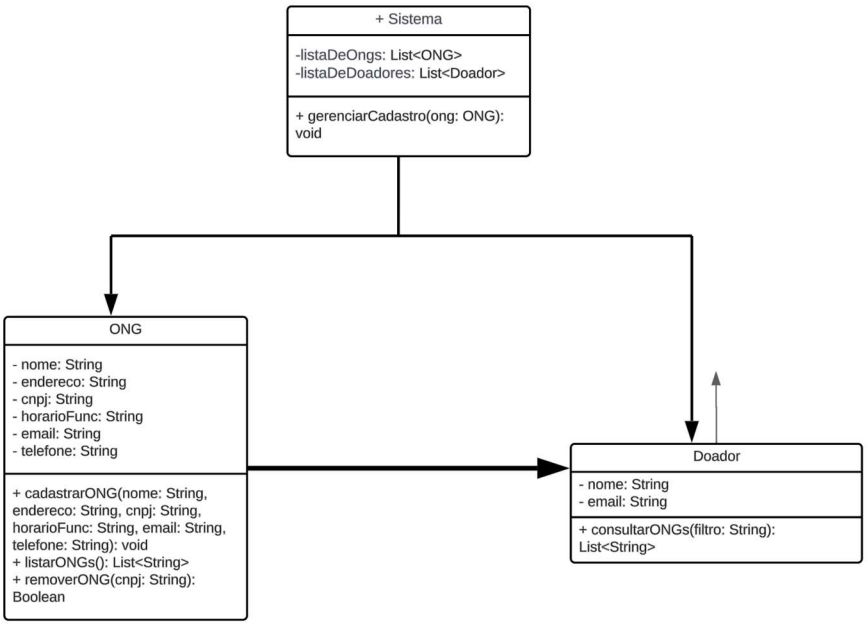
RFS06 - Banco de Dados Leve	
Função	Garantir bom desempenho do banco de dados.
Descrição	O sistema deve usar um banco de dados leve (SQLite) para garantir performance em servidores com recursos limitados.
Entradas	Operações de leitura/escrita no banco de dados.
Fonte	Requisitos de desempenho e ambiente.
Saídas	Consultas e inserções realizadas rapidamente.
Ação	Configurar e otimizar o SQLite para operações rápidas.

4. CASOS DE USO

Apresentar 3 casos de uso do sistema



5. DIAGRAMA DE CLASSE



6. ARQUITETURA DO SISTEMA

O sistema **Sabor Solidário** segue uma arquitetura baseada em três camadas:

1. **Frontend**: Desenvolvido utilizando React, uma biblioteca JavaScript moderna para a criação de interfaces dinâmicas e responsivas.
2. **Backend**: Implementado em Node.js com o framework Express, responsável pela lógica de negócios, manipulação de rotas e conexão com o banco de dados.
3. **Banco de Dados**: SQLite, um banco de dados leve, utilizado para o armazenamento de informações relacionadas às ONGs cadastradas, com suporte total a operações CRUD (Create, Read, Update, Delete).

A comunicação entre o frontend e o backend ocorre via APIs RESTful, garantindo uma troca eficiente de dados.