

FECAP

EcoRota

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

São Paulo
2024

INTEGRANTES DO PROJETO e RA'S

João Pedro Brosselin de Albuquerque Souza - 24026155

Sumário

1 INTRODUÇÃO	3
2. DOCUMENTO DE ABERTURA DO PROJETOS	3
3. REQUISITOS DE SISTEMA	5
3.1 REQUISITOS FUNCIONAIS DE SOFTWARE	5
3.2 REQUISITOS NÃO FUNCIONAIS DE SOFTWARE	7
4. CASOS DE USO	9
5. ARQUITETURA DO SISTEMA	10
6. REFERÊNCIAS BIBLIOGRÁFICAS	11

1 INTRODUÇÃO

A poluição e o consumo excessivo trazem um grande problema para a sociedade moderna, a produção de lixo em massa, que afeta ecossistemas, provoca alagamentos, e prejudica a vida em diversos lugares ao redor do mundo, e em meio a uma situação de que o consumo não está previsto para diminuir foi pensada a ideia da EcoRota, que visa trazer um incentivo à população para destinar seu lixo no local correto e de forma consciente.

Com a ferramenta de busca que auxilia o usuário a encontrar um ponto de coleta de lixo reciclável mais próximo da onde reside e permite que o usuário armazene no seu perfil suas movimentações, podendo guardar a informação de quanto ele está reciclando, quanto papel, plástico, metal e afins estão sendo levados para o local correto, e podendo gerar relatórios para poder mensurar seu impacto positivo no mundo.

O sistema tem a intuição de incentivar a população de forma mais consciente, podendo ver os impactos que uma boa iniciativa podem causar, e assim caminhar para um futuro melhor e mais limpo..

2. DOCUMENTO DE ABERTURA DO PROJETO

Prefácio

Este documento é destinado a desenvolvedores de software, analistas de sistemas, gerentes de projeto e stakeholders que buscam compreender os requisitos e a arquitetura do EcoRota. Ele é uma fonte essencial para a equipe envolvida na implementação e manutenção do software, além de servir como referência para futuras atualizações e melhorias.

Introdução

A necessidade do sistema vem da falta de apoio existente na mídia atual sobre como lidar com a quantidade excessiva de lixo sendo gerado diariamente no mundo todo, e o sistema vem para tentar chamar a atenção para a causa.

As principais funções do sistema incluem o cadastro de usuários, login e autenticação, localização de pontos de coleta, além de permitir que os usuários armazenem suas reciclagens. O sistema funcionará em sinergia com outros sistemas através de APIs, garantindo que as informações sejam acessíveis e integradas de forma segura e eficiente.

Além disso, o sistema visa proporcionar uma plataforma robusta e amigável que incentive a atividade de reciclar, potencializando o engajamento do público e o impacto causado.

Glossário

Cadastro de Usuários: Processo onde um usuário fornece suas informações para criar uma conta no sistema.

Login: Acessar uma conta no sistema utilizando credenciais previamente cadastradas.

Evento: Atividade organizada, como concertos, workshops ou palestras.

Inscrição: Registro de interesse do usuário em participar de um evento específico.

API: Interface de Programação de Aplicações, que permite a comunicação entre diferentes sistemas.

Definição de requisitos de usuário

Cadastro e Autenticação: Permitir que os usuários se cadastrem com informações básicas e realizem login para acessar suas contas.

Criação de Pontos de Coleta: Usuários autorizados/empresas poderão criar novos pontos, especificando endereço, tipo de material, e como chegar.

Pesquisa de Pontos: Usuários poderão buscar pontos de coleta baseando se em um endereço ou CEP inserido,

Inscrição em Eventos: Usuários poderão registrar suas coletas em pontos específicos, gerando assim relatórios para cada ponto..

Os requisitos não funcionais incluem:

Desempenho: O sistema deve carregar em até 8 segundos.

Segurança: Implementação de criptografia para proteção de dados.

Responsividade: O sistema deve ser acessível em dispositivos móveis e desktop de diferentes tamanhos e telas..

Arquitetura do sistema

A arquitetura do sistema é composta por várias camadas que interagem entre si:

1. **Camada de Apresentação:** Utiliza HTML, CSS, REACT.js e JavaScript para exibir a interface do usuário.

2. **Camada de Lógica de Negócios:** Implementada em Node.js, onde as regras de negócio são processadas.
3. **Camada de Acesso a Dados:** Realiza operações de CRUD no banco de dados.
4. **Banco de Dados:** Utiliza SQLiteStudio para armazenamento de dados.
5. **Camada de Segurança:** Implementa protocolos de segurança como HTTPS.
6. **Camada de API:** Facilita a comunicação entre o frontend e o backend.

Especificação de requisitos do sistema

Os requisitos funcionais detalhados incluem funcionalidades como cadastro de usuários, login e autenticação, criação e exibição de pontos de coleta, pesquisa de pontos, e registro de reciclagens. Os requisitos não funcionais, como desempenho, escalabilidade e segurança, são igualmente cruciais e devem ser atendidos.

Evolução do sistema

O sistema foi projetado com a flexibilidade em mente, permitindo futuras modificações conforme as necessidades dos usuários, escalabilidade e aplicação de novas features. A evolução de hardware e software, juntamente com novas demandas do mercado, serão consideradas durante a manutenção do sistema.

3. REQUISITOS DE SISTEMA

3.1 REQUISITOS FUNCIONAIS DE SOFTWARE

RFS01	
Função	Cadastro de usuários.
Descrição	O sistema deve permitir que usuários se cadastrem com e-mail válido e senha + nome e sobrenome para acessar a plataforma.
Entradas	Primeiro nome, sobrenome, e-mail e senha
Fonte	Usuário.
Saídas	"Cadastrado realizado com sucesso!"
Ação	Registro de usuário no banco de dados.

RFS02	
Função	Login e Autenticação
Descrição	O sistema deve permitir que os usuários façam login com suas credenciais, verificando se estão registrados.

Entradas	E-mail e senha.
Fonte	Banco de dados.
Saídas	Login realizado, informação incorreta ou usuário não encontrado + direcionamento para página de cadastro.
Ação	Verificar registro no banco de dados, caso e-mail corresponda com senha o login deve ser realizado, caso o e-mail exista mas a senha esteja incorreta, o usuário deve ser informado que alguma informação está incorreta e por fim, se o e-mail não for encontrado o usuário deve ser instruído para realizar o cadastro.

RFS03	
Função	Criação de Pontos de Coleta
Descrição	O sistema deve possibilitar que usuários específicos registrem pontos de coleta com informações de localização, tipo de material, e como chegar.
Entradas	Nome do ponto, endereço, CEP, e guia.
Fonte	Parceiros.
Saídas	Criação do ponto de coleta no mapa.
Ação	Criar rota do ponto no mapa da plataforma.

RFS04	
Função	Pesquisa de Pontos de Coleta
Descrição	O sistema deve permitir que os usuários pesquisem pontos para reciclar, por CEP ou endereço inserido.
Entradas	Nome do ponto ou CEP ou endereço.
Fonte	Usuário.
Saídas	Pontos correspondentes aos filtros utilizados.
Ação	Buscar no banco de dados os pontos de coleta que estão próximos dos parâmetros informados.

RFS05	
Função	Exibição do Ponto
Descrição	O sistema deve exibir o ponto de coleta informando suas informações como endereço, empresa responsável, guia de chegada e materiais aceitos.
Entradas	Dados do ponto.
Fonte	Banco de dados.
Saídas	Card com informações do ponto de coleta.
Ação	Receber informações dos eventos e separá-los por categorias na interface do usuário.

RFS06	
Função	Registro de Reciclagens
Descrição	O sistema deve permitir que os usuários registrem suas reciclagem por material e em qual ponto de coleta foi despejado.
Entradas	Qual material, peso, quantidade.
Fonte	Usuário.
Saídas	Salvamento do registro na página do usuário para que ele consulte suas reciclagem e possa gerar um relatório de acordo com o tempo determinado
Ação	Registrar reciclagem no banco de dados e reunir suas reciclagem em um gráfico na página do usuário.

3.2 REQUISITOS NÃO FUNCIONAIS DE SOFTWARE

RFS01	
Função	Desempenho
Descrição	O site deve carregar as páginas de eventos em até 6 segundos, podendo variar dependendo da conexão do usuário.
Entradas	Requisições de páginas.
Fonte	Navegador do usuário.
Saídas	Páginas carregadas.
Ação	Otimizar tempo de carregamento.

RFS02

Função	Escalabilidade
Descrição	O sistema deve ser escalável para suportar um grande número de pontos de coleta e usuários simultâneos.
Entradas	Aumento de usuários e eventos.
Fonte	Acesso simultâneo ao sistema.
Saídas	Manutenção de performance.
Ação	Implementar soluções escaláveis.

RFS03	
Função	Segurança
Descrição	As informações de login dos usuários devem ser criptografadas e o site deve seguir padrões de segurança (SSL, proteção contra SQL Injection).
Entradas	Dados do usuário (login, inscrição).
Fonte	Interações do usuário.
Saídas	Dados protegidos.
Ação	Implementar criptografia e padrões de segurança.

RFS04	
Função	Compatibilidade
Descrição	O sistema deve ser compatível com todos os principais navegadores e dispositivos móveis.
Entradas	Navegadores e dispositivos.
Fonte	Variados (Chrome, Firefox, mobile).
Saídas	Interface funcional.
Ação	Testar e ajustar para compatibilidade.

RFS05	
Função	Responsividade
Descrição	O layout do site deve ser responsivo, adaptando-se bem a diferentes tamanhos de tela, especialmente em dispositivos móveis.
Entradas	Tamanhos de tela variados.
Fonte	Dispositivos móveis e desktops.
Saídas	Layout adaptável.
Ação	Implementar design responsivo.

RFS06	
Função	Usabilidade
Descrição	O sistema deve ser fácil de usar e intuitivo, com interfaces claras para navegação entre os eventos e ações como cadastro e login.
Entradas	Ações do usuário.
Fonte	Interface do sistema.
Saídas	Navegação intuitiva.
Ação	Testes de usabilidade e melhorias de interface.

4. CASOS DE USO

1. Cadastro de Evento:

Ator: Admin ou empresa responsável.

Descrição: O usuário logado acessa a página de cadastro de ponto de coleta, preenche os campos necessários e confirma o cadastro. O sistema registra o ponto e o disponibiliza no mapa para que seja exibido se estiver na área escolhida pelo usuário.

Fluxo Alternativo: Se o usuário não estiver logado, o sistema redireciona para a página de login.

2. Busca de Evento:

Ator: Usuário autenticado ou visitante.

Descrição: O usuário utiliza o campo de busca na página principal para localizar pontos de coleta. O sistema exibe os resultados filtrados conforme os parâmetros de busca.

Fluxo Alternativo: Se nenhum ponto de coleta por perto for encontrado, o sistema exibe uma mensagem de “Não há pontos de coleta nessa área”.

3. Inscrição em Evento:

Ator: Usuário autenticado.

Descrição: O usuário seleciona um ponto de coleta do mapa e seleciona a opção registrar nova coleta. O sistema leva o usuário a uma tela para salvar suas reciclagens.

Fluxo Alternativo: Se o usuário não estiver logado, o sistema solicita que faça login antes de prosseguir com o registro, porém é possível utilizar a ferramenta de busca.

5. ARQUITETURA DO SISTEMA

1. Camada de Apresentação (Frontend)

Tecnologias: HTML, CSS, JavaScript, REACT.js.

Responsabilidade: Interface do usuário, onde os usuários podem interagir com o sistema. Exibe formulários de cadastro, login, criação de eventos e exibição de pontos de coleta.

2. Camada de Lógica de Negócios (Backend)

Tecnologias: Node.js e Express.

Responsabilidade: Processamento das regras de negócio, validação de dados, autenticação de usuários, manipulação de eventos e gerenciamento de inscrições.

3. Camada de Acesso a Dados

Responsabilidade: Interação com o banco de dados. Realiza operações de CRUD (Create, Read, Update, Delete) para usuários e eventos.

4. Banco de Dados

Tecnologias: SQLiteStudio.

Responsabilidade: Armazenamento das informações de usuários, eventos e inscrições. Deve ser otimizado para consultas rápidas.

5. Camada de Segurança

Tecnologias: HTTPS para segurança de comunicação.

Responsabilidade: Gerenciar autenticação e autorização, proteger dados sensíveis e evitar vulnerabilidades.

6. Camada de API

Responsabilidade: Comunicação entre o frontend e o backend. Fornece endpoints para operações como cadastro de usuários, registro de pontos de coleta, pesquisa de pontos e registro de materiais.

7. Infraestrutura

Responsabilidade: Gerenciar a implantação e a escalabilidade do sistema. Prover um ambiente seguro e de alto desempenho.

Fluxo Geral

1. **Usuário interage com a interface** (cadastro, login, registro de pontos, busca por pontos de coleta e registro de reciclagens).
2. **Frontend faz requisições para o backend** através da API.
3. **Backend processa as requisições**, aplicando as regras de negócio.
4. **Backend acessa o banco de dados** para armazenar ou recuperar informações.
5. **Retorna a resposta ao frontend**, que atualiza a interface.

6. REFERÊNCIAS BIBLIOGRÁFICAS

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