FECAP

PROJETO SyP

Requisitos da disciplina Modelagem de Software e Arquitetura de Sistemas

São Paulo

2024

INTEGRANTES DO PROJETO e RA'S

Arthur Felipe Alves Nunes	-	24026007
Diogo Bonfim Moreira	-	24026300
Érika Santana da Silva	-	24026205
Guilherme Luis Martins Passos	-	24025973

Sumário

1 INTRODUÇÃO

A SyP tem como objetivo o desenvolvimento de uma plataforma digital focada em doações de roupas e livros, promovendo a solidariedade e a sustentabilidade ao conectar doadores e receptores de forma prática e eficiente. Inspirada em conceitos de economia circular e consciente, a plataforma permite que usuários registrem, filtrem e agendem a entrega ou retirada de itens, garantindo uma experiência intuitiva e segura. A solução busca otimizar o processo de doações, com funcionalidades de geolocalização, avaliações de transações e notificações em tempo real, além de ser projetada para ser escalável, segura e acessível em diversos dispositivos.

2. DOCUMENTO DE ABERTURA DO PROJETO

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	Cadastro de usuários e doadores
Descrição	Permitir o cadastro de novos usuários na plataforma

Entradas	Nome, e-mail, senha, telefone, endereço.
Fonte	Requisitos do sistema, usuário
Saídas	Confirmação de cadastro, e-mail de verificação
Ação	O sistema armazena os dados do usuário e envia um e-mail de verificação

RFS02	
Função	Validar o acesso de usuários cadastrados
Descrição	Os usuários devem autenticar-se usando e-mail e senha ou através de redes sociais (Google, Facebook)
Entradas	E-mail, senha ou credenciais de redes sociais
Fonte	Requisitos de segurança
Saídas	Acesso à plataforma ou mensagem de erro (credenciais incorretas)
Ação	O sistema verifica as credenciais e permite ou nega o acesso

RFS03	
Função	Facilitar o agendamento de entrega ou retirada entre doador e receptor
Descrição	Após solicitar um item, o receptor e o doador podem agendar um horário para retirada ou entrega
Entradas	Data e horário disponíveis, localização, preferências do usuário
Fonte	Doador e receptor
Saídas	Confirmação do agendamento via notificação ou e-mail
Ação	O sistema envia confirmações e registra o agendamento no perfil de ambos

RFS04	
Função	Exibir uma listagem dos itens disponíveis para doação
Descrição	Os doadores podem listar itens (roupas e livros) disponíveis, e os receptores podem visualizá-los com filtros
Entradas	Filtros (categoria, estado de conservação, localização)
Fonte	Base de dados de itens
Saídas	Lista de itens filtrada com imagens, descrição e localização
Ação	O sistema busca e exibe os itens disponíveis de acordo com os filtros

RFS05	
Função	Permitir que usuários cadastrem itens para doação
Descrição	O doador insere informações sobre o item, incluindo fotos e descrição
Entradas	Descrição do item, fotos, categoria (roupa/livro), estado de conservação, localização
Fonte	Requisitos de usuário
Saídas	Item listado na plataforma
Ação	O sistema armazena os dados do item e o exibe na listagem pública

RFS06	
Função	Permitir que doadores e receptores avaliem a experiência da doação
Descrição	Após a entrega, os usuários podem deixar uma avaliação e comentário sobre a transação
Entradas	Nota (1 a 5 estrelas), comentário
Fonte	Usuário
Saídas	Avaliação registrada e visível no perfil do usuário avaliado
Ação	O sistema armazena as avaliações e as exibe para os outros usuários

3.2 REQUISITOS NÃO FUNCIONAIS DE SOFTWARE

Necessários 6 requisitos

RFS01	
Função	Garantir que a plataforma possa lidar com um grande volume de usuários e itens
Descrição	O sistema deve ser capaz de crescer conforme a demanda aumenta
Entradas	Volume de dados e acessos simultâneos
Fonte	Requisitos de sistema e negócios
Saídas	Desempenho consistente, sem queda de velocidade
Ação	Implementar uma arquitetura em nuvem com balanceamento de carga

RFS02	
Função	Proteger dados pessoais e transações
Descrição	A plataforma deve garantir a segurança dos dados dos usuários
Entradas	Dados pessoais (e-mail, senha, informações de doações)
Fonte	Requisitos de privacidade e legislação (LGPD, GDPR)
Saídas	Dados criptografados e protegidos
Ação	Implementar criptografia de dados, autenticação em dois fatores e práticas seguras de armazenamento

RFS03	
Função	Garantir alta disponibilidade do sistema
Descrição	A plataforma deve estar disponível 24/7, com no máximo 1% de tempo de inatividade
Entradas	Demandas dos usuários e transações
Fonte	Requisitos de negócios e experiência do usuário
Saídas	Plataforma sempre acessível
Ação	Implementar servidores redundantes e monitoramento em tempo real

RFS04	
Função	Garantir tempos de resposta rápidos
Descrição	O sistema deve carregar páginas e realizar buscas em menos de 3 segundos
Entradas	Consultas, requisições de usuários
Fonte	Padrões de usabilidade
Saídas	Respostas rápidas, sem atrasos
Ação	Otimização de código, banco de dados eficiente e caching

RFS05	
Função	Suportar acesso em diferentes dispositivos
Descrição	O sistema deve funcionar bem em desktops, tablets e smartphones
Entradas	Resoluções de tela, diferentes navegadores e sistemas operacionais
Fonte	Requisitos de design responsivo
Saídas	Interface adaptável e otimizada para todos os dispositivos
Ação	Implementar design responsivo com frameworks como Bootstrap

RFS06	
Função	Garantir que a plataforma seja intuitiva e fácil de usar
Descrição	A interface deve ser simples, com fluxo de navegação claro
Entradas	Ações do usuário
Fonte	Requisitos de UX/UI
Saídas	Satisfação do usuário, facilidade de navegação
Ação	Conduzir testes de usabilidade e ajustar a interface com base no feedback

4. CASOS DE USO

Registrar Doação de Item

- **Ator:** Doador.
- **Descrição:** Um doador acessa a plataforma, seleciona a opção de doar um item, insere as informações necessárias e o item é listado para outros usuários.
- **Ações:**
    - O doador faz login na plataforma.
    - Seleciona a opção "Doar Item".
    - Preenche as informações (descrição, fotos, estado de conservação, etc.).
    - O item é registrado no sistema e exibido para receptores.

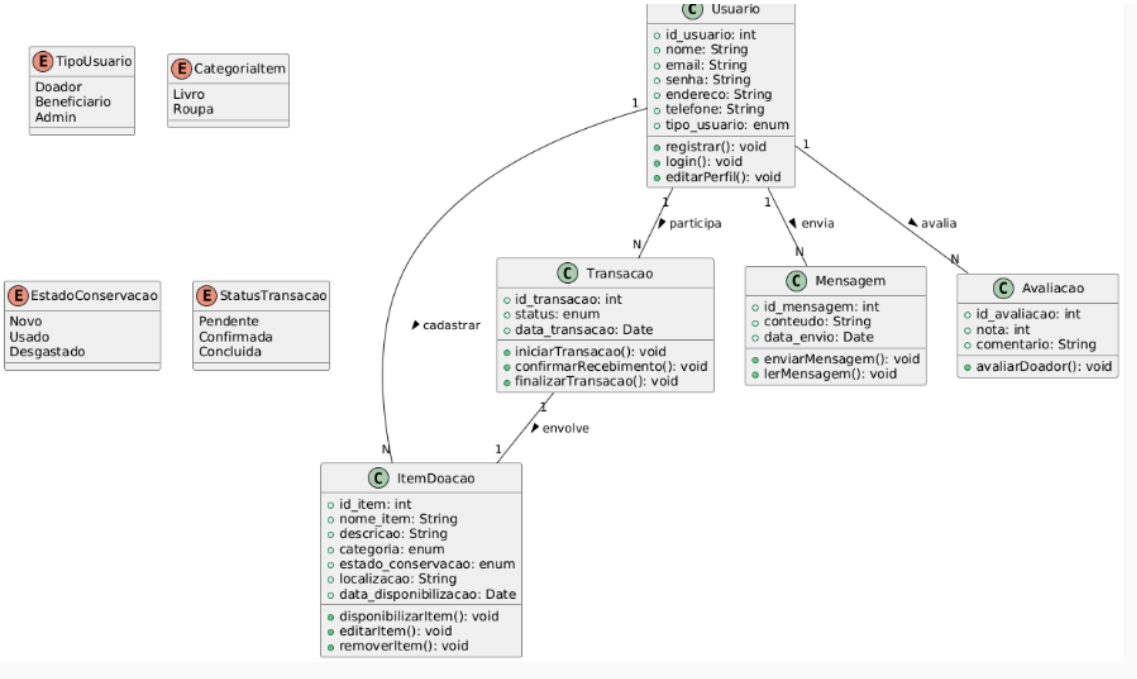
2. Buscar e Solicitar Item para Doação

- **Ator:** Receptor.
- **Descrição:** Um receptor acessa a plataforma, navega pelos itens disponíveis, filtra por interesse e solicita a doação de um item.
- **Ações:**
    - O receptor faz login na plataforma.
    - Utiliza os filtros de busca (roupas, livros, estado, localização).
    - Visualiza um item de interesse e solicita a doação.
    - O doador é notificado e ambos combinam a retirada/entrega.

3. Avaliar Transação

- **Ator:** Doador ou Receptor.
- **Descrição:** Após a transação, os usuários podem avaliar e comentar a experiência da doação.
- **Ações:**
    - O usuário faz login na plataforma.
    - Acessa o histórico de transações e seleciona a opção de avaliação.
    - Insere uma nota e um comentário sobre a experiência.
    - O sistema armazena e exibe a avaliação no perfil do outro usuário

5. ARQUITETURA DO SISTEMA



feito com PlantUML

código:

```
@startuml
!define Enum enum

class Usuario {
  +id_usuario: int
  +nome: String
  +email: String
  +senha: String
  +endereco: String
  +telefone: String
  +tipo_usuario: Enum
  +registrar(): void
  +login(): void
  +editarPerfil(): void
```

```
}

class ItemDoacao {
  +id_item: int
  +nome_item: String
  +descricao: String
  +categoria: Enum
  +estado_conservacao: Enum
  +localizacao: String
  +data_disponibilizacao: Date
  +disponibilizarItem(): void
  +editarItem(): void
  +removerItem(): void
}

class Transacao {
  +id_transacao: int
  +status: Enum
  +data_transacao: Date
  +iniciarTransacao(): void
  +confirmarRecebimento(): void
  +finalizarTransacao(): void
}

class Mensagem {
  +id_mensagem: int
  +conteudo: String
  +data_envio: Date
  +enviarMensagem(): void
  +lerMensagem(): void
}

class Avaliacao {
  +id_avaliacao: int
  +nota: int
  +comentario: String
  +avaliarDoador(): void
}

Enum TipoUsuario {
  Doador
  Beneficiario
  Admin
}

Enum CategoriaItem {
```

```
  Livro
  Roupa
}

Enum EstadoConservacao {
  Novo
  Usado
  Desgastado
}

Enum StatusTransacao {
  Pendente
  Confirmada
  Concluida
}

Usuario "1" -- "N" ItemDoacao : cadastrar >
Transacao "1" -- "1" ItemDoacao : envolve >
Usuario "1" -- "N" Transacao : participa >
Usuario "1" -- "N" Mensagem : envia >
Usuario "1" -- "N" Avaliacao : avalia >
@enduml
```

6. REFERÊNCIAS BIBLIOGRÁFICAS

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