**UNIVERSIDADE DE MOGI DAS CRUZES**

SUSTENTIFY APP

Descarte de Produtos Sustentáveis

**Mogi das Cruzes,**

**SP 2020**

**UNIVERSIDADE DE MOGI DAS CRUZES**

**Guilherme Ferreira Silva**

SUSTENTIFY APP

Descarte de Produtos Sustentáveis

Trabalho de conclusão de curso apresentado ao curso de Sistemas de Informação da Universidade de Mogi das Cruzes como parte dos requisitos para o desenvolvimento de um aplicativo.

**Professores Orientadores**:

Alessandro Aparecido da Silva Horas

Leonardo Cavalcante Alvino

**Mogi das Cruzes,**

**SP 2020**

SUSTENTIFY APP

Descarte de Produtos Sustentáveis

Trabalho de conclusão de curso apresentado ao curso de Sistemas de Informação da Universidade de Mogi das Cruzes como parte dos requisitos para o desenvolvimento de um aplicativo.

Aprovado em: ....................................

**BANCA EXAMINADORA**

Componente da Banca – Titulação, Nome Instituição a que pertence

**BANCA EXAMINADORA**

Componente da Banca – Titulação, Nome Instituição a que pertence

**BANCA EXAMINADORA**

Componente da Banca – Titulação, Nome Instituição a que pertence

RESUMO

Este projeto apresenta o desenvolvimento de uma plataforma digital voltada para a sustentabilidade empresarial, alinhada aos princípios de ESG (Environmental, Social, Governance). A plataforma permitirá que empresas cadastrem produtos sustentáveis destinados ao descarte e encontrem produtos reutilizáveis de interesse, promovendo a economia circular e reduzindo impactos ambientais. A solução, tem como objetivo otimizar processos de descarte sustentável buscando alternativas econômicas alinhados a aquisição e descarte de materiais. Além disso, ajuda a cumprir metas de responsabilidade ambiental, conectando empresas de forma eficiente. Desenvolvido como projeto do curso de Sistemas de Informação, busca demonstrar como a tecnologia pode ser aplicada para estimular práticas sustentáveis, boas práticas com embalagens, geração, cuidado e descarte de plásticos e outros materiais; incentivar a transparência nos negócios e criar um impacto positivo no meio ambiente.

**Palavras-chave**: ESG; Website; Sustentabilidade; Sustentáveis; Meio Ambiente; Preservação; Descarte; Economia Circular; Tecnologia; Impacto;

ABSTRACT

This project presents the development of a digital platform aimed at corporate sustainability, in line with ESG (Environmental, Social, Governance) principles. The platform will allow companies to register sustainable products destined for disposal and find reusable products of interest, promoting the circular economy and reducing environmental impacts. The solution aims to optimize sustainable disposal processes by seeking economic alternatives aligned with the acquisition and disposal of materials. It also helps to meet environmental responsibility targets by connecting companies efficiently. Developed as a project for the Information Systems course, it seeks to demonstrate how technology can be applied to encourage sustainable practices, good practices with packaging, generation, care and disposal of plastics and other materials; encourage transparency in business and create a positive impact on the environment.

**Keywords**: ESG; Website; Sustainability; Sustainable; Environment; Preservation; Disposal; Circular Economy; Technology; Impact;

LISTA DE ILUSTRAÇÔES

Figura 1 - Diagrama de classes 13

Figura 2 - Prototipo de tela 14

Figura 3 - Prototipo de tela 14

Figura 4 - Prototipo de Tela 14

Figura 5 - Diagrama de comunicação dos serviços 15

Figura 6 - Método de criação do produto 16

Figura 7 - Método de solicitação do produto 17

Figura 8 - Script da criação do banco de dados 18

Figura 9 - Script da criação do banco de dados 18

Figura 10 - Script da criação do banco de dados 19

Figura 11 - Classe de comunicação e configuraçao do Chatbot 20

Figura 12 - Pagina para registrar nova empresa 22

Figura 13 - Exibição dos produtos registrados pela empresa 23

Figura 14 - Modal para registrar novo produto 23

Figura 15 - Pagina do produto selecionado 24

Figura 16 - Modal para registro de interesse no produto 24

LISTA DE ABREVIATURAS E SLIGAS

Downtime Tempo em que um sistema ou processo não está operacional;

CSS *Cascading Style Sheet* – estiliza elementos em uma página;

Container Forma leve e portável de empacotar e executar aplicativos;

Docker Orquestrador de containers;

HTTP *Hypertext Transfer Protocol* – protocolo usado para enviar dados entre um navegador web e um site;

HTTPS *Hypertext Transfer Protocol Secure* – versão segura do HTTP, garante criptografia, protegendo informações transmitidas;

CI/CD Automatiza integrações repetitivas;

CI Integração Contínua – automatiza testes e confiabilidade das mudanças no código;

CD Entrega Contínua – liberação automática das mudanças feitas por desenvolvedores;

ESG *Environmental, Social and Governance* (Ambiental, Social e Governança);

Javaagent Ferramenta que roda junto a aplicação Java e pode espiar, modificar ou medir o que ela está fazendo;

Ollama Plataforma que traz modelos de linguagem (LLMs) localmente para o seu dispositivo, sem depender da nuvem;

LLMs *Large language models* – modelos de inteligência artificial, capaz de interpretar textos, conversações e outros tipos de conteúdo;

DTO *Data Transfer Object* – transporte de dados entre diferentes componentes de um sistema;

API *Application Programming Interface* (Interface de Programação de Aplicações)– conjunto de padrões, ferramentas e protocolos que permite a criação mais simplificada e segura de plataformas.

Node.Js Ambiente de execução da linguagem de programação Javascript do lado do servidor.

Github Serviço baseado em nuvem que hospeda projetos, aplicações de desenvolvedores;

Assíncrono Quando os processos podem iniciar mesmo que outros estejam em execução;

**SUMARIO**

1INTRODUÇÃO 7

2 DESENVOLVIMENTO 8

2.1 LEVANTAMENTO DE REQUISITOS 8

2.2 REQUISITOS FUNCIONAIS 8

2.3 REQUISITOS NÃO FUNCIONAIS 9

2.4 FERRAMENTAS E METÓDOS 10

2.4.1 BACKEND 10

2.4.2 FRONTEND 10

2.4.4 OBSERVABILIDADE 10

2.4.5 BANCO DE DADOS 11

2.4.5 CHATBOT 11

2.4.6 INFRAESTRUTURA E DEPLOY 11

3 ANEXOS 13

3.1 DIAGRAMA DE CLASSE 13

3.2 PROTOTIPOS DE TELA 14

3.3 GIT DIAGRAM 15

4 APÊNDICES 16

4.1 FUNCIONALIDADE DE CADASTRO DE PRODUTO 16

4.2 FUNCIONALIDADE DE SOLICITAÇÂO DE PRODUTO 17

4.3 CRIAÇÂO DO BANCO DE DADOS 18

4.4 CHATBOT 20

4.5 MANUAL DE USO 21

4.5.1 BOTÔES DO SISTEMA 21

4.5.2 CADASTRO DE EMPRESA 22

4.5.3 CADASTRO DE PRODUTO 23

4.5.4 SOLICITAR PRODUTO 24

5 CONSIDERAÇÔES FINAIS 25

REFERÊNCIAS 26

# INTRODUÇÃO

A crescente preocupação com a sustentabilidade e a responsabilidade ambiental tem levado empresas a adotarem práticas mais alinhadas aos princípios de ESG (Environmental, Social, Governance).

No entanto, muitas organizações enfrentam desafios na destinação adequada de produtos descartáveis e na busca por insumos reutilizáveis. Com o objetivo de atender a essa demanda, este projeto propõe o desenvolvimento de uma plataforma digital que facilita o cadastro, a busca e a negociação de produtos sustentáveis entre empresas, promovendo a economia circular e reduzindo desperdícios.

Para a implementação da plataforma, foram empregadas tecnologias modernas e escaláveis.

O backend – parte responsável pelo processamento dos dados, lógica de negócios e comunicação com o banco de dados – foi desenvolvido em Java, uma linguagem de programação fortemente tipada que se destaca pela robustez, legibilidade e eficiência na gestão de dados e serviços.

O MySQL foi escolhido como banco de dados devido à sua capacidade de gerenciamento relacional, garantindo precisão e confiabilidade na manipulação das informações.

O frontend – camada responsável pela interface gráfica e interação com o usuário – foi desenvolvido utilizando Angular, um framework robusto e flexível baseado em TypeScript, amplamente utilizado para a criação de aplicações web dinâmicas. O Angular permite a construção de componentes reutilizáveis, além de oferecer uma arquitetura baseada em módulos que facilita a escalabilidade e a manutenção do código. Para garantir uma interface responsiva e moderna, foi utilizado o Tailwind CSS, uma biblioteca de estilos que possibilita a criação de layouts personalizados de forma rápida e eficiente.

Além disso, a plataforma é conteinerizada utilizando Docker, o que facilita a portabilidade e a escalabilidade da aplicação. O Nginx atua como proxy reverso, gerenciando o tráfego entre os serviços de forma eficiente e segura. Para reforçar a segurança e a performance, a solução também utiliza o Cloudflare, uma plataforma de DNS, que protege contra-ataques cibernéticos e otimiza o carregamento das páginas. Com essa combinação de tecnologias, o projeto visa criar uma solução inovadora e eficiente, demonstrando como a tecnologia pode ser aplicada para impulsionar práticas sustentáveis no ambiente corporativo.

# 2 DESENVOLVIMENTO

O projeto Sustentify foi desenvolvido com o objetivo de criar uma plataforma digital para empresas que buscam adotar práticas mais sustentáveis em sua cadeia produtiva. A plataforma facilita o descarte responsável de produtos e a compra de itens que atendem aos critérios de sustentabilidade, conectando empresas interessadas em práticas ecológicas. Segundo Sachs (2015), a sustentabilidade empresarial exige ferramentas práticas que viabilizem a adoção de atitudes ambientais conscientes no setor produtivo. O desenvolvimento foi focado em proporcionar uma experiência eficiente e intuitiva, utilizando tecnologias modernas que garantem a escalabilidade e segurança da aplicação (Bass et al., 2012).

## 2.1 LEVANTAMENTO DE REQUISITOS

Levantamento de requisitos é o processo de identificação das necessidades do sistema, quais são as funcionalidades esperadas e a serem solucionadas. O principal objetivo do levantamento de requisitos é que usuários e desenvolvedores tenham a mesma visão do problema a ser resolvido (Pressman, 2016).

Podemos classificar os requisitos em dois grupos:

* **Requisitos Funcionais (RF):** Especificam o que o sistema deve fazer.
* **Requisitos não Funcionais (RNF):** Indicam características de qualidade que o sistema deve possuir e que estão relacionadas às funcionalidades previstas (Sommerville, 2011).

## 2.2 REQUISITOS FUNCIONAIS

1. **Cadastro de Empresas**: As empresas deverão ser capazes de se cadastrar na plataforma, fornecendo informações como nome, CNPJ, endereço, email, telefone, departamento.
2. **Cadastro de Produtos**: As empresas poderão cadastrar produtos sustentáveis, fornecendo informações como categoria, condição, descrição, imagem, quantidade disponível para descarte ou compra, preço e material.
3. **Busca e Filtro de Produtos**: A plataforma deverá permitir que as empresas filtrem e busquem produtos por categoria, material e condição.
4. **Sistema de Autenticação**: As empresas deverão ter um sistema de login seguro para acessar suas contas e realizar ações na plataforma, como editar seus produtos ou gerenciar solicitações. Para isso, é recomendável o uso de autenticação baseada em tokens JWT, práticas modernas em aplicações web seguras (Stallings, 2020).
5. **Armazenamento de Imagens**: A plataforma deverá armazenar imagens de produtos, incluindo uma imagem principal (thumbnail) e imagens adicionais, garantindo fácil visualização para as empresas.
6. **Envio de Email**: A plataforma, será capaz de enviar e-mails, para validação dos dados da empresa cadastrada, restauração de senha, envio de promoções e notícias referentes a plataforma.
7. **Solicitações**: As empresas poderão fazer uma solicitação do produto escolhido, sendo informado a quantidade desejada.
8. **Dashboard**: A plataforma terá um dashboard visual, para trazer informações de cada produto, além de solicitações da própria empresa.
9. **Chatbot**: Pequena Inteligência Artificial para responder dúvidas sobre a plataforma e o funcionamento dela.

## 2.3 REQUISITOS NÃO FUNCIONAIS

1. **Escalabilidade**: A plataforma deve ser capaz de suportar um número crescente de usuários e produtos, garantindo que a performance não seja impactada conforme o volume de dados aumente. A escalabilidade é essencial em sistemas modernos distribuídos e pode ser obtida com arquiteturas baseadas em microsserviços (Newman, 2015).
2. **Segurança**: O sistema precisa garantir a segurança dos dados das empresas, com uma autenticação robusta e criptografia de informações sensíveis.
3. **Usabilidade**: A plataforma deve ser intuitiva e fácil de usar, com interfaces amigáveis e fluxos claros para todas as funcionalidades, reduzindo a curva de aprendizado dos usuários.
4. **Desempenho**: A plataforma deverá responder rapidamente a requisições dos usuários, com baixa latência, especialmente nas operações de busca e filtro de produtos.
5. **Disponibilidade**: O sistema deve ser altamente disponível, sem downtime, garantindo que as empresas possam acessar a plataforma sem interrupções significativas.
6. **Manutenibilidade**: O código-fonte deve ser bem documentado e modular, facilitando manutenções futuras e a integração de novas funcionalidades. De acordo com Fowler (2004), a modularização e o uso de boas práticas de engenharia contribuem diretamente para a manutenibilidade de sistemas complexos.

## 2.4 FERRAMENTAS E METÓDOS

Para o desenvolvimento da Sustentify, foi escolhida uma Stack tecnológica moderna, robusta, com o objetivo de garantir escalabilidade, manutenção, performance e boa experiência para o usuário final.

### 2.4.1 BACKEND

A linguagem **Java** foi escolhida pela sua maturidade, estabilidade e suporte extenso para aplicações corporativas. O projeto pode ser expandido com novos recursos, além da necessidade de escalar à medida que mais empresas e produtos entram, **Java** consegue isso sem comprometer a consistência. A linguagem tem excelente suporte para monitoramento, métricas, logs e tracing distribuído, essencial para diagnosticar problemas futuros.

### 2.4.2 FRONTEND

O **Angular** foi adotado como framework para o desenvolvimento do frontend devido à sua estrutura organizada, necessidade de um SPA onde a penas os dados mudam, não a estrutura da página, proporcionando uma navegação mais rápida, forte separação entre lógica e visual, além de utilizar classes e anotações, combinando bem com a arquitetura do Java.

### 2.4.4 OBSERVABILIDADE

Por utilizarmos o Java como linguagem, é possível integrar de forma simples ferramentas para observabilidade. Utilizando-se do **javaagent** da distribuição **opentelemetry-java-instrumentation**, conseguimos a observabilidade automática da aplicação, sem necessidade de modificar o código-fonte. As requisições HTTP, interações com banco de dados e outras operações são automaticamente exportadas para ferramentas de análise e visualização, como:

* **Jaeger**: Responsável por exibir requisições, identificando tempos de resposta, gargalos e falhas.
* **Prometheus**: Utilizado para coletar métricas numéricas como uso de CPU, memória, latência, fornecendo uma visão da saúde da aplicação.

Com isso podemos acompanhar como a aplicação está desempenhando, garantir tempos de respostas mais rápidos e melhoria contínua.

### 2.4.5 BANCO DE DADOS

Como banco de dados foi escolhido o **MySQL**, pois tem grande compatibilidade com o ecossistema Java. Como requisito funcional temos filtros de produtos então temos a necessidade de garantir **consistência forte dos dados** e realizar **consultas complexas** que envolvem múltiplas tabelas e filtros, algo que o MySQL realiza de maneira eficiente.

### 2.4.5 CHATBOT

Para implementação do chatbot na Sustentify, foi escolhido a biblioteca **Ollama**, integrada com uma aplicação em **Node.Js**. A escolha do **Ollama** se deu pela sua capacidade de executar modelos de linguagem (**LLMs**) localmente, de maneira gratuita e sem depender de **APIs** externas.

O **Node.js** foi utilizado por ser leve e **assíncrono**, essencial para um chatbot. Ele permite a criação de servidores HTTP rápidos e compatíveis com bibliotecas de IA.

### 2.4.6 INFRAESTRUTURA E DEPLOY

Para atender aos objetivos funcionais, foram adotadas ferramentas modernas de infraestrutura e entrega contínua, como **Docker**, **Nginx, Ngrok** e **GitHub Actions**.

A escolha do **Docker** permite que a aplicação seja empacotada em containers, garantindo que ela funcione da mesma forma em qualquer ambiente. Cada serviço da Sustentify (frontend, backend, banco de dados, chatbot) é executado em seu próprio container, o que facilita o isolamento, a escalabilidade e a manutenção dos componentes.

Para o gerenciamento de acesso ao serviço frontend, foi utilizado o **Nginx**. Ele atua como ponto de entrada para as aplicações, roteando e protegendo as requisições. Além de garantir segurança servindo HTTPS.

O **Ngrok** plataforma que expõe serviços que estão sendo executados localmente (do seu computador) para a Internet, de forma rápida e segura, foi utilizado para servir tanto o Chatbot, quanto o Backend.

Como a plataforma já está hospedada no **GitHub**, foi possível integrar de forma simples o processo de Integração e Entrega Contínua (CI/CD) utilizando o **GitHub Actions**. A cada atualização no repositório, os serviços são automaticamente reconstruídos, enviados para o **Docker Hub** (repositório de containers) e implantados no servidor, garantindo que as alterações entrem em produção de forma rápida e segura.

# 3 ANEXOS

## 3.1 DIAGRAMA DE CLASSE

Figura 1 - Diagrama de classes

**Linha do tempo

O conteúdo gerado por IA pode estar incorreto.**

**Fonte: Desenvolvimento Próprio (2025)**

## 3.2 PROTOTIPOS DE TELA

Figura 2 - Prototipo de tela

Site

O conteúdo gerado por IA pode estar incorreto.

Figura 3 - Prototipo de tela

Interface gráfica do usuário, Texto, Aplicativo

O conteúdo gerado por IA pode estar incorreto.

Figura 4 - Prototipo de Tela

Interface gráfica do usuário, Site

O conteúdo gerado por IA pode estar incorreto.

## 3.3 GIT DIAGRAM

Figura 5 - Diagrama de comunicação dos serviços

Diagrama

O conteúdo gerado por IA pode estar incorreto.

**Fonte: GitDiagram (2025)**

# 4 APÊNDICES

Os apêndices deste projeto têm como objetivo complementar as informações apresentadas ao longo do trabalho, fornecendo detalhes técnicos relevantes para a compreensão e reprodução do sistema desenvolvido.

## 4.1 FUNCIONALIDADE DE CADASTRO DE PRODUTO

A funcionalidade de cadastro de produto sustentável é uma das principais operações da plataforma Sustentify, sendo responsável por registrar os produtos que serão disponibilizados pelas empresas para reutilização, descarte sustentável ou negociação com outras organizações.

O codigo abaixo apresenta o método principal de cadastro, implementado na camada de serviço da aplicação backend.

Figura 6 - Método de criação do produto

Texto

O conteúdo gerado por IA pode estar incorreto.

**Fonte: Desenvolvimento Próprio (2025)**

Esse método recebe um objeto ***RegisterProductDto***, que contém os dados preenchidos pela empresa no formulário de cadastro, além de uma referência à empresa autenticada. Entre os dados cadastrados, destacam-se:

* **Nome:** Para identificação e pesquisa do produto;
* **Categoria**: Classifica o tipo de produto (ex.: eletrônico, plástico, papel, metal etc.);
* **Material**: Identifica a composição principal do item (ex.: alumínio, metal, vidro etc.);
* **Condição**: Estado de conservação (novo, usado, danificado etc.);
* **Data de produção**: Data da criação do produto;
* **Localização:** Localidade do produto, facilitando o planejamento logístico;
* **Data de Descarte**: Data do cadastro do produto na plataforma, colocado automaticamente;
* **Preço:** Preço de exibição do produto;
* **Quantidade:** Quantidade disponível do produto;
* **Descrição:** Para contexto de uso e informações adicionais;

## 4.2 FUNCIONALIDADE DE SOLICITAÇÂO DE PRODUTO

A funcionalidade de solicitação de produto da plataforma, permite que empresas demonstrem interesse em adquirir produtos cadastrados por outras organizações. Esse recurso é fundamental para empresas se conectarem e realizarem negociações de forma agil.

Figura 7 - Método de solicitação do produto

Texto

O conteúdo gerado por IA pode estar incorreto.

**Fonte: Desenvolvimento Próprio (2025)**

Esse método é chamado quando uma empresa manifesta interesse em um produto listado na plataforma. Ele recebe como parâmetros:

* A empresa compradora (*company*), identificada a partir da autenticação;
* O produto de interesse (*product*);
* Um **DTO** com os dados da solicitação (*RegisterInterestProductDto*), contendo a quantidade desejada e uma mensagem opcional.

Essa funcionalidade permite às empresas, a avaliar propostas e organizar negociações. Com isso, a Sustentify estimula conexões sustentáveis e eficientes no ciclo de vida dos produtos.

## 4.3 CRIAÇÂO DO BANCO DE DADOS

Utilizamos como banco de dados o MySQL para armazenar, consultar e manter a integridade dos dados relacionados ás empresas, produtos e interações entre os usuários. A escolha se deu por sua consistente aplicação relacional, pois a integridade referencial era um fator predominante para o projeto.

Figura 8 - Script da criação do banco de dados

Texto

O conteúdo gerado por IA pode estar incorreto.

**Fonte: Desenvolvimento Próprio (2025)**

Figura 9 - Script da criação do banco de dados

Texto

O conteúdo gerado por IA pode estar incorreto.

**Fonte: Desenvolvimento Próprio (2025)**

Figura 10 - Script da criação do banco de dados

Texto

O conteúdo gerado por IA pode estar incorreto.

**Fonte: Desenvolvimento Próprio (2025)**

* ***companies***: Tabela que armazena os dados das empresas cadastradas;
* ***companies\_deleted***: Estrutura semelhante à tabela ***companies***, criada para manter um histórico de empresas excluídas
* ***products***: Armazena os produtos disponibilizados pelas empresas;
* ***products\_images* e *products\_thumbnail****:* Tabelas auxiliares que armazenam URLs de imagens de produto;
* ***company\_document\_images****:* Tabela que armazena documentação da empresa, para validação de identidade e conformidade com a plataforma;
* ***interested\_products****:* Representa a manifestação de interesse de uma empresa em um produto publicado por outra;

## 4.4 CHATBOT

Ao receber uma pergunta do usuário, o sistema lê um prompt base de um arquivo (prompt.txt) que define o contexto e as instruções para o modelo responder as perguntas. Em seguida, essa pergunta é enviada para a API do Ollama via uma requisição HTTP POST, junto ao prompt e configurações predefinidas para garantir respostas assertivas.

Figura 11 - Classe de comunicação e configuraçao do Chatbot

Texto

O conteúdo gerado por IA pode estar incorreto.

**Fonte: Desenvolvimento Próprio (2025)**

## 4.5 MANUAL DE USO

Este capítulo tem por objetivo detalhar como o usuário deve proceder para a execução das principais funcionalidades disponibilizadas na Sustentify.

### 4.5.1 BOTÔES DO SISTEMA

**Tabela 01 – Descritivo dos botões da plataforma**

|  |  |  |
| --- | --- | --- |
| BOTÕES DO SISTEMA | | |
| Representação | Nome do botão ou ícones | Descrição da ação |
|  | Botão Menu | Apresenta um menu de acesso principal do sistema |
|  | Botão Login | Apresenta a tela de login, para o usuário entrar no sistema |
|  | Botão Contato | Apresenta a tela de contato, para enviar um email a Sustentify |
|  | Botão Sobre | Apresenta a tela, explicando sobre a Sustentify |
|  | Botão Produtos | Ao clicar em produtos, você escolhe o filtro de produtos, para exibir produtos de acordo com o filtro especificado. |
|  | Botão Categoria | Exibe as Categorias de produtos disponiveis. |
|  | Botão Condição | Exibe as Condições de produtos disponiveis. |
|  | Botão Material | Exibe os Materiais de produtos disponiveis. |
|  | Botão Buscar | Realiza a busca de produtos de acordo com os filtros passados. |
|  | Botões de Paginação | Seleciona as paginas, podendo ir para a proxima ou anterior. |
|  | Botão de Deleção | Realiza a deleção do item selecionado. |
|  | Botão do Produto | Apresenta os detalhes do produto selecionado. |
|  | Botão de Logout | Realiza o logout do usuário, saindo da sessão e levando até o login |

### 4.5.2 CADASTRO DE EMPRESA

Entre na página de */signup* e coloque os dados solicitados. É necessário colocar todos os dados, incluindo documentação para validação e verificação de fraudes.

Figura 12 - Pagina para registrar nova empresa

Interface gráfica do usuário, Texto, Aplicativo

O conteúdo gerado por IA pode estar incorreto.

**Fonte: Desenvolvimento Próprio (2025)**

### 4.5.3 CADASTRO DE PRODUTO

Para realizar o cadastro de novos produtos, é necessário ter autorização da plataforma para realizar a ação. Com isso podemos entrar na dashboard e clicar no botão de título **Novo Produto.**

Figura 13 - Exibição dos produtos registrados pela empresa

Interface gráfica do usuário, Aplicativo

O conteúdo gerado por IA pode estar incorreto.

**Fonte: Desenvolvimento Próprio (2025)**

Figura 14 - Modal para registrar novo produto

Interface gráfica do usuário, Aplicativo

O conteúdo gerado por IA pode estar incorreto.

**Fonte: Desenvolvimento Próprio (2025)**

### 4.5.4 SOLICITAR PRODUTO

Para realizar uma solicitação de produto interessado, entre na página do produto, utilize o scroll de rolagem para descer até encontrar o botão de **solicitação**, clicando nele aparecerá o modal, insira as informações necessárias e realize o cadastro.

Figura 15 - Pagina do produto selecionado

Interface gráfica do usuário, Texto, Aplicativo, chat ou mensagem de texto, Site

O conteúdo gerado por IA pode estar incorreto.

**Fonte: Desenvolvimento Próprio (2025)**

Figura 16 - Modal para registro de interesse no produto

Interface gráfica do usuário, Texto, Aplicativo, chat ou mensagem de texto

O conteúdo gerado por IA pode estar incorreto.

**Fonte: Desenvolvimento Próprio (2025)**

# 5 CONSIDERAÇÔES FINAIS

O desenvolvimento da plataforma, representou uma oportunidade de aplicar conhecimentos, tanto de cursos e experiencias extras, quantos adquiridos ao longo do curso de Sistemas de Informação, em um projeto com impacto social, ambiental e tecnológico. A proposta de conectar empresas por meio de reutilização e descarte sustentável se mostrou viável, pois sustentabilidade é um assunto muito pertinente atualmente e vem crescendo cada vez mais.

A importância de ter um ambiente sustentável não só contribui para o meio ambiente, mas para a sociedade e pessoas envolvidas em sua operação, assim como para as gerações futuras. Ela reflete em longevidade e até mesmo na forma como clientes e investidores a enxergam.

Um dos principais desafios enfrentados foi a adoção da linguagem Java. Apesar de já possuir experiência com outras linguagens, foi necessário aprender Java com Spring do zero, para implementar o backend. Esse processo não apenas ampliou minha visão sobre linguagens fortemente tipadas e o ecossistema Java, como também fortaleceu minha capacidade de aprender rapidamente novas tecnologias.

Outro marco foi a integração de modelos de linguagem (LLMs), algo completamente novo, que residiu com muita curiosidade e resiliência. Pude explorar na prática conceitos de inteligência artificial generativa, prompt engineering e comunicação com APIs de IA. Essa experiência demonstrou o potencial das LLMs em aplicações reais, como também abriu portas para futuras explorações nessa área.

Com o projeto, foi possível aprofundar habilidades em desenvolvimento de software, DevOps, integração contínua, modelagem de dados e arquitetura de sistemas, além de lidar com desafios reais de escopo, segurança, usabilidade e desempenho.

Por fim, a Sustentify se apresenta como uma solução promissora que pode ser expandida e aprimorada, contribuindo para práticas empresariais mais sustentáveis e conscientes.

# REFERÊNCIAS

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. **Coletânea de normas técnicas:**

elaboração de TCC, dissertação e tese. São Paulo: ABNT, 2011. 76 p.

* <https://medium.com/@lcoldebella/java-%C3%A9-a-melhor-op%C3%A7%C3%A3o-para-servi%C3%A7os-banc%C3%A1rios-e-financeiros-7e28f48dcbfd>
* <https://www.java.com/pt-BR/download/help/whatis_java.html>
* <https://www.totvs.com/blog/developers/back-end/>
* [O que é um Downtime | Conheça suas causas e como evitá-los!](https://www.opservices.com.br/o-que-e-um-downtime/)
* ANDRESS, J. *The Basics of Information Security: Understanding the Fundamentals of InfoSec*. Syngress, 2014.
* BASS, L.; CLEMENTS, P.; KAZMAN, R. *Software Architecture in Practice*. Addison-Wesley, 2012.
* FOWLER, M. *Refactoring: Improving the Design of Existing Code*. Addison-Wesley, 2004.
* NIELSEN, J. *Usability Engineering*. Morgan Kaufmann, 1994.
* NEWMAN, S. *Building Microservices: Designing Fine-Grained Systems*. O'Reilly Media, 2015.
* PRESSMAN, R. S. *Engenharia de Software: uma abordagem profissional*. McGraw-Hill Brasil, 2016.
* SACHS, I. *Caminhos para o desenvolvimento sustentável*. Garamond, 2015.
* SOMMERVILLE, I. *Engenharia de Software*. 9. ed. São Paulo: Pearson, 2011.
* STALLINGS, W. *Cryptography and Network Security: Principles and Practice*. 8th ed. Pearson, 2020.