

<Nome do Projeto>

Especificação de Sistema

EasyRecycle

Versão 1.0

**Andrey De Freitas Souza**

**Erick Garcia Andrighetti**

**Gabrielle Garcia Paz**

**Giovanna Kunioshi**

**Gustavo Vicente Domingues**

**Kevin Rodrigues Garcia**

**Leonardo Souza Borda**

**Pedro Henrique Marques da Silva**

Histórico da Revisão

|  |  |  |  |
| --- | --- | --- | --- |
| **Data** | **Versão** | **Descrição** | **Autor** |
| 10/04/2025 | 1.0 | Estruturação inicial do tópico 1 ao 2,inclusão das informações referentes aos integrantes do projeto, professores, disciplinas e dos requisitos do projeto | Andrey de Freitas Souza |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Índice Analítico

1. Identificação 4

1.1 Nome e RA: 4

*1.2* Turma: 4

1.3 Professor: 4

1.4 Disciplinas envolvidas: 4

1.4.1 Modelos, métodos e técnicas da engenharia de software: 4

1.4.2 Gestão e qualidade de software: 4

2. Especificação Geral do Sistema. 4

2.1 Escopo 4

2.1.1 Componentes Internos: 5

2.1.2 Módulo de autenticação e gerenciamento de usuários 5

2.1.3 Componentes Externos: 5

2.1.4 Hardware: 5

2.1.5 Software: 5

2.2 Descrição dos requisitos de usuário. 5

2.2.1 Breve Descrição 5

2.2.2 Requisitos do produto 6

3. Distribuição das Funcionalidades (Modelagem). 7

3.1 Funcionalidades do Sistema 8

3.1.1 Descrição da Funcionalidade-1 do Sistema 8

3.1.2 Descrição da Funcionalidade-2 do Sistema 8

4. Representação Gráfica do Software. 8

4.1 Diagrama de Casos de Uso 8

4.1.1 Especificação do Caso de Uso-1 8

4.1.2 Especificação do Caso de Uso-2 8

4.2 Diagrama de Classes. 8

4.3 Diagrama de Sequência. 8

4.3.1 Diagrama de Sequência-1. 8

4.3.2 Diagrama de Sequência-2. 8

4.4 Diagrama de Comunicação/Colaboração. 8

4.5 Diagrama de Atividades. 9

4.6 Diagrama de Estados. 9

4.6.1 Diagrama de Estados da classe-x atributo-x. 9

4.6.2 Diagrama de Estados da classe-y atributo-y. 9

4.7 Diagrama de Implementação. 9

4.7.1 Diagrama de Pacotes. 9

4.7.2 Diagrama de Componentes. 9

4.7.3 Diagrama de Implantação/Distribuição 9

4.8 Diagrama de Objetos. 9

4.9 Diagrama de Estrutura Composta. 9

4.10 Diagrama de Interação Geral. 9

4.11 Diagrama de Tempo ou Temporal. 9

EasyRecycle

# Identificação

## Nome e RA:

|  |  |
| --- | --- |
| **Nome** | **RA** |
| Andrey De Freitas Souza | 823217536 |
| Erick Garcia Andrighetti | 823144062 |
| Gabrielle Garcia Paz | 823126085 |
| Giovanna Kunioshi | 822145564 |
| Gustavo Vicente Domingues | 823162177 |
| Kevin Rodrigues Garcia | 823155644 |
| Leonardo Souza Borda | 823153965 |
| Pedro Henrique Marques da Silva | 823112663 |

## Turma:

## Professor:

Magda Aparecida Silverio Miyashiro e Sidny de Almeida Molina Pereira

## Disciplinas envolvidas:

[Indicar o nome das disciplinas, repetir de acordo com o número de disciplinas envolvidas]

### Modelos, métodos e técnicas da engenharia de software:

[Onde consta Disciplina alterar para o nome da disciplina]

#### Professor responsável:

Magda Aparecida Silverio Miyashiro

[Nome do professor da disciplina]

#### Ponto a ser verificado:

[Solicitação do professor da disciplina para ser desenvolvida no projeto]

### Gestão e qualidade de software:

[Onde consta Disciplina alterar para o nome da disciplina]

#### Professor responsável:

Sidny de Almeida Molina Pereira

[Nome do professor da disciplina]

#### Ponto a ser verificado:

[Solicitação do professor da disciplina para ser desenvolvida no projeto]

Etc......

# Especificação Geral do Sistema.

## Escopo

Este documento tem como objetivo descrever a especificação do sistema Easy Recycle, uma plataforma web voltada à educação ambiental, à promoção da reciclagem e à análise de impacto ambiental individual e coletivo.

O escopo desta especificação abrange todas as funcionalidades, fluxos e componentes técnicos necessários para o desenvolvimento e a operação do sistema, desde a interface de usuário até a camada de persistência de dados. O Easy Recycle será acessível por navegadores web e contará com um front-end responsivo, desenvolvido em HTML, CSS e JavaScript, além de um back-end desenvolvido em Python, responsável pelo processamento das lógicas de negócio, integração com o banco de dados e execução das funcionalidades dinâmicas do sistema.

O armazenamento e gerenciamento de dados será realizado através do MySQL Workbench, que permitirá o controle eficiente de informações dos usuários, registros de simulações, dados estatísticos e conteúdos educativos.

Componentes afetados/influenciados por esta especificação:

### Componentes Internos:

#### Interface do usuário (front-end responsivo)

#### API e lógicas de negócio (Python)

#### Banco de dados relacional (MySQL)

### Módulo de autenticação e gerenciamento de usuários

#### Simulador de impacto ambiental personalizado

#### Dashboard de dados regionais

### Componentes Externos:

#### Navegador web (acesso via cliente)

#### Bibliotecas externas (ex: Chart.js para gráficos, bibliotecas Python como Flask/Django)

#### Fontes de dados externa

### Hardware:

#### Servidor de hospedagem para o sistema (nuvem ou servidor dedicado)

#### Dispositivos de acesso (computadores, smartphones, tablets)

### Software:

#### Navegadores modernos compatíveis (Chrome, Firefox, Edge, Safari)

#### IDEs para desenvolvimento (VSCode, PyCharm)

#### Ferramentas de gerenciamento de banco de dados (MySQL Workbench)

Esta especificação define os limites do sistema, seu comportamento esperado e os requisitos para seu funcionamento adequado. Qualquer modificação em sua estrutura, arquitetura ou escopo funcional deve ser avaliada em conformidade com este documento.

[Uma breve descrição do escopo desta **Especificação de Sistema**, e de tudo o que for afetado ou influenciado por este documento, componentes externos, internos, hardware e software]

## Descrição dos requisitos de usuário.

### Breve Descrição

O **Easy Recycle** é uma plataforma digital desenvolvida com o objetivo de promover a educação ambiental e facilitar o processo de separação e descarte correto de resíduos recicláveis por parte da população. A aplicação visa unir tecnologia, dados e conscientização em uma solução centralizada e escalável.

[A descrição aborda brevemente o papel e a finalidade do sistema/software segundo a perspectivas do usuário]

### Requisitos do produto

[A descrição dos requisitos do produto são referentes a solicitação do usuário, todos os desejos do usuário, independentemente da arquitetura do sistema]

#### Requisitos 1 RF01 – Cadastro de Usuário:

O sistema deve permitir que o usuário realize cadastro com informações básicas (nome, email, localização).

[Descrição de cada requisito do sistema]

#### Requisitos 2 RF02 – Autenticação de Usuário:

O sistema deve permitir login e logout com segurança.

#### Requisitos 3 RF03 – Exibição de Conteúdo Educacional:

O sistema deve exibir informações educativas sobre como separar corretamente os resíduos recicláveis.

#### Requisitos 4RF03 – Exibição de Conteúdo Educacional:

O sistema deve exibir informações educativas sobre como separar corretamente os resíduos recicláveis.

#### Requisitos 5 RF04 – Visualização de Dados de Reciclagem Local:

O sistema deve exibir dados sobre reciclagem da cidade do usuário, com filtros por período (semanal, mensal, anual).

#### Requisitos 6 RF05 – Exibição de Impactos Ambientais:

O sistema deve apresentar informações sobre os benefícios ambientais da reciclagem.

#### Requisitos 7 RF06 – Simulador de Impacto Pessoal:

O sistema deve permitir que o usuário informe a média de itens reciclados por semana e receba uma estimativa de impacto ambiental gerado.

#### Requisitos 8 RF07 – Localização Geográfica:

O sistema deve identificar ou permitir a seleção da cidade do usuário para personalizar os dados regionais.

#### Requisitos 9 RF08 – Responsividade:

O sistema deve se adaptar a diferentes tamanhos de tela (mobile, tablet, desktop).

#### Requisitos 10 RNF01 – Usabilidade:

A interface deve ser intuitiva e de fácil navegação, mesmo para usuários com baixo nível de letramento digital.

#### Requisitos 11 RNF02 – Desempenho:

As páginas e funcionalidades devem carregar em até 3 segundos com conexão padrão.

#### Requisitos 12 RNF03 – Segurança:

O sistema deve proteger os dados dos usuários utilizando criptografia e autenticação segura.

#### Requisitos 13 RNF04 – Escalabilidade:

A aplicação deve suportar aumento de usuários simultâneos sem perda significativa de desempenho.

#### Requisitos 14 RNF05 – Disponibilidade:

O sistema deve estar disponível para acesso 24/7 com no mínimo 99% de uptime mensal.

#### Requisitos 15 RNF06 – Manutenibilidade:

O código-fonte deve seguir boas práticas de organização e documentação para facilitar futuras manutenções.

#### Requisitos 16 RNF07 – Compatibilidade:

O sistema deve ser compatível com os navegadores modernos (Chrome, Firefox, Safari, Edge).

#### Requisitos 17 RI01

O sistema não deve permitir acesso a dados de reciclagem de outras cidades sem autorização.

#### Requisitos 18 RI02

O sistema não deve permitir que o simulador funcione sem entrada de dados válidos.

#### Requisitos 19 RI03

O sistema não deve apresentar informações técnicas complexas ao usuário final leigo.

#### Requisitos 20 RI04

O sistema não deve salvar dados do usuário sem consentimento expresso.

#### Requisitos 21 RI05

O sistema não deve expor visualizações de dados de impacto ambiental sem fonte ou base científica.

# Distribuição das Funcionalidades (Modelagem).

[ (explicação deste tópico 4) E a atividade de construção de modelos que explique as características e o comportamento da parte software e da parte hardware do Sistema a ser desenvolvido. Em sua construção os modelos devem ser usados na elaboração e representação da arquitetura das características e funcionalidades que o Sistema deve prover e contribui para o seu planejamento e na sua construção. Em geral implica na construção de modelos gráficos que simbolizam os artefatos dos Sistemas de software e hardware utilizados e os seus inter-relacionamentos. A escolha da técnica e da ferramenta a ser utilizada, deve ser escolhida de acordo com as características do Sistema a ser desenvolvido com as características do ambiente de desenvolvimento e dos envolvidos. Pode-se utilizar representação através de fluxogramas, linguagens gráficas, ferramentas e técnicas direcionadas para o Sistema a ser produzido. ]

## Funcionalidades do Sistema

[Listar as funcionalidades do sistema identificadas no enunciado]

### Descrição da Funcionalidade-1 do Sistema

[Descrever cada funcionalidade do sistema pela visão do software]

### Descrição da Funcionalidade-2 do Sistema

[Descrever cada funcionalidade do sistema pela visão do software]

# Representação Gráfica do Software.

## Diagrama de Casos de Uso

[Representar o conjunto de comportamentos de alto nível do sistema - A Representação gráfica das funcionalidades do sistema]

### Especificação do Caso de Uso-1

[Especificação/detalhamento de cada caso de uso do sistema, o detalhamento deve ser feito de forma textual e pode ser demonstrado através de outros diagramas]

### Especificação do Caso de Uso-2

[Especificação/detalhamento de cada caso de uso do sistema, o detalhamento deve ser feito de forma textual e pode ser demonstrado através de outros diagramas]

## Diagrama de Classes.

[Representar a coleção de classes do sistema e seus inter-relacionamentos]



## Diagrama de Sequência.

[Representar uma perspectiva, orientada por tempo, da colaboração/troca de mensagens entre os objetos do sistema, para cada instancia/cenário diferentes do sistema]

### Diagrama de Sequência-1.

[Representação do cenário-1]

### Diagrama de Sequência-2.

[Representação do cenário-2]

## Diagrama de Comunicação/Colaboração.

[Representar o envio ou o recebimento de dados entre um ator e um caso de uso]

## Diagrama de Atividades.

[Representar o fluxo de tarefas que podem ser executadas pelo sistema ou por um ator]

## Diagrama de Estados.

[Representar um conjunto de estados dos objetos das classes]

### Diagrama de Estados da classe-x atributo-x.

[Representar um conjunto de estados que um objeto pode estar e os “gatilhos” que estimulam a transição do objeto de um estado para outro]

### Diagrama de Estados da classe-y atributo-y.

[Representar um conjunto de estados que um objeto pode estar e os “gatilhos” que estimulam a transição do objeto de um estado para outro]

## Diagrama de Implementação.

[Representar a Arquitetura Física do sistema]

### Diagrama de Pacotes.

[Representar os pacotes ou pedaços do sistema divididos em agrupamentos lógicos mostrando as dependências entre eles, representaros módulos físicos que o desenvolvedor deve utilizar]

### Diagrama de Componentes.

[Representar a coleção de componentes de software e seus inter-relacionamentos]

### Diagrama de Implantação/Distribuição

[Representar a arquitetura dos processadores, dispositivos e os componentes de software que rodam na arquitetura utilizada no sistema, a modelagem das plataformas físicas e conexões de rede que do sistema]

## Diagrama de Objetos.

[Representar os objetos que foram instanciados das classes]

## Diagrama de Estrutura Composta.

[Representar os relacionamentos entre os elementos/colaboração interna de classes, interfaces ou componentes para especificar uma funcionalidade do sistema]

## Diagrama de Interação Geral.

[Representar o envio ou o recebimento de dados entre um ator e um caso de uso]

## Diagrama de Tempo ou Temporal.

[Representar o comportamento dos objetos e sua interação em uma escala de tempo, focalizando as condições que mudam no decorrer desse período]