



UNIVERSIDADE SÃO JUDAS TADEU

GABRIEL NEULES GOMES RODRIGUES SOARES - RA 822167394

LUCAS VASCONCELLOS RAMOS DE SOUSA - RA 8222242697

PALOMA LOPES DE SOUSA - RA 822167506

WELLERSON RESENDE MONTEIRO - RA 8222243349

PROJETO A3 – GESTÃO E QUALIDADE
DE SOFTWARE ECOTROCA

SÃO PAULO

2025

UNIVERSIDADE SÃO JUDAS TADEU

GABRIEL NEULES GOMES RODRIGUES SOARES - RA 822167394

LUCAS VASCONCELLOS RAMOS DE SOUSA - RA 8222242697

PALOMA LOPES DE SOUSA - RA 822167506

WELLERSON RESENDE MONTEIRO - RA 8222243349

PROJETO A3 – GESTÃO E QUALIDADE
DE SOFTWARE ECOTROCA

SÃO PAULO

2025

Sumário

1. INTRODUÇÃO	5
1.1. Tema	5
1.2. Objetivos a serem alcançados	5
1.2.1 Objetivo Geral:	5
1.2.2 Objetivos Específicos:	6
2. PLANEJAMENTO DE TESTE DE SOFTWARE	6
2.1. Cronograma de atividades	7
2.2. Alocação de recursos	8
2.3. Marcos do projeto	9
3. DOCUMENTOS DE DESENVOLVIMENTO DE SOFTWARE	9
3.1 Plano de Projeto: ECOTROCA	9
3.1.1. Planejamento do projeto	10
3.1.2. Escopo	10
3.1.3. Recursos	11
3.1.4. Estimativas de projeto	12
3.2 Documentos de requisitos	13
3.2.1 Requisitos funcionais	13
3.2.2 Requisitos não funcionais	14
3.2.3 Regras de negócio	15
3.2.4 Casos de uso	16
3.3 Planejamento de testes	18
3.3.1 Plano de Testes	18
3.3.1.1 Introdução	18
3.3.1.2 Escopo	18
3.3.1.3 Objetivos	18
3.3.1.4 Requisitos a serem testados	18
3.3.1.5 Estratégias, tipos de testes e ferramentas	19
3.3.1.6 Recursos a serem empregados	19
3.3.1.7 Cronograma das Atividades	19
3.3.1.8 Definição dos Marcos do Projeto	19
3.3.2 Casos de Testes	20
3.3.3 Roteiro de Testes	21
4. GESTÃO E CONFIGURAÇÃO DE SOFTWARE	21
4.1. Aplicação dos Conceitos de GCS.	22
4.1.1 Identificação de Itens de Configuração (ICs)	22
4.1.2 Controle de Versão	22
4.1.3 Controle de Mudanças	22
4.1.4 Auditoria e Status de Configuração	22
4.1.5 Reprodutibilidade	23
4.2 Estrutura do projeto	23
5. DOCUMENTOS DE DESENVOLVIMENTO DE SOFTWARE	24
Ferramentas Utilizadas:	24

Estrutura de Branches:	24
5.1. Organização em Pacotes (Packages)	24
5.2. Separação de Responsabilidades (SRP - Single Responsibility Principle)	24
5.3. README.md	25
5.4. CHANGELOG.md	25
5.5. Gitignore	25
5.6. Versionamento com Git	25
5.7. Código Modular e Reutilizável	25
5.8. Preparação para Expansão	25
5.9. Nomenclatura Clara e Consistente	26
5.10. Separação de Configurações	26
5.11 Políticas de Merge	26
5.12 Tags e Releases	26
REFERÊNCIA BIBLIOGRÁFICA	27

1. INTRODUÇÃO

O presente trabalho descreve o Ecotroca, um software desenvolvido com o propósito de incentivar a reciclagem e o descarte adequado de materiais, alinhado aos Objetivos de Desenvolvimento Sustentável (ODS) estabelecidos pela Organização das Nações Unidas (ONU). O software contribui de forma significativa para os ODS 12 (Consumo e Produção Sustentáveis) e 13 (Ação contra a Mudança Global do Clima), ao promover padrões sustentáveis de consumo, redução do desperdício, gestão adequada de resíduos e combate às mudanças climáticas.

Por meio do uso do Ecotroca, os EcoCidadãos são incentivados a participar ativamente na promoção da sustentabilidade e na redução do impacto ambiental. A documentação apresenta detalhadamente o software, incluindo seu propósito, métodos de trabalho e estratégias para engajar a comunidade em práticas sustentáveis. Além disso, são exploradas formas de aprimorar o Ecotroca, buscando constantemente novas maneiras de envolver a comunidade e ampliar sua influência positiva.

O Ecotroca faz parceria com EcoParceiros do setor privado para que elas criem seus próprios pontos de coleta em parceria com o aplicativo. Fornecemos todo o suporte logístico e know-how através de nossos especialistas para que a EcoParceiro possa ter e gerir seu primeiro ponto de coleta de resíduos.

1.1. Tema

Documentação do desenvolvimento de software para promover o descarte sustentável de resíduos, através de um sistema de pontuação e recompensas em cashback.

1.2. Objetivos a serem alcançados

1.2.1 Objetivo Geral:

Desenvolver um software para otimizar o processo de destinação de resíduos de forma sustentável, por meio de pontos de coleta seletiva.

1.2.2 Objetivos Específicos:

- Implementar um programa de bonificação para incentivar a participação dos cidadãos na destinação correta de resíduos.
- Estimular o descarte responsável de materiais, permitindo que os EcoCidadãos acumulem pontos por cada descarte realizado, os quais poderão ser resgatados posteriormente.
- Contribuir para a conscientização ambiental da população, promovendo práticas de descarte sustentável e a correta destinação de resíduos.
- Salientar toda a parte de qualidade de software associada ao desenvolvimento do software Ecotroca

2. PLANEJAMENTO DE TESTE DE SOFTWARE

Este plano de testes de Software foi elaborado para organizar, documentar e orientar todas as atividades de verificação e validação do sistema Ecotroca, assegurando que o produto atenda aos requisitos funcionais e não funcionais especificados durante o desenvolvimento.

O plano segue as boas práticas de engenharia de software, alinhado ao ciclo de vida ágil adotado pela equipe de desenvolvimento, e está estruturado para garantir cobertura adequada dos testes em todas as fases do projeto, desde os testes unitários até os testes de aceitação e regressão.

O Plano de Testes de Software tem como objetivo definir as diretrizes, estratégias, cronogramas, recursos e critérios necessários para garantir a qualidade e confiabilidade do sistema Ecotroca. Este documento visa:

- Validar os requisitos funcionais e não funcionais implementados.
- Garantir que o software atenda aos critérios de usabilidade, segurança, desempenho e compatibilidade.
- Planejar as atividades de testes em sintonia com o ciclo de desenvolvimento incremental e as Sprints do projeto.
- Minimizar riscos de falhas em produção e identificar erros nas fases iniciais.
- Promover rastreabilidade entre os casos de teste, requisitos e casos de uso definidos.

Este documento é direcionado a todos os atores envolvidos no ciclo de desenvolvimento e qualidade do sistema Ecotroca, incluindo:

- Time de Qualidade (QA): referência principal para execução e documentação de testes.
- Desenvolvedores Backend e Frontend: para suporte à integração e testes unitários.
- Product Owner e Scrum Master: para validação de entregas e alinhamento com os critérios de aceite.
- Equipe de DevOps: para configuração de ambientes e automação de testes.
- Stakeholders e clientes: para acompanhamento da qualidade e transparência no processo.

2.1. Cronograma de atividades

Sprint	Período Estimado	Atividades de Teste	Requisitos Envolvidos
Sprint 1	Semana 1 a 2	Planejamento de Testes, Definição de critérios de aceitação, Testes de autenticação (login e senha), Testes de recuperação de senha	RF01, RF02, USC001
Sprint 2	Semana 3 a 4	Testes de cadastro de EcoCidadão, Testes de listagem e visualização de pontos e ranking, Teste da funcionalidade de moedas por dia	RF03, RF08, RF09, USC002
Sprint 3	Semana 5 a 6	Testes de troca de moedas por prêmios, Emissão de extrato e histórico do cidadão	RF11, USC003, USC004

Sprint 4	Semana 7 a 8	Testes de cadastro de EcoDestinos, Testes de busca de EcoDestinos, Testes de gestão de resíduos	RF07, RF10, USC005, USC007
----------	--------------	---	----------------------------

2.2. Alocação de recursos

Recurso	Papel	Responsabilidades
QA Analyst 1	Planejamento/Testes Funcionais	Elaboração dos planos de teste, casos de uso, execução manual de testes
QA Analyst 2	Testes Automatizados/API	Automação de testes em endpoints REST (login, cadastro, relatórios)
Desenvolvedor 1	Backend	Correção de bugs, apoio nos testes de integração
Desenvolvedor 2	Frontend (Web/Mobile)	Suporte a testes UI e ajustes visuais baseados nos testes
Scrum Master	Facilitador	Acompanhamento do progresso de testes nas dailies
Product Owner	Validador	Revisão dos critérios de aceitação e testes de aceitação (UAT)
Usuário Representativo (Beta Tester)	Feedback	Realização de testes reais em ambiente de homologação

2.3. Marcos do projeto

Marco	Conclusão	Detalhamento
M1: Planejamento de testes aprovado	Final da Semana 1	Documento de planejamento de testes revisado e aprovado
M2: Ambiente de testes configurado	Final da Semana 2	Ambiente com dados de teste, usuários fictícios e mocks configurado
M3: Primeira rodada de testes concluída	Final da Semana 4	Relatório de bugs e resultados de testes de login, cadastro e pontuação
M4: Testes de funcionalidades completas	Final da Semana 10	Execução dos testes de funcionalidades principais e relatórios associados
M5: Testes de integração e segurança	Final da Semana 12	Resultados de testes de API, desempenho e segurança registrados
M6: Homologação	Semana 14	Versão estável aprovada pelo cliente/testes de aceite concluídos
M7: Go-live (produção)	Semana 15	Publicação da aplicação para usuários finais

3. DOCUMENTOS DE DESENVOLVIMENTO DE SOFTWARE

3.1 Plano de Projeto: ECOTROCA

Este documento detalha o planejamento, escopo, recursos e estimativas para o desenvolvimento da plataforma ECOTROCA, um software concebido para incentivar o descarte sustentável de resíduos por meio de um sistema de recompensas.

3.1.1. Planejamento do projeto

Este documento detalha o planejamento, escopo, recursos e estimativas para o desenvolvimento da plataforma ECOTROCA, um software concebido para incentivar o descarte sustentável de resíduos por meio de um sistema de recompensas.

O planejamento do projeto ECOTROCA se baseia em uma abordagem moderna e adaptativa, alinhada com as melhores práticas da engenharia de software, conforme descrito por Pressman e Maxim (2016) e Sommerville (2011).

Modelo de Processo: Será adotado o processo incremental. O software foi desenvolvido e entregue em partes funcionais (incrementos), permitindo avaliações contínuas com os stakeholders e a rápida entrega de valor. Esta abordagem minimiza riscos e permite que o produto evolua com base no feedback real.

Metodologia de Gestão: A gestão do projeto será conduzida utilizando o framework ágil Scrum. Esta escolha promove flexibilidade, transparência e colaboração. O trabalho será organizado em ciclos curtos e iterativos.

Sprints: Ciclos de desenvolvimento com duração fixa de 1 a 2 semanas.

Cerimônias do Scrum: Para garantir o alinhamento e a melhoria contínua do processo, serão realizadas as seguintes cerimônias:

- Reunião de Planejamento da Sprint (Sprint Planning): Definição do escopo a ser trabalhado em cada ciclo.
- Daily Scrum: Reunião diária de 15 minutos para sincronização da equipe.
- Revisão da Sprint (Sprint Review): Apresentação do incremento funcional desenvolvido no ciclo.
- Retrospectiva da Sprint: Análise do processo de trabalho para identificar pontos de melhoria.

3.1.2. Escopo

O escopo do projeto é o desenvolvimento de uma plataforma digital completa para incentivar a coleta seletiva, conectando cidadãos, empresas parceiras (EcoParceiros) e pontos de coleta (EcoDestinos).

Visão Geral do Produto: Uma aplicação web e mobile que automatiza o registro, a validação e a recompensa pelo descarte correto de materiais recicláveis, utilizando uma moeda virtual como mecanismo de engajamento.

Funcionalidades Principais:

- ❖ **Gestão de Usuários:** Cadastro e autenticação segura para múltiplos perfis (EcoCidadãos, Funcionários, Administradores).
- ❖ **Validação de Descarte:** Sistema de validação por meio da leitura de QR Codes gerados nas EcoBox após a análise do material.
- ❖ **Gamificação e Recompensas:** Acúmulo de moedas virtuais e XP (pontos de experiência) por descarte, com um ranking para incentivar a participação.
- ❖ **Marketplace de Trocas:** Uma loja virtual onde os usuários podem trocar suas moedas por cupons, produtos ou serviços dos EcoParceiros.
- ❖ **Gestão para EcoParceiros:** Ferramentas para gerenciar funcionários, cadastrar EcoDestinos e gerar relatórios sobre o volume de materiais recebidos.
- ❖ **Dashboards e Extratos:** Visualização de desempenho para todos os usuários, com histórico de atividades e extratos.

Inicialmente, estarão fora do escopo:

A logística física de coleta e transporte dos materiais;

O desenvolvimento do hardware das "EcoBox" (o sistema irá apenas se integrar a elas).

3.1.3. Recursos

Recursos Humanos: A equipe será estruturada com base nos papéis do Scrum:

- **Product Owner:** Responsável por maximizar o valor do produto e gerenciar o Product Backlog.
- **Scrum Master:** Facilitador do processo, responsável por remover impedimentos.
- **Time de Desenvolvimento:** Equipe multidisciplinar responsável pela entrega do incremento.

- **Membros:** Lucas, Wellerson, Paloma, João R., Gabriel, Lucas T., Amanda P., Fernanda M.

Recursos Tecnológicos:

- **Gestão de Projeto e Comunicação:** Trello, Discord, GitHub.
- **Design e Prototipagem:** Figma.
- **Arquitetura de Software:** Padrão MVC (Model-View-Controller), complementado por uma arquitetura em camadas (Service e Repository). Serão utilizados também padrões de design como DTO, Singleton e Strategy para garantir um código limpo, modular e escalável.
- **Monitoramento e DevOps:** Para garantir a confiabilidade e o desempenho do sistema, será implementada uma stack de observabilidade com ferramentas como OpenTelemetry, Prometheus, Jaeger e Grafana.

3.1.4. Estimativas de projeto

As estimativas são baseadas no Roadmap inicial e serão refinadas a cada Sprint. O foco é entregar valor de forma contínua.

- **Sprint 1-2: Fundação e Engajamento Inicial**
 - ❖ Prototipagem do MVP no Figma.
 - ❖ Desenvolvimento do fluxo de cadastro e autenticação segura do EcoCidadão.
 - ❖ Início do desenvolvimento da validação de descarte por QR Code.
- **Sprint 2-4: Ecossistema de Recompensas e Parceiros**
 - ❖ Conclusão da validação de descarte e geração de moedas.
 - ❖ Desenvolvimento do sistema de troca de recompensas.
 - ❖ Implementação do cadastro de EcoDestinos e gestão de materiais pelos EcoParceiros.
- **Sprint 3-4: Inteligência e Visualização de Dados**

- ❖ Criação dos dashboards de desempenho para os usuários.
- ❖ Desenvolvimento dos relatórios de descarte para os EcoParceiros.
- ❖ Refinamento do sistema de ranking e gamificação.

3.2 Documentos de requisitos

3.2.1 Requisitos funcionais

ID	Descrição
RF 01	O sistema deverá solicitar ao EcoCidadão seu login e senha e verificar se o mesmo possui permissão de acesso ao sistema
RF 02	O EcoCidadão que esqueceu a senha poderá solicitá-la ao sistema, bastando informar seu login. Neste caso o sistema irá enviar um e-mail com o link de confirmação de esqueci minha senha. Caso o EcoCidadão clique no link, o sistema enviará outra senha aleatória, gerada pelo mesmo. No primeiro login do EcoCidadão o sistema deverá solicitá-lo que cadastre uma nova senha.
RF 03	O sistema deverá permitir que os EcoCidadãos se cadastrem obrigatoriamente com nome, e-mail, senha e cpf
RF 04	A EcoParceiro parceira poderá cadastrar, excluir ou alterar dados dos funcionários no sistema para que eles possam atuar
RF 05	A EcoParceiro poderá gerenciar relatórios de quantos materiais foram recebidos nos Ecopontos da EcoParceiro, realizando filtros de data de recebimento, data descarte e podendo extrair os dados em CSV, PDF, XLSX
RF 06	A EcoParceiro poderá ver as pontuações do EcoCidadão com o objetivo de fornecer benefícios/produtos em troca dos pontos alcançados pelo Cidadão
RF 07	O EcoParceiro poderá cadastrar novos EcoDestinos no sistema, cadastrando endereço, horário de funcionamento, resíduos recebidos e telefone do local.

RF 08	O EcoCidadão ganhará XP em cada descarte diário e participará de um ranking de XPs como outros EcoCidadãos.
RF 09	O EcoCidadão receberá uma moeda por dia mediante descarte diário de resíduos aceitos no Ecotroca
RF 10	As EcoBox irão validar o recebimento do resíduo mediante apresentação pelo EcoCidadão nas câmeras inteligentes e inserção na EcoBox, após a validação com sucesso, aparecerá o QR CODE para o EcoCidadão escanear confirmando o descarte.
RF 11	O EcoCidadão poderá trocar as moedas adquiridas por cupons de desconto, itens ou serviços de parceiros, desde que a quantidade necessária de moedas seja respeitada.
RF 12	O Ecotroca terá todos os RFs do EcoParceiros e dos EcoCidadãos, além de poder gerenciar os EcoCidadãos cadastrados no sistema.

3.2.2 Requisitos não funcionais

ID	Descrição	Categoria ISO/IEC 25010
RNF 01	O sistema deverá estar disponível 24 horas por dia, 7 dias por semana.	Disponibilidade
RNF 02	O tempo de resposta do sistema não deverá ultrapassar 2 segundos para operações comuns.	Desempenho
RNF 03	A plataforma deverá ser responsiva, funcionando em dispositivos móveis, tablets e desktops.	Usabilidade
RNF 04	O sistema deverá armazenar informações sensíveis (como senhas e CPF) com criptografia.	Segurança
RNF 05	Toda comunicação entre cliente e servidor deverá ocorrer por meio do protocolo HTTPS.	Segurança

RNF 06	O sistema deverá suportar pelo menos 10.000 acessos simultâneos sem perda de desempenho.	Escalabilidade
RNF 07	O sistema deverá manter um tempo médio de recuperação de falhas (MTTR) inferior a 1 hora.	Confiabilidade
RNF 08	O sistema deverá oferecer backup automático diário dos dados.	Manutenibilidade
RNF 09	A interface deverá seguir boas práticas de acessibilidade (contraste, navegação por teclado, etc).	Usabilidade / Acessibilidade
RNF 10	O sistema deverá estar em conformidade com a LGPD (Lei Geral de Proteção de Dados).	Conformidade Legal

3.2.3 Regras de negócio

ID	Nome da Regra	Descrição
RN 01	Limitação de Pontuação Diária	Cada descarte válido realizado por um EcoCidadão gera 1 moeda por dia. Não é possível acumular mais de uma moeda por dia.
RN 02	Validação via QR Code	Para validar o descarte, o usuário deve escanear o QR Code gerado após a aprovação da câmera inteligente da EcoBox.
RN 03	Troca Condicionada ao Saldo	Os cupons, itens ou serviços disponíveis para troca só podem ser resgatados caso o EcoCidadão tenha saldo suficiente.
RN 04	Cadastramento Restrito de EcoPontos	Apenas empresas com CNPJ ativo podem se cadastrar como EcoPontos.
RN 05	Controle de Acesso para Funcionários	Funcionários só podem atuar no sistema se forem previamente cadastrados e associados a uma empresa parceira.

RN 06	Recuperação Segura de Senha	Um usuário só pode alterar sua senha mediante confirmação por link enviado ao e-mail cadastrado.
RN 07	Atualização do Ranking	O ranking de EcoCidadãos será atualizado semanalmente com base na quantidade de moedas acumuladas.
RN 08	Prevenção de Fraudes	Não é permitido resgatar recompensas com moedas obtidas sem validação pelo QR Code.
RN 09	Imutabilidade de Relatórios	O sistema não permitirá alterações retroativas nos relatórios de recebimento de materiais pelas empresas.
RN 10	Retenção de Dados para Auditoria	Os dados de pontuação e descarte dos usuários serão armazenados por pelo menos 5 anos para fins de auditoria.

3.2.4 Casos de uso



Código	Caso de Uso	Atores	Descrição
USC001	Acessar o sistema	EcoParceiro, Cidadão	Permite o login/autenticação no sistema para acesso às funcionalidades.
USC002	Visualizar pontuação	Cidadão	Cidadão pode consultar a pontuação atual acumulada.
USC003	Resgatar prêmios	Cidadão	Permite que o cidadão troque seus pontos por prêmios disponíveis.
USC004	Emitir extrato	Cidadão	Gera um extrato detalhado com todas as ações, pontuações e trocas feitas.
USC005	Buscar EcoDestinos	Cidadão	Lista ecopontos (EcoDestinos) disponíveis próximos ao cidadão.
USC006	Gerenciar premiações	EcoParceiro	Permite cadastro, edição ou exclusão de prêmios disponíveis para troca.
USC007	Gerenciar resíduos	EcoParceiro	Permite registrar os resíduos entregues pelos cidadãos.
USC008	Gerenciar pontuações	EcoParceiro	Permite adicionar ou ajustar pontos dos cidadãos com base nos resíduos.
USC009	Gerar relatório	EcoParceiro	Gera relatórios administrativos sobre pontuação, resíduos e prêmios.
USC010	Gerenciar funcionários	EcoParceiro	Permite o cadastro, edição e exclusão de funcionários da empresa parceira.
USC011	Cadastrar EcoDestinos	EcoParceiro	Permite cadastrar novos ecopontos no sistema.
USC012	Gerenciar usuários	ECOTROCA	Caso de uso administrativo que gerencia todos os perfis de usuário.

3.3 Planejamento de testes

O planejamento de testes define a estratégia de validação da qualidade do software Ecotroca, descrevendo objetivos, escopo, tipos de testes, requisitos a serem verificados, ferramentas utilizadas, recursos necessários, cronograma e marcos. Este planejamento garante que os requisitos (funcionais e não funcionais) sejam devidamente atendidos antes da entrega final.

3.3.1 Plano de Testes

3.3.1.1 Introdução

Este Plano de Testes de Software foi elaborado para organizar, documentar e orientar todas as atividades de verificação e validação do sistema Ecotroca, assegurando que o produto atenda aos requisitos funcionais e não funcionais especificados durante o desenvolvimento. O plano segue práticas ágeis e cobre testes unitários, de integração, sistema, regressão e aceitação.

3.3.1.2 Escopo

Serão testadas as funcionalidades mais críticas da plataforma, abrangendo autenticação, cadastro, pontuação, descarte, relatórios, troca de moedas e integração com EcoDestinos. Funcionalidades previstas para versões futuras (ex.: analytics) estão fora do escopo deste ciclo.

3.3.1.3 Objetivos

- Validar os requisitos funcionais e não funcionais implementados.
- Assegurar conformidade com critérios de usabilidade, segurança, desempenho e compatibilidade.
- Identificar e mitigar falhas em fases iniciais do desenvolvimento.
- Promover rastreabilidade entre casos de teste, requisitos e casos de uso.

3.3.1.4 Requisitos a serem testados

Todos os requisitos funcionais RF01 a RF12 e não funcionais RNF01 a RNF10, além das regras de negócio RN01 a RN10, com foco em: Autenticação e autorização; Validação de descartes com XP e moedas; Integração com EcoBox e QR Code; Troca de recompensas; Relatórios e dashboards; Conformidade com a LGPD

3.3.1.5 Estratégias, tipos de testes e ferramentas

Tipo de Teste	Objetivo	Ferramentas
Testes de Unidade	Validar métodos e serviços isoladamente	JUnit, Mockito
Testes de Integração	Verificar comunicação entre APIs e banco de dados	Postman, SpringBoot Test
Testes Funcionais	Validar funcionalidades do ponto de vista do usuário final	Selenium, Cypress
Testes de Carga	Avaliar desempenho sob múltiplos acessos simultâneos	Apache JMeter
Testes de Usabilidade	Avaliar a experiência do usuário (UX)	Sessões com usuários
Testes de Segurança	Verificar criptografia e proteção contra ataques	OWASP ZAP

3.3.1.6 Recursos a serem empregados

- Ambiente de homologação para teste
- Equipe de QA (1 analista), backend (1), frontend (1)
- Apoio de Product Owner e Scrum Master
- Dados de teste com usuários fictícios e simulações de descarte

3.3.1.7 Cronograma das Atividades

Conforme cronograma geral (ver seção 2.1), os testes serão realizados ao longo das sprints, começando pela autenticação (Sprint 1), cadastro e pontuação (Sprint 2), funcionalidades de recompensa (Sprint 3), e gestão de destinos e resíduos (Sprint 4).

3.3.1.8 Definição dos Marcos do Projeto

Marco	Conclusão	Detalhamento
M1: Planejamento de testes aprovado	Final da Semana 1	Documento de planejamento de testes revisado e aprovado

M2: Ambiente de testes configurado	Final da Semana 2	Ambiente com dados de teste, usuários fictícios e mocks configurado
M3: Primeira rodada de testes concluída	Final da Semana 4	Relatório de bugs e resultados de testes de login, cadastro e pontuação
M4: Testes de funcionalidades completas	Final da Semana 10	Execução dos testes de funcionalidades principais e relatórios associados
M5: Testes de integração e segurança	Final da Semana 12	Resultados de testes de API, desempenho e segurança registrados
M6: Homologação	Semana 14	Versão estável aprovada pelo cliente/testes de aceite concluídos
M7: Go-live (produção)	Semana 15	Publicação da aplicação para usuários finais

3.3.2 Casos de Testes

Código	Requisito	Cenário	Condição	Resultado Esperado
CT01	RF01	Login com credenciais válidas	login: eco@email.com / senha:123	Acesso autorizado
CT02	RF01	Login com credenciais inválidas	login: eco@email.com / senha:errada	Mensagem de erro "Credenciais inválidas"
CT03	RF10	Validação de descarte com EcoBox	imagem e QR code válidos	Moeda adicionada ao perfil
CT04	RF05	Geração de relatório com filtro de data	datas entre 01/06 e 30/06	Relatório exportado em CSV
CT05	RNF02	Performance de login com 100 usuários	100 threads simultâneas	Tempo de resposta < 2 segundos

CT06	RNF10	Exclusão segura de conta e anonimização	Conta real e confirmação via e-mail	Dados anonimizados e removidos do perfil
------	-------	---	-------------------------------------	--

3.3.3 Roteiro de Testes

Roteiro CT03 – Validação de Descarte com EcoBox

1. Iniciar câmera da EcoBox em ambiente de homologação
2. Simular descarte de item permitido
3. Verificar reconhecimento da imagem pelo sistema
4. Confirmar geração de QR Code no painel
5. Escanear QR Code com app do EcoCidadão
6. Verificar se a moeda foi creditada corretamente no perfil

Roteiro CT05 – Teste de Performance (Login)

1. Preparar script no Apache JMeter com endpoint POST /login
2. Simular 100 usuários acessando simultaneamente
3. Monitorar tempo médio e máximo de resposta
4. Validar se 95% das requisições obtêm resposta < 2s

Roteiro CT06 – Teste de Exclusão Segura (LGPD)

1. Logar como EcoCidadão com perfil de teste
2. Acessar configurações de conta e selecionar "Excluir conta"
3. Confirmar via link de e-mail
4. Verificar anonimização dos dados no banco
5. Validar remoção dos dados sensíveis e históricos

4. GESTÃO E CONFIGURAÇÃO DE SOFTWARE

A Gestão de Configuração de Software (GCS) é uma prática essencial para garantir o controle, rastreabilidade e integridade de um projeto de software. Neste

trabalho, será apresentado um exemplo prático da aplicação da GCS em um sistema fictício de reciclagem de materiais

4.1. Aplicação dos Conceitos de GCS.

4.1.1 Identificação de Itens de Configuração (ICs)

Os principais ICs definidos foram:

- Código-fonte do sistema (/src)
- Scripts de banco de dados

(/Prometheus/OpenSearch/Thanos/Jaeger/Grafana/ OpenTelemetry Collector)

- Documentação técnica (/docs)
- Arquivos de configuração (/Trello,Github)

4.1.2 Controle de Versão

Foi utilizado o Github para versionamento. Cada nova funcionalidade foi desenvolvida em uma branch separada.

4.1.3 Controle de Mudanças

As mudanças foram controladas por meio de pull requests no GitHub, com revisão por outro membro da equipe antes da integração à branch principal (main).

4.1.4 Auditoria e Status de Configuração

Foi mantido um arquivo CHANGELOG.md com o histórico de versões e mudanças. Exemplo:

```
1 # Changelog
2
3 Todas as mudanças notáveis neste projeto serão documentadas neste arquivo.
4
5 O formato é baseado em [Keep a Changelog
6 e este projeto segue o versionamento semântico.
7
8 ## [1.2.0] - 2025-06-10
9 ### Adicionado
10 - Relatório de materiais reciclados por tipo.
11 - Exportação de dados em CSV.
12
13 ### Alterado
14 - Interface gráfica atualizada com novos ícones.
15
16 ## [1.1.0] - 2025-05-20
17 ### Adicionado
18 - Cadastro de usuários com autenticação simples.
19 - Tela de login com validação.
20
21 ## [1.0.0] - 2025-05-01
22 ### Adicionado
23 - Cadastro de materiais recicláveis.
24 - Registro de quantidade reciclada.
25 - Interface em linha de comando.
26
```

4.1.5 Reprodutibilidade

Para garantir a reprodutibilidade do ambiente, foi utilizado

- OpenTelemetry - coleta métricas e logs.
- Prometheus - armazena as métricas.
- Thanos - garante que essas métricas não sejam perdidas com o tempo.
- Jaeger - mostra o caminho das requisições.
- OpenSearch - guarda os logs detalhados.
- Grafana - exibe tudo isso em dashboards que podem ser salvos e compartilhados.

4.2 Estrutura do projeto

```
reciclagem-pontos/  
|  
├─ src/  
|   └─ br/  
|       └─ com/  
|           └─ reciclagem/  
|               └─ Main.java  
|               └─ model/  
|                   └─ Usuario.java  
|                   └─ Material.java  
|                   └─ RegistroReciclagem.java  
|               └─ service/  
|                   └─ SistemaReciclagem.java  
|                   └─ PontuacaoService.java  
|               └─ util/  
|                   └─ Relatorio.java  
|  
├─ resources/  
|   └─ application.properties  
|  
├─ .gitignore  
├─ README.md  
└─ CHANGELOG.md
```

5. DOCUMENTOS DE DESENVOLVIMENTO DE SOFTWARE

O Repositório de Gestão de Configuração de Software é um ambiente centralizado (físico ou lógico) onde são armazenados todos os artefatos de software (código-fonte, documentação, bibliotecas, scripts, dados de teste, etc.), sob controle de versão e com histórico completo de modificações, garantindo integridade, rastreabilidade e consistência do projeto ao longo do tempo.

Ferramentas Utilizadas:

- Git (controle de versão)
- GitHub (repositório remoto)
- Visual Studio Code (editor de código)
- Terminal / Git Bash

Estrutura de Branches:

- main → código em produção
- develop → integração contínua
- feature/* → desenvolvimento de novas funcionalidades
- hotfix/* → correções emergenciais

5.1. Organização em Pacotes (Packages)

- Separação clara entre model, service, util e main.
- Facilita a manutenção, leitura e escalabilidade do código.

5.2. Separação de Responsabilidades (SRP - Single Responsibility Principle)

Cada classe tem uma única responsabilidade:

- Usuario, Material, RegistroReciclagem → apenas dados.
- PontuacaoService → lógica de pontuação.
- SistemaReciclagem → orquestra o funcionamento do sistema.
- Relatorio → geração de relatórios.

5.3. README.md

- Documentar o propósito do projeto, como rodar, e como contribuir.
- Essencial para colaboração e entendimento rápido do sistema.

5.4. CHANGELOG.md

- Histórico claro de versões e mudanças.
- Facilita rastrear o que foi alterado e quando.

5.5. Gitignore

- Evita que arquivos desnecessários (como bin/, .class, etc.) sejam versionados.
- Mantém o repositório limpo e profissional.

5.6. Versionamento com Git

- Controle de versões com commits claros.
- Possibilidade de colaboração e rastreamento de alterações.

5.7. Código Modular e Reutilizável

- Serviços e modelos podem ser reutilizados em outras partes do sistema ou até em outros projetos.

5.8. Preparação para Expansão

- Estrutura pronta para integração com banco de dados, interface gráfica ou APIs REST.
- Possível integração futura com ferramentas como Prometheus, Grafana, etc.

5.9. Nomenclatura Clara e Consistente

- Classes, métodos e pacotes com nomes descritivos e em conformidade com convenções Java.

5.10. Separação de Configurações

- Arquivo `application.properties` reservado para configurações futuras (como caminhos, banco de dados, etc.).

5.11 Políticas de Merge

- Pull requests revisados por, pelo menos, 2 membros da equipe.
- Continuous Integration disparado automaticamente para executar testes de unidade e integração antes do merge.

5.12 Tags e Releases

- Versão gerenciada por tags semânticas (v1.0.0, v1.1.0, ...).
- Releases criadas no GitHub contendo changelog e artefatos (pacotes, scripts).

REFERÊNCIA BIBLIOGRÁFICA

PRESSMAN, Roger; MAXIM, Bruce. Engenharia de Software. Uma abordagem profissional. 9a. Ed. Bookman, 2021. Parte III – Gestão da qualidade; Capítulo 19 – Conceitos de qualidade, pg. 412

SOMMERVILLE, Ian. Engenharia de Software. 9. ed. São Paulo: Pearson Prentice Hall, 2011.

GONÇALVES, Priscila de Fátima.; BARRETO, Jeanine dos Santos.; ZENKER, Aline Maciel.; FAGUNDES, Rubem. Testes de software e gerência de configuração. Soluções Educacionais Integradas, 2019.

PRESSMAN, Roger S.; MAXIM, Bruce R. Engenharia de Software: uma abordagem profissional. 8. ed. Porto Alegre: AMGH, 2016.

SOMMERVILLE, Ian. Engenharia de Software. 9. ed. São Paulo: Pearson Prentice Hall, 2011.

GONÇALVES, Priscila de Fátima; BARRETO, Jeanine dos Santos; ZENKER, Aline Maciel; FAGUNDES, Rubem. Testes de software e gerência de configuração. Soluções Educacionais Integradas, 2019.

PRESSMAN, Roger; MAXIM, Bruce. *Engenharia de Software: Uma abordagem profissional*. 9. ed. Porto Alegre: Bookman, 2021. Parte III – Gestão da Qualidade; Capítulo 19 – Conceitos de qualidade, p. 412-435.

SOMMERVILLE, Ian. *Engenharia de Software*. 9. ed. São Paulo: Pearson Prentice Hall, 2011. Capítulo 8 – Validação de Software e Capítulo 24 – Garantia de Qualidade de Software.

GONÇALVES, Priscila de Fátima; BARRETO, Jeanine dos Santos; ZENKER, Aline Maciel; FAGUNDES, Rubem. *Testes de software e gerência de configuração*. Soluções Educacionais Integradas, 2019. Capítulos 2 e 3.

Observação: Conforme os princípios ágeis defendidos por Pressman e Maxim (2016), estas são estimativas iniciais. O escopo pode ser ajustado ao final de cada Sprint para melhor atender às necessidades do mercado e dos usuários, garantindo que o produto final seja relevante e de alta qualidade.