
IDENTIFICAÇÃO DO PROJETO

Projeto

- Recicla Run

Responsáveis pelo Projeto

- Pedro Makson Fontes da Costa
- Arthur Kellyson Pinheiro de Negreiros

PLANO DE ESPECIFICAÇÃO FORMAL

1. Introdução

Este documento apresenta o plano de especificação formal do software *Recicla Run*, um jogo educativo baseado em fases, cujo objetivo é estimular práticas sustentáveis por meio da coleta e classificação de lixo em cenários interativos. O jogador controla um personagem que deve coletar resíduos, descartá-los corretamente e avançar progressivamente até a tela de finalização.

Para garantir clareza, consistência e ausência de ambiguidades na definição dos requisitos, será utilizada a notação **Redes de Petri Ordinárias**, que permite representar formalmente estados, transições e condições do sistema de forma gráfica e matemática.

2. Objetivos da Especificação

A especificação formal tem como objetivo:

- Representar de maneira rigorosa os requisitos funcionais (**RF01–RF10**) e não funcionais (**RNF01–RNF04**).
- Evitar ambiguidades e contradições na definição das regras do jogo.
- Apoiar a verificação e validação do sistema por meio de modelos formais.
- Fornecer uma base para a criação de **casos de teste** e para futuras evoluções do projeto, garantindo preservação de restrições e invariantes.

3. Escopo

O software consiste em um jogo baseado em cenários sucessivos, onde o jogador deve:

1. Iniciar o jogo e movimentar o personagem.
2. Coletar resíduos espalhados pelo cenário.
3. Desviar de obstáculos (colisões implica reinício da fase).
4. Classificar corretamente os resíduos em lixeiras específicas.
5. Avançar para novos cenários e visualizar a evolução (contador de lixo, transformação do ambiente).
6. Ao final, exibir uma tela de conclusão com caráter educativo.

A formalização por Redes de Petri dará ênfase ao fluxo principal do jogo, às condições de coleta, descarte e reinício de fases, e ao controle de progressão entre cenários.

4. Notação Utilizada

Será utilizada a **Rede de Petri Ordinária (RPO)**, composta por:

- **Lugares (círculos):** representam estados do sistema, como "Lixo disponível", "Personagem em movimento", "Inventário cheio".
- **Transições (retângulos):** representam eventos/ações, como "Coletar lixo", "Classificar lixo", "Reiniciar cenário".
- **Arcos direcionados:** conectam lugares e transições, estabelecendo a dinâmica.
- **Fichas (tokens):** representam a marcação dos lugares, indicando o estado atual do jogo (ex.: número de itens coletados).

Essa abordagem é adequada pois modela a concorrência, paralelismo e condições, aspectos presentes no fluxo do jogo.

5. Estrutura da Especificação Formal

A especificação será estruturada em:

- **Definição de conjuntos básicos:** Jogador, Lixos, Lixeiras, Obstáculos, Cenários.
- **Definição de lugares principais:** Jogo iniciado, Personagem em movimento, Lixo no inventário, Obstáculo encontrado, Lixeira correta/incorreta, Cenário concluído, Jogo finalizado.
- **Definição de transições principais:** Iniciar jogo, Movimentar personagem, Coletar lixo, Desviar obstáculo, Classificar lixo, Avançar cenário, Reiniciar fase, Finalizar jogo.
- **Invariantes estruturais:**
 - Um lixo só pode ser classificado em lixeira correta.
 - O avanço de cenário depende da coleta e classificação de todos os resíduos.
 - Em colisão com obstáculo, a fase deve reiniciar.

6. Operações

As principais operações formalizadas serão:

- **Iniciar jogo:** liberar tokens iniciais para o estado *Personagem em movimento*.
- **Coletar lixo:** transferir token de *Lixo disponível* para *Inventário*.
- **Classificar lixo:** mover token de *Inventário* para *Lixeira correta*; se incorreta, retornar token ao inventário.
- **Desviar obstáculo:** se o jogador não desviar, token vai para *Colisão*, disparando reinício.
- **Avançar cenário:** apenas habilitada se todos os tokens de lixo estiverem em *Lixeira correta*.
- **Exibir finalização:** ocorre quando o último cenário é concluído.

7. Requisitos a serem especificados

Requisitos Funcionais (RF)

- RF01: Iniciar jogo.
- RF02: Movimentar personagem.
- RF03: Coletar lixo.
- RF04: Desviar obstáculos.
- RF05: Classificar lixo nas lixeiras.
- RF06: Avançar para o próximo cenário.
- RF07: Exibir contador de lixo coletado.
- RF08: Reiniciar cenário em caso de colisão.
- RF09: Exibir e transformar cenários.
- RF10: Exibir tela de finalização.

Requisitos Não-Funcionais (RNF)

- RNF01: O sistema deve ser intuitivo e de fácil interação.
- RNF02: O sistema deve ser modular e adaptável a novas fases.
- RNF03: O sistema deve garantir baixo consumo de recursos, mantendo desempenho estável.
- RNF04: Mensagens educativas relacionadas aos ODS

8. Metodologia

A metodologia seguirá as etapas:

1. Identificação dos requisitos críticos.
2. Modelagem das condições e fluxos principais com Redes de Petri.
3. Definição das marcações iniciais (estado inicial do jogo).
4. Especificação dos invariantes e restrições de consistência.
5. Validação do modelo por meio de simulações de marcações.
6. Derivação de casos de teste a partir das condições formais.

9. Critérios de Aceitação

A especificação formal será considerada concluída quando:

- Todos os requisitos funcionais (RF01–RF10) estiverem mapeados na Rede de Petri.
- Os requisitos não funcionais (RNF01–RNF03) forem considerados como restrições adicionais de projeto.
- O modelo não apresentar contradições e permitir simulação consistente.
- Servir como base para os testes de software, garantindo correspondência entre comportamento esperado e implementação.