

## MODELO PARA O DOCUMENTO DE REQUISITOS DO PROJETO

### REQUISITOS FUNCIONAIS

[RF001]		
Nome:	Iniciar Jogo	
Descrição:	O sistema deve permitir que o jogador inicie uma nova partida, carregando o cenário inicial, exibindo o personagem em sua posição inicial e preparando os elementos de jogo (obstáculos, lixos e lixeiras).	
Atores:	Jogador	
Prioridade:	Alta	
Entradas e pré-condições:	<ul style="list-style-type: none"><li>● O jogo deve estar aberto.</li><li>● O jogador deve selecionar a opção de iniciar partida.</li></ul>	
Saídas e pós-condições:	<ul style="list-style-type: none"><li>● O jogo é iniciado no cenário.</li><li>● O personagem aparece ativo e pronto para receber comandos de movimento.</li><li>● Os elementos do cenário (obstáculos, lixos e lixeiras) são carregados.</li></ul>	
Fluxos de eventos		
Fluxo principal:	<ol style="list-style-type: none"><li>1. O jogador seleciona a opção “Iniciar Jogo”.</li><li>2. O sistema carrega o cenário inicial.</li><li>3. O sistema posiciona o personagem no ponto inicial do cenário.</li><li>4. O sistema disponibiliza os elementos do cenário (obstáculos, lixos e lixeiras).</li><li>5. O jogo inicia, permitindo que o jogador controle o personagem.</li></ol>	
Fluxo secundário 1:	-	

[RF002]		
Nome:	Movimentar o personagem	
Descrição:	O sistema deve permitir que o jogador controle o personagem em diferentes formas de movimento, de acordo com o cenário. Nos cenários da cidade e da floresta, o personagem pode realizar movimentos de pulo. Já no cenário aquático, o personagem pode se mover em todas as direções (direita, esquerda, cima e baixo).	
Atores:	Jogador	
Prioridade:	Alta	
Entradas e pré-condições:		<ul style="list-style-type: none"><li>• O jogo deve estar em execução.</li><li>• O personagem deve estar ativo em um cenário.</li></ul>
Saídas e pós-condições:		<ul style="list-style-type: none"><li>• O personagem se move conforme os comandos permitidos para o cenário em que se encontra.</li></ul>
Fluxos de eventos		
Fluxo principal:		<div>6. O jogador pressiona uma tecla de movimento (direita, esquerda, acima, abaixo e pulo).</div> <div>7. O sistema identifica a entrada.</div> <div>8. O personagem se movimenta de acordo com o cenário.</div>
Fluxo secundário 1:		<div>1. Se a tecla pressionada não for válida, nenhuma ação é realizada e o jogo permanece em andamento.</div>

[RF003]		
Nome:	Coletar lixo	
Descrição:	O sistema deve permitir que o jogador colete objetos de lixo espalhados pelos cenários.	
Atores:	Jogador	
Prioridade:	Essencial	
Entradas e pré-condições:	<ul style="list-style-type: none"><li>• O personagem deve se movimentar até a posição do objeto de lixo.</li></ul>	
Saídas e pós-condições:	<ul style="list-style-type: none"><li>• O lixo é removido do cenário e adicionado ao inventário do jogador.</li></ul>	
Fluxos de eventos		
Fluxo principal:	<ol style="list-style-type: none"><li>1. O jogador se aproxima do item.</li><li>2. O sistema detecta colisão com o objeto.</li><li>3. O item é coletado e removido do cenário.</li></ol>	
Fluxo secundário 1:	-	

[RF004]		
Nome:	Exibir contador de lixo	
Descrição:	O sistema deve exibir, em tempo real, um contador na interface do jogo que apresente a quantidade de lixo coletado pelo jogador, organizado por tipo (ex.: plástico, metal, vidro, papel, orgânico, etc.). O contador deve ser atualizado automaticamente conforme o jogador coleta novos itens.	
Atores:	Jogador	
Prioridade:	Essencial	
Entradas e pré-condições:	<ul style="list-style-type: none"><li>• O jogador deve estar em uma partida ativa.</li><li>• O jogo deve ter itens de lixo disponíveis no cenário para coleta.</li></ul>	
Saídas e pós-condições:	<ul style="list-style-type: none"><li>• O contador permanece visível e atualizado durante toda a partida.</li><li>• Os valores de lixo coletado são armazenados até o fim da rodada.</li></ul>	
Fluxos de eventos		
Fluxo principal:	<ol style="list-style-type: none"><li>1. O jogador inicia a partida.</li><li>2. O jogo posiciona diferentes tipos de lixo no cenário.</li><li>3. O jogador coleta um item de lixo.</li><li>4. O sistema identifica o tipo do item coletado.</li><li>5. O contador em tela é atualizado, incrementando na categoria correspondente ao tipo de lixo</li><li>6. O jogador pode visualizar em tempo real a quantidade total e por tipo de lixo coletado.</li></ol>	
Fluxo secundário 1:	-	

[RF005]		
Nome:	Desviar obstáculos	
Descrição:	O sistema deve permitir que o jogador evite obstáculos no cenário (ex.: carros, árvores caídas, pedras).	
Atores:	Jogador	
Prioridade:	Média	
Entradas e pré-condições:	<ul style="list-style-type: none"><li>• O obstáculo deve estar posicionado no cenário.</li></ul>	
Saídas e pós-condições:	<ul style="list-style-type: none"><li>• O jogador passa pelo obstáculo sem colisão.</li><li>• Caso colida, a fase reinicia.</li></ul>	
Fluxos de eventos		
Fluxo principal:	<ol style="list-style-type: none"><li>1. O jogador se move em direção ao obstáculo.</li><li>2. O jogador desvia a tempo.</li><li>3. O jogo continua normalmente.</li></ol>	
Fluxo secundário 1:	<ol style="list-style-type: none"><li>1. O jogador não desvia.</li><li>2. O sistema aplica penalidade (a fase reinicia).</li></ol>	

[RF006]		
Nome:	Reiniciar cenário	
Descrição:	O sistema deve reiniciar automaticamente o cenário do jogo sempre que o jogador colidir com um obstáculo ou elemento que represente falha. Dessa forma, a partida é reiniciada sem necessidade de intervenção manual, garantindo a continuidade do jogo.	
Atores:	Jogador	
Prioridade:	Média	
Entradas e pré-condições:	<ul style="list-style-type: none"><li>• O jogador deve estar em uma partida ativa.</li><li>• O cenário deve conter obstáculos ou elementos que possam causar colisão.</li></ul>	
Saídas e pós-condições:	<ul style="list-style-type: none"><li>• O cenário é reiniciado imediatamente após a colisão.</li><li>• O contador de lixo coletado e a pontuação retornam ao valor inicial.</li></ul>	
Fluxos de eventos		
Fluxo principal:	<ol style="list-style-type: none"><li>1. O jogador inicia a partida.</li><li>2. O jogador percorre o cenário e coleta itens de lixo.</li><li>3. O jogador colide com um obstáculo.</li><li>4. O sistema detecta a colisão.</li><li>5. O sistema reinicia automaticamente o cenário, reposicionando o jogador e os elementos do jogo em seu estado inicial.</li></ol>	
Fluxo secundário 1:	<ol style="list-style-type: none"><li>1. O jogador não desvia do obstáculo.</li><li>2. O jogo continua normalmente.</li></ol>	

[RF007]		
Nome:	Classificar lixo nas lixeiras	
Descrição:	O sistema deve permitir que o jogador descarte o lixo coletado em lixeiras específicas (plástico, papel, vidro, metal, orgânico) no final de cada fase.	
Atores:	Jogador	
Prioridade:	Alta	
Entradas e pré-condições:	<ul style="list-style-type: none"><li>• O jogador deve ter lixo coletado no inventário.</li><li>• Lixeira deve estar disponível no cenário.</li></ul>	
Saídas e pós-condições:	<ul style="list-style-type: none"><li>• Lixo é removido do inventário e contabilizado na lixeira correta.</li></ul>	
Fluxos de eventos		
Fluxo principal:	<ol style="list-style-type: none"><li>1. O jogador interage com uma lixeira.</li><li>2. O sistema verifica o tipo de lixo.</li><li>3. O lixo é corretamente classificado.</li><li>4. O cenário é atualizado (melhora visual).</li></ol>	
Fluxo secundário 1:	<ol style="list-style-type: none"><li>1. O jogador interage com uma lixeira incorreta.</li><li>2. O sistema detecta a inconsistência.</li><li>3. O sistema exibe uma mensagem de erro educativa.</li><li>4. O lixo permanece no inventário do jogador.</li></ol>	

[RF008]		
Nome:	Exibir e transformar cenários	
Descrição:	O sistema deve exibir três cenários (cidade, floresta e oceano), que são transformados após a coleta e correta classificação do lixo.	
Atores:	Jogador	
Prioridade:	Essencial	
Entradas e pré-condições:	<ul style="list-style-type: none"><li>O jogador deve completar a coleta e classificação de todos os lixos do cenário atual.</li></ul>	
Saídas e pós-condições:	<ul style="list-style-type: none"><li>O cenário é transformado em sua versão limpa (ex.: cidade sem poluição, floresta com árvores saudáveis, oceano limpo).</li></ul>	
Fluxos de eventos		
Fluxo principal:	<ol style="list-style-type: none"><li>O jogador completa a classificação correta de todos os lixos.</li><li>O sistema transforma o cenário exibindo sua versão limpa.</li></ol>	
Fluxo secundário 1:	<ol style="list-style-type: none"><li>Se houver lixo não classificado, o cenário não é transformado.</li></ol>	



[RF009]		
Nome:	Avançar para o próximo cenário	
Descrição:	O sistema deve avançar automaticamente para o próximo cenário após a conclusão do anterior.	
Atores:	Jogador	
Prioridade:	Essencial	
Entradas e pré-condições:	<ul style="list-style-type: none"><li>O cenário atual deve estar limpo e concluído.</li></ul>	
Saídas e pós-condições:	<ul style="list-style-type: none"><li>O próximo cenário é carregado.</li></ul>	
Fluxos de eventos		
Fluxo principal:	<ol style="list-style-type: none"><li>O jogador conclui o cenário atual.</li><li>O sistema carrega o próximo cenário automaticamente.</li></ol>	
Fluxo secundário 1:	<ol style="list-style-type: none"><li>Se o cenário não for concluído, o jogo permanece no mesmo estágio.</li></ol>	

[RF010]		
Nome:	Exibir tela de finalização	
Descrição:	O sistema deve exibir uma tela de finalização com mensagem educativa ao término do último cenário.	
Atores:	Jogador	
Prioridade:	Essencial	
Entradas e pré-condições:	<ul style="list-style-type: none"><li>O jogador deve concluir todos os cenários.</li></ul>	
Saídas e pós-condições:	<ul style="list-style-type: none"><li>O sistema apresenta mensagem educativa e encerra o jogo.</li></ul>	
Fluxos de eventos		
Fluxo principal:	<ol style="list-style-type: none"><li>O jogador conclui o último cenário.</li><li>O sistema exibe tela de finalização com mensagem educativa.</li></ol>	
Fluxo secundário 1:	-	

## REQUISITOS NÃO FUNCIONAIS

[RNF01]		
Nome:	Interface gráfica em 2D	
Descrição:	O jogo deve apresentar gráficos bidimensionais (2D).	
Atores:	Jogador	
Prioridade:	Alta	
Entradas e pré-condições:	<ul style="list-style-type: none"><li>• O jogo deve estar em execução.</li></ul>	
Saídas e pós-condições:	<ul style="list-style-type: none"><li>• O cenário é exibido em estilo 2D.</li></ul>	
Fluxos de eventos		
Fluxo principal:	-	
Fluxo secundário 1:	-	

[RNF02]		
Nome:	Usabilidade intuitiva na jogabilidade	
Descrição:	O jogo deve ser intuitivo, permitindo que os jogadores compreendam facilmente a dinâmica e consigam jogar normalmente, sem dificuldades para interagir com os controles ou objetivos do jogo.	
Atores:	Jogador	
Prioridade:	Alta	
Entradas e pré-condições:	<ul style="list-style-type: none"><li>Jogador deve ter acesso ao jogo.</li></ul>	
Saídas e pós-condições:	<ul style="list-style-type: none"><li>Jogador compreende facilmente os comandos e objetivos.</li></ul>	
Fluxos de eventos		
Fluxo principal:	-	
Fluxo secundário 1:	-	

[RNF03]		
Nome:	Feedback visual e sonoro para ações do jogador	
Descrição:	O sistema deve fornecer respostas visuais (animações, mudanças gráficas) e sonoras (efeitos de som) às ações do jogador.	
Atores:	Jogador	
Prioridade:	Média	
Entradas e pré-condições:		<ul style="list-style-type: none"><li>O jogador realiza uma ação (movimento, coleta, reciclagem).</li></ul>
Saídas e pós-condições:		<ul style="list-style-type: none"><li>O sistema responde com feedback correspondente.</li></ul>
Fluxos de eventos		
Fluxo principal:	-	
Fluxo secundário 1:	-	

[RNF04]		
Nome:	Mensagens educativas relacionadas aos ODS	
Descrição:	O sistema deve exibir mensagens de conscientização ambiental alinhadas aos Objetivos de Desenvolvimento Sustentável (ODS).	
Atores:	Jogador	
Prioridade:	Média	
Entradas e pré-condições:	<ul style="list-style-type: none"><li>Jogador deve concluir cenários ou ações importantes.</li></ul>	
Saídas e pós-condições:	<ul style="list-style-type: none"><li>Mensagem educativa é apresentada na tela.</li></ul>	
Fluxos de eventos		
Fluxo principal:	-	
Fluxo secundário 1:	-	