

REQUISITOS ECO RUNNER

REQUISITOS FUNCIONAIS

RF 01	
Nome:	Login de Usuário.
Descrição:	Permitir que o jogador acesse o jogo por meio de autenticação de usuário registrada no banco de dados PostgreSQL.
Atores:	Jogador, Sistema.
Prioridade:	Alta.
Entradas e pré-condições:	<ul style="list-style-type: none">• Jogador informa login e senha válidos.• O usuário já deve estar cadastrado no banco de dados.
Saídas e pós-condições:	<ul style="list-style-type: none">• Usuário autenticado com sucesso.• Caso não seja autenticado, exibir mensagem de erro.
Fluxos de eventos	
Fluxo principal:	<ol style="list-style-type: none">1. Jogador informa os dados necessários para a autenticação:<ol style="list-style-type: none">a. Nome de usuário.b. Senha.c. Clica no botão de continuar.2. O sistema valida os dados no banco PostgreSQL.3. Se válido, acesso liberado e progresso do jogador carregado e o jogo iniciado.
Fluxo secundário 1:	<ol style="list-style-type: none">1. Caso o login falhe, o sistema exibe mensagem de erro e permite nova tentativa, voltando ao passo 1 do fluxo principal.

RF 02	
Nome:	Cadastro de jogador
Descrição:	Um novo usuário deve se cadastrar para acessar o jogo, informando nickname (nome de usuário) e senha.
Atores:	Jogador, Sistema.
Prioridade:	Alta.
Entradas e pré-condições:	<ul style="list-style-type: none"> O usuário não pode possuir registro no banco de dados com o nickname informado.
Saídas e pós-condições:	<ul style="list-style-type: none"> Usuário cadastrado com sucesso caso seja informado um nickname sem registro anterior no banco de dados. Caso o nome de usuário informado possua registro no banco, é informado ao usuário uma mensagem de erro.
Fluxos de eventos	
Fluxo principal:	<ol style="list-style-type: none"> Jogador informa os dados necessários para o cadastro: <ol style="list-style-type: none"> Nome de usuário. Senha. Clica no botão de cadastrar. O sistema valida os dados no banco PostgreSQL. Se válido, acesso ao jogo liberado.
Fluxo secundário 1:	<ol style="list-style-type: none"> Caso o cadastro falhe, o sistema exibe ao usuário uma mensagem de erro.

RF 03		
Nome:	Coleta de Materiais Recicláveis (Fase 1).	
Descrição:	O jogador deve coletar 20 itens recicláveis durante a fase inicial.	
Atores:	Jogador, Sistema.	
Prioridade:	Alta.	
Entradas e pré-condições:	<ul style="list-style-type: none">• O usuário deve estar autenticado.	
Saídas e pós-condições:	<ul style="list-style-type: none">• Ao coletar todos os 20 itens, o jogador avança para a próxima fase.• Se o jogador colidir com 5 poluentes, ele perde e a fase é reiniciada.	
Fluxos de eventos		
Fluxo principal:	<ol style="list-style-type: none">1. O jogador move o personagem pelo cenário.2. O sistema registra cada item coletado.3. Quando todos os 20 itens forem coletados, a fase é concluída.	
Fluxo secundário 1:	<ol style="list-style-type: none">1. Caso o jogador não desvie de poluentes, perde vida/energia.2. Se perder as 5 vidas a fase reinicia.	

RF 04		
Nome:	Classificação dos Itens Recicláveis (Fase 2)	
Descrição:	O jogador deve separar os itens coletados em suas lixeiras corretas (azul, vermelha, verde e amarela). Ao final, ganha uma arma feita de material reciclado e sua vida é restaurada por completo.	
Atores:	Jogador, Sistema.	
Prioridade:	Alta.	
Entradas e pré-condições:	<ul style="list-style-type: none">• Itens da Fase 1 coletados.	
Saídas e pós-condições:	<ul style="list-style-type: none">• Itens classificados corretamente.• O jogador ganha uma arma de material reciclado.• O jogador avança para a fase final.	
Fluxos de eventos		
Fluxo principal:	<ol style="list-style-type: none">1. O sistema exibe as lixeiras de reciclagem.2. Jogador arrasta/solta ou seleciona a lixeira correspondente ao item.3. O sistema valida a escolha.4. Se correto, o item é aceito.	
Fluxo secundário 1:	<ol style="list-style-type: none">1. Caso o jogador escolha a lixeira errada, o sistema não permite a ação, o item continua na mão do personagem até que seja colocado na lixeira correta.	

RF 05	
Nome:	Combate contra Yluh.
Descrição:	O jogador deve derrotar o monstro da poluição utilizando a arma construída com materiais reciclados. Cada disparo acertado em Yluh elimina 1 das 20 vidas dele. Yluh lança 5 disparos (cada disparo num intervalo de 1 segundo) em um intervalo de tempo aleatório que se atingir o jogador, ele perde 1 vida. Se o jogador perde todas as vidas ele é derrotado e a fase é reiniciada.
Atores:	Jogador, Sistema.
Prioridade:	Alta.
Entradas e pré-condições:	<ul style="list-style-type: none"> • Fase 1 e 2 concluídas. • Arma construída.
Saídas e pós-condições:	<ul style="list-style-type: none"> • Yluh derrotado e jogo concluído.
Fluxos de eventos	
Fluxo principal:	<ol style="list-style-type: none"> 1. Sistema inicia a batalha. 2. O jogador utiliza a arma reciclada para disparar contra Yluh. 3. Cada disparo acertado reduz em 1 a vida do inimigo. 4. Se a vida de Yluh chegar a zero, o jogador vence. 5. É mostrado uma mensagem de parabéns. 6. Jogador clica no botão Jogar novamente.
Fluxo secundário 1:	<ol style="list-style-type: none"> 1. Caso o jogador perca toda a energia, o jogo exibe mensagem de game over e retoma ao passo 1 do fluxo principal.

RF 06		
Nome:	Pausar e Salvar Progresso.	
Descrição:	O jogador pode pausar o jogo a qualquer momento e retomar à fase ou sair com o progresso em que parou.	
Atores:	Jogador, Sistema.	
Prioridade:	Média.	
Entradas e pré-condições:	<ul style="list-style-type: none">• Jogo em andamento.	
Saídas e pós-condições:	<ul style="list-style-type: none">• Progresso armazenado no banco de dados.• Retomada a partir da fase salva.	
Fluxos de eventos		
Fluxo principal:	<ol style="list-style-type: none">1. O jogador seleciona “Pausar”.2. A tela de menu aparece.3. O jogador seleciona “Sair e salvar”.4. Os dados são salvos e o jogo é fechado.	
Fluxo secundário	<ol style="list-style-type: none">1. O jogador seleciona “Pausar”.2. A tela de menu aparece.3. O jogador seleciona “Retomar”.4. O jogo é retomado.	

REQUISITOS NÃO FUNCIONAIS

RNF 01		
Nome:	Tempo de Resposta.	
Descrição:	O jogo deve responder a comandos do jogador (movimento, coleta, combate) em até 200ms.	
Atores:	Sistema.	
Prioridade:	Alta.	
Entradas e pré-condições:	<ul style="list-style-type: none">• Jogo em execução.	
Saídas e pós-condições:	<ul style="list-style-type: none">• Ações executadas sem atrasos perceptíveis.	
Fluxos de eventos		
Fluxo principal:	<ol style="list-style-type: none">1. O jogador realiza um comando.2. O personagem responde na tela em até 200ms.	

RNF 02		
Nome:	Interface Amigável.	
Descrição:	O sistema deve apresentar interface intuitiva, com instruções claras em cada fase.	
Atores:	Sistema.	
Prioridade:	Alta.	
Entradas e pré-condições:		<ul style="list-style-type: none">• O jogador acessa uma fase de acordo com o progresso.
Saídas e pós-condições:		<ul style="list-style-type: none">• O jogador entende as ações sem necessidade de treinamento adicional.
Fluxos de eventos		
Fluxo principal:	<ol style="list-style-type: none">1. O jogador inicia uma nova fase.2. O sistema apresenta a fase e o objetivo.	

RNF 03		
Nome:	Execução Multiplataforma.	
Descrição:	O jogo deve rodar em sistemas operacionais Windows e Linux.	
Atores:	Sistema.	
Prioridade:	Média.	
Entradas e pré-condições:	<ul style="list-style-type: none">Sistema operacional compatível instalado.	
Saídas e pós-condições:	<ul style="list-style-type: none">Jogo executado corretamente nas plataformas.	
Fluxos de eventos		
Fluxo principal:	1. O jogo é iniciado normalmente independente do sistema operacional.	

RNF 04		
Nome:	Integridade do Progresso	
Descrição:	O sistema deve garantir que dados de login e progresso do jogador sejam gravados corretamente no banco de dados relacional PostgreSQL. A cada fase concluída o progresso é salvo no banco de dados. Se o usuário resolver sair do jogo pelo menu, os dados são salvos.	
Atores:	Sistema, Banco de Dados.	
Prioridade:	Alta.	
Entradas e pré-condições:	<ul style="list-style-type: none">• Ação de salvar progresso.	
Saídas e pós-condições:	<ul style="list-style-type: none">• Dados gravados de forma íntegra e recuperáveis em sessões futuras.	
Fluxos de eventos		
Fluxo principal:	<ol style="list-style-type: none">1. O jogador passa de fase.2. O sistema salva o progresso no banco de dados.	
Fluxo secundário 1:	<ol style="list-style-type: none">1. O jogador seleciona o menu.2. Seleciona o botão salvar e sair.3. O sistema salva o progresso no banco de dados.	