

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"> 1. Jogador informa os dados necessários para o cadastro: <ol style="list-style-type: none"> a. Nome de usuário. b. Senha. c. Clica no botão de cadastrar. 2. O sistema valida os dados no banco PostgreSQL. 3. Se válido, acesso ao jogo liberado.
Fluxo secundário 1:	<ol style="list-style-type: none"> 1. Caso o cadastro falhe, o sistema exibe ao usuário uma mensagem de erro.

RF 03		
Nome:	Coleta de Materiais Recicláveis (Fase 1).	
Descrição:	O objetivo do jogador é navegar pelo cenário da Fase 1 para coletar 20 itens recicláveis designados, enquanto desvia de obstáculos (poluentes) para evitar perder vidas.	
Atores:	Jogador, Sistema.	
Prioridade:	Alta.	
Entradas e pré-condições:		<ul style="list-style-type: none">• O usuário deve estar autenticado.• O jogador inicia a fase com 5 vidas.• O contador de itens recicláveis coletados inicia em 0/20.
Saídas e pós-condições:		<ul style="list-style-type: none">• Sucesso: Ao coletar o 20º item reciclável, a fase é concluída e o sistema transiciona o jogador para a próxima fase.• Falha: Ao perder a 5ª vida (vidas chegam a 0), a fase é reiniciada, retornando às pré-condições (vidas restauradas para 5, contador de itens zerado, e itens reposicionados).
Fluxos de eventos		
Fluxo principal:	<ol style="list-style-type: none">1. O jogador move o personagem pelo cenário.2. O personagem colide com um "item reciclável".3. O sistema remove o item do cenário.4. O sistema atualiza a interface do usuário (HUD) para incrementar o contador de itens .5. Os passos 1 a 4 se repetem até o contador atingir 20/20.6. Ao atingir 20/20, o sistema exibe uma mensagem de "Fase Concluída!" e aciona a pós-condição de sucesso.	
Fluxo secundário 1:	<ol style="list-style-type: none">1. Durante o Fluxo Principal, o personagem do jogador colide com um "poluente".2. O sistema desconta uma vida do total do jogador.3. O sistema atualiza a HUD para exibir o novo total de vidas.4. Se o total de vidas se tornar 0, o sistema aciona o Fluxo Alternativo A2.5. Senão, o Fluxo Principal é retomado.	

Fluxo secundário 2:	<ol style="list-style-type: none">1. O jogador perde sua última vida.2. O sistema exibe uma mensagem de "Fim de Jogo" ou "Tente Novamente".3. O sistema aciona a pós-condição de falha, reiniciando a fase.
----------------------------	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

RF 04	
Nome:	Classificação dos Itens Recicláveis (Fase 2)
Descrição:	Nesta fase de transição, o jogador deve classificar corretamente os 20 itens coletados na Fase 1, associando cada um à sua lixeira de reciclagem correspondente para construir um novo equipamento.
Atores:	Jogador, Sistema.
Prioridade:	Alta.
Entradas e pré-condições:	<ul style="list-style-type: none"> • Jogador completou com sucesso a Fase 1 (RF 03)
Saídas e pós-condições:	<ul style="list-style-type: none"> • Sucesso: <ul style="list-style-type: none"> ○ O jogador recebe a "Arma Reciclada" em seu inventário. ○ As vidas do jogador são totalmente restauradas. ○ O sistema salva o progresso e transiciona o jogador para a Fase Final.
Fluxos de eventos	
Fluxo principal:	<ol style="list-style-type: none"> 1. A interface da Fase 2 é exibida, mostrando as quatro lixeiras coloridas (azul, vermelha, verde, amarela) e a área de apresentação dos 20 itens coletados na fase 1. 2. Jogador arrasta/solta ou seleciona a lixeira correspondente ao item. 3. O sistema valida a escolha. 4. Se a escolha for correta, o sistema aciona o Fluxo secundário 1. 5. Senão, o sistema aciona o Fluxo secundário 2. 6. Após a classificação do último item, o sistema aciona as Pós-condições de Sucesso.
Fluxo secundário 1:	<ol style="list-style-type: none"> 1. O jogador associa um item à sua lixeira correta. 2. O sistema "aceita" o item fazendo ele sumir da tela. 3. O sistema apresenta o próximo item a ser classificado.
Fluxo secundário 2:	<ol style="list-style-type: none"> 1. O jogador associa um item a uma lixeira incorreta 2. O sistema "rejeita" o item. 3. O item permanece na tela até que seja feita a escolha 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">• O jogador completou as Fases 1 e 2• O jogador está equipado com a "Arma Reciclada"• A batalha inicia com o jogador possuindo seu total de 5 vidas.• O chefe Yluh inicia a batalha com um total de 20 vidas.
Saídas e pós-condições:		<ul style="list-style-type: none">• Sucesso: O chefe é derrotado. O sistema exibe a tela de finalização do jogo, parabenizando o jogador e oferecendo opções como "Jogar Novamente".• Falha: As vidas do jogador chegam a 0. O sistema exibe uma mensagem de "Fim de jogo" e reinicia a batalha (retornando às pré-condições de batalha).
Fluxos de eventos		
Fluxo principal:	<ol style="list-style-type: none">1. Sistema inicia a batalha.2. O jogador controla seu personagem, podendo se mover e atirar.3. O chefe Yluh executa seu padrão de ataque em intervalos. Ele não se movimenta no mapa.4. Enquanto vida de Yluh > 0 E vida do Jogador > 0:<ol style="list-style-type: none">a. Se o jogador atinge Yluh, o sistema aciona o Fluxo secundário 1.b. Se Yluh atinge o jogador, o sistema aciona o Fluxo secundário 2.	
Fluxo secundário 1:	<ol style="list-style-type: none">1. Um disparo da "Arma Reciclada" atinge Yluh.2. A vida de Yluh é reduzida em 1 ponto.3. A HUD da vida do chefe é atualizada.4. Se a vida de Yluh chegar a 0, a Pós-condição de Sucesso é acionada.	

Fluxo secundário 2:	<ol style="list-style-type: none">1. Um disparo de Yluh atinge o jogador.2. A vida do jogador é reduzida em 1 ponto.3. A HUD da vida do jogador é atualizada.4. Se a vida do jogador chegar a 0, a Pós-condição de Falha é acionada.
----------------------------	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

RF 06	
Nome:	Gerenciamento de Jogo: Pausar, Salvar e Carregar Progresso.
Descrição:	Permite que o jogador pause o jogo durante a jogabilidade ativa para retomar a sessão, ou para salvar o progresso e sair. O progresso salvo pode ser carregado posteriormente a partir do menu principal do jogo.
Atores:	Jogador, Sistema.
Prioridade:	Média.
Entradas e pré-condições:	<ul style="list-style-type: none"> • Para Pausar/Salvar: O jogador está em um estado de jogabilidade ativa (controlando o personagem nas Fases 1, 2 ou 3). • Para carregar: O jogador está no menu principal e existe um progresso salvo anteriormente.
Saídas e pós-condições:	<ul style="list-style-type: none"> • Jogo Salvo: O estado atual do progresso do jogador é armazenado de forma persistente. • Jogo Retomado: O jogo continua exatamente do ponto em que foi pausado. • Jogo Carregado: O sistema restaura o último estado de progresso salvo e inicia a jogabilidade a partir daquele ponto.
Fluxos de eventos	
Fluxo secundário 1:	<ol style="list-style-type: none"> 1. Durante a jogabilidade ativa, o jogador aciona o comando "Pausar" (ex: tecla ESC) 2. O sistema congela a ação do jogo e exibe um menu de pausa com as opções: "Retomar", "Salvar e Sair" 3. O jogador seleciona "Retomar". 4. O menu é fechado e a jogabilidade é restaurada imediatamente.
Fluxo secundário 2:	<ol style="list-style-type: none"> 1. No menu de pausa, o jogador seleciona "Salvar e Sair". 2. O sistema salva os dados de progresso (conforme a definição acima) em um armazenamento persistente. 3. Após salvar, a aplicação do jogo é encerrada.
Fluxo secundário 3:	<ol style="list-style-type: none"> 1. O jogador seleciona a opção "Continuar" 2. O sistema verifica a existência de um progresso salvo 3. O sistema carrega os dados e inicia o jogo diretamente na fase e no estado em que foi salvo

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.	