

## REQUISITOS

### REQUISITOS FUNCIONAIS

RF01		
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"><li>• Jogador informa login e senha válidos.</li><li>• O usuário já deve estar cadastrado no banco de dados.</li></ul>	
Saídas e pós-condições:	<ul style="list-style-type: none"><li>• Usuário autenticado com sucesso.</li><li>• Caso não seja autenticado, exibir mensagem de erro.</li></ul>	
Fluxos de eventos		
Fluxo principal:	<ol style="list-style-type: none"><li>1. Jogador informa os dados necessários para a autenticação:<ol style="list-style-type: none"><li>a. Nome de usuário.</li><li>b. Senha.</li><li>c. Clica no botão de continuar.</li></ol></li><li>2. O sistema valida os dados no banco PostgreSQL.</li><li>3. Se válido, acesso liberado e progresso do jogador carregado e o jogo iniciado.</li></ol>	
Fluxo secundário 1:	<ol style="list-style-type: none"><li>1. Caso o login falhe, o sistema exibe mensagem de erro e permite nova tentativa, voltando ao passo 1 do fluxo principal.</li></ol>	

RF02		
Nome:	Cadastrar jogador	
Descrição:	Um novo usuário deve se cadastrar para acessar o jogo, informando nickname (nome de usuário) e senha. O nome de usuário deve conter no mínimo 3 caracteres e no máximo 12 e a senha deve conter entre 8 e 64 caracteres.	
Atores:	Jogador, Sistema.	
Prioridade:	Alta.	
Entradas e pré-condições:		<ul style="list-style-type: none"><li>● O usuário não pode possuir registro no banco de dados com o nickname informado.</li></ul>
Saídas e pós-condições:		<ul style="list-style-type: none"><li>● Usuário cadastrado com sucesso caso seja informado um nickname sem registro anterior no banco de dados.</li><li>● Caso o nome de usuário informado possua registro no banco, é informado ao usuário uma mensagem de erro.</li></ul>
Fluxos de eventos		
Fluxo principal:	<ol style="list-style-type: none"><li>1. Jogador informa os dados necessários para o cadastro:<ol style="list-style-type: none"><li>a. Nome de usuário (entre 3 e 12 caracteres).</li><li>b. Senha (no mínimo 8 e no máximo 64).</li><li>c. Clica no botão de cadastrar.</li></ol></li><li>2. O sistema valida os dados no banco PostgreSQL.</li><li>3. Se válido, acesso ao jogo liberado.</li></ol>	
Fluxo secundário 1:	<ol style="list-style-type: none"><li>1. Caso o cadastro falhe, o sistema exibe ao usuário uma mensagem de erro.</li></ol>	

RF03		
Nome:	Coletar 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"><li>● O usuário deve estar autenticado.</li><li>● O jogador inicia a fase com 5 vidas.</li><li>● O contador de itens recicláveis coletados inicia em 0/20.</li></ul>	
Saídas e pós-condições:	<ul style="list-style-type: none"><li>● Sucesso: Ao coletar o 20º item reciclável, a fase é concluída e o sistema transiciona o jogador para a próxima fase.</li><li>● 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).</li></ul>	
Fluxos de eventos		
Fluxo principal:	<ol style="list-style-type: none"><li>1. O jogador move o personagem pelo cenário.</li><li>2. O personagem colide com um "item reciclável".</li><li>3. O sistema remove o item do cenário.</li><li>4. O sistema atualiza a interface do usuário (HUD) para incrementar o contador de itens .</li><li>5. Os passos 1 a 4 se repetem até o contador atingir 20/20.</li><li>6. Ao atingir 20/20, o sistema exibe uma mensagem de "Fase Concluída!" e aciona a pós-condição de sucesso.</li></ol>	
Fluxo secundário 1:	<ol style="list-style-type: none"><li>1. Durante o Fluxo Principal, o personagem do jogador colide com um "poluente".</li><li>2. O sistema desconta uma vida do total do jogador.</li><li>3. O sistema atualiza a HUD para exibir o novo total de vidas.</li></ol>	

	<ol style="list-style-type: none"> <li>4. Se o total de vidas se tornar 0, o sistema aciona o Fluxo secundário 2.</li> <li>5. Senão, o Fluxo Principal é retomado.</li> </ol>
<b>Fluxo secundário 2:</b>	<ol style="list-style-type: none"> <li>1. O jogador perde sua última vida.</li> <li>2. O sistema exibe uma mensagem de "Fim de Jogo! Tente Novamente".</li> <li>3. O sistema aciona a pós-condição de falha, reiniciando a fase.</li> </ol>

RF04	
<b>Nome:</b>	Classificar Itens Recicláveis (Fase 2)
<b>Descrição:</b>	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.
<b>Atores:</b>	Jogador, Sistema.
<b>Prioridade:</b>	Alta.
<b>Entradas e pré-condições:</b>	<ul style="list-style-type: none"> <li>● Jogador completou com sucesso a Fase 1 (RF 03)</li> </ul>
<b>Saídas e pós-condições:</b>	<ul style="list-style-type: none"> <li>● Sucesso: <ul style="list-style-type: none"> <li>○ O jogador recebe a "Arma Reciclada".</li> <li>○ As vidas do jogador são totalmente restauradas.</li> <li>○ O sistema salva o progresso e transiciona o jogador para a Fase Final.</li> </ul> </li> </ul>
Fluxos de eventos	
<b>Fluxo principal:</b>	<ol style="list-style-type: none"> <li>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.</li> <li>2. Jogador arrasta/solta ou seleciona a lixeira correspondente ao item.</li> <li>3. O sistema valida a escolha.</li> <li>4. Se a escolha for correta, o sistema aciona o Fluxo secundário 1.</li> <li>5. Senão, o sistema aciona o Fluxo secundário 2.</li> <li>6. Após a classificação do último item, o sistema aciona as Pós-condições de Sucesso.</li> </ol>
<b>Fluxo secundário 1:</b>	<ol style="list-style-type: none"> <li>1. O jogador associa um item à sua lixeira correta.</li> <li>2. O sistema "aceita" o item fazendo ele sumir da tela.</li> <li>3. O sistema exibe a mensagem de acerto: "Correto! ponto para o meio ambiente."</li> </ol>

<b>Fluxo secundário 2:</b>	<ol style="list-style-type: none"><li>1. O jogador associa um item a uma lixeira incorreta.</li><li>2. O sistema "rejeita" o item.</li><li>3. O item permanece na tela até que seja feita a escolha correta.</li></ol>
----------------------------	--

RF05		
Nome:	Combater 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"><li>• O jogador completou as Fases 1 e 2</li><li>• O jogador está equipado com a "Arma Reciclada"</li><li>• A batalha inicia com o jogador possuindo seu total de 5 vidas.</li><li>• O chefe Yluh inicia a batalha com um total de 20 vidas.</li></ul>
Saídas e pós-condições:		<ul style="list-style-type: none"><li>• Sucesso: O chefe é derrotado. O sistema exibe a tela de finalização do jogo, parabenizando o jogador e oferecendo opções como "Jogar Novamente".</li><li>• 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).</li></ul>
Fluxos de eventos		
Fluxo principal:	<ol style="list-style-type: none"><li>1. Sistema inicia a batalha.</li><li>2. O jogador controla seu personagem, podendo se mover e atirar.</li><li>3. O chefe Yluh executa seu padrão de ataque em intervalos. Ele não se movimenta horizontalmente no mapa.</li><li>4. Enquanto vida de Yluh &gt; 0 e vida do Jogador &gt; 0:</li></ol>	

	<ul style="list-style-type: none"> <li>a. Se o jogador atinge Yluh, o sistema aciona o Fluxo secundário 1.</li> <li>b. Se Yluh atinge o jogador, o sistema aciona o Fluxo secundário 2.</li> </ul>
<b>Fluxo secundário 1:</b>	<ul style="list-style-type: none"> <li>1. Um disparo da "Arma Reciclada" atinge Yluh.</li> <li>2. A vida de Yluh é reduzida em 1 ponto.</li> <li>3. A HUD da vida do chefe é atualizada.</li> <li>4. Se a vida de Yluh chegar a 0, a Pós-condição de Sucesso é acionada.</li> </ul>
<b>Fluxo secundário 2:</b>	<ul style="list-style-type: none"> <li>1. Um disparo de Yluh atinge o jogador.</li> <li>2. A vida do jogador é reduzida em 1 ponto.</li> <li>3. A HUD da vida do jogador é atualizada.</li> <li>4. Se a vida do jogador chegar a 0, a Pós-condição de Falha é acionada.</li> </ul>



RF06	
<b>Nome:</b>	Gerenciar jogo: Pausar, Salvar e Carregar Progresso.
<b>Descrição:</b>	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.
<b>Atores:</b>	Jogador, Sistema.
<b>Prioridade:</b>	Média.
<b>Entradas e pré-condições:</b>	<ul style="list-style-type: none"> <li>• Para Pausar/Salvar: O jogador está em um estado de jogabilidade ativa (controlando o personagem nas Fases 1, 2 ou 3).</li> <li>• Para carregar: O jogador está no menu principal e existe um progresso salvo anteriormente.</li> </ul>
<b>Saídas e pós-condições:</b>	<ul style="list-style-type: none"> <li>• Jogo Salvo: O estado atual do progresso do jogador é armazenado de forma persistente.</li> <li>• Jogo Retomado: O jogo continua exatamente do ponto em que foi pausado.</li> <li>• Jogo Carregado: O sistema restaura o último estado de progresso salvo e inicia a jogabilidade a partir daquele ponto.</li> </ul>
Fluxos de eventos	
<b>Fluxo secundário 1:</b>	<ol style="list-style-type: none"> <li>1. Durante a jogabilidade ativa, o jogador aciona o comando "Pausar" (ex: tecla ESC)</li> <li>2. O sistema congela a ação do jogo e exibe um menu de pausa com as opções: "Retomar", "Salvar e Sair"</li> <li>3. O jogador seleciona "Retomar".</li> <li>4. O menu é fechado e a jogabilidade é restaurada imediatamente.</li> </ol>
<b>Fluxo secundário 2:</b>	<ol style="list-style-type: none"> <li>1. No menu de pausa, o jogador seleciona "Salvar e Sair".</li> <li>2. O sistema salva os dados de progresso (conforme a definição acima) em um armazenamento persistente.</li> <li>3. Após salvar, a aplicação do jogo é encerrada.</li> </ol>

<b>Fluxo secundário 3:</b>	<ol style="list-style-type: none"><li>1. O jogador seleciona a opção "Continuar" .</li><li>2. O sistema verifica a existência de um progresso salvo.</li><li>3. O sistema carrega os dados e inicia o jogo diretamente na fase e no estado em que foi salvo.</li></ol>
----------------------------	--

## REQUISITOS NÃO FUNCIONAIS

RNF01		
Nome:	Tempo de Resposta.	
Descrição:	O jogo deve responder visualmente aos comandos do jogador (movimento, coleta, combate) em até 200 milissegundos.	
Atores:	Sistema.	
Prioridade:	Alta.	
Entradas e pré-condições:		<ul style="list-style-type: none"><li>• Jogo em execução.</li></ul>
Saídas e pós-condições:		<ul style="list-style-type: none"><li>• A resposta visual ao comando do jogador é executada e exibida na tela dentro do tempo limite de 200ms.</li></ul>
Fluxos de eventos		
Fluxo principal:	<ol style="list-style-type: none"><li>1. O jogador realiza um comando.</li><li>2. O Sistema processa a entrada e renderiza a resposta correspondente em um intervalo de tempo inferior a 200ms a partir do comando inicial.</li></ol>	

RNF02		
<b>Nome:</b>	Consistência e Legibilidade da Interface.	
<b>Descrição:</b>	Para garantir que a interface seja de fácil aprendizado e utilização, todos os elementos de UI (menus, botões, HUD) devem seguir um padrão consistente, e toda informação textual deve atender a critérios mínimos de legibilidade.	
<b>Atores:</b>	Sistema.	
<b>Prioridade:</b>	Alta.	
<b>Entradas e pré-condições:</b>	<ul style="list-style-type: none"> <li>• O sistema renderiza qualquer tela que contenha elementos de interface ou texto.</li> </ul>	
<b>Saídas e pós-condições:</b>	<ul style="list-style-type: none"> <li>• Elementos com funções idênticas (ex: botões de "Confirmar" ou "Voltar") mantêm a mesma aparência, cor e posição relativa em todas as telas.</li> </ul>	

RNF03		
<b>Nome:</b>	Compatibilidade Windows/Linux.	
<b>Descrição:</b>	O jogo deve ser totalmente funcional e executável, atendendo a todos os demais requisitos (funcionais e não funcionais) nos sistemas operacionais Windows e Linux especificados abaixo.	
<b>Atores:</b>	Sistema.	
<b>Prioridade:</b>	Média.	
<b>Entradas e pré-condições:</b>	<ul style="list-style-type: none"> <li>• O ambiente de execução ou teste possui uma instalação de um dos seguintes sistemas operacionais: <ul style="list-style-type: none"> <li>○ Windows 10.</li> <li>○ Windows 11.</li> <li>○ Ubuntu 22.04 LTS.</li> </ul> </li> </ul>	
<b>Saídas e pós-condições:</b>	<ul style="list-style-type: none"> <li>• O jogo é instalado e iniciado com sucesso em cada um dos sistemas operacionais listados.</li> <li>• Todas as funcionalidades do jogo operam conforme especificado nos requisitos funcionais, sem apresentar erros específicos da plataforma.</li> </ul>	

RNF04	
<b>Nome:</b>	Integridade e Resiliência dos Dados Salvos.
<b>Descrição:</b>	O sistema deve garantir que os dados de progresso, uma vez salvos, permaneçam íntegros e possam ser recuperados em sessões futuras. O processo deve ser resiliente a falhas, como o fechamento inesperado do jogo.
<b>Atores:</b>	Sistema, Banco de Dados.
<b>Prioridade:</b>	Alta.
<b>Entradas e pré-condições:</b>	<ul style="list-style-type: none"> <li>• O jogo é iniciado após ter sido fechado (de forma normal ou forçada).</li> <li>• Existe um arquivo/registro de progresso salvo.</li> </ul>
<b>Saídas e pós-condições:</b>	<ul style="list-style-type: none"> <li>• Os dados de progresso carregados na nova sessão correspondem exatamente ao último ponto de salvamento bem-sucedido.</li> </ul>