

---

Nomes:

Matéria: Laboratório de Engenharia de Software

Ariany de Holanda Monteiro	RA.: 2760482411014
Beatriz Morais Lima	2760482411003
Carolina Fernanda Rodrigues Rabello	2760482311016
Italo Giandomenico Vieck	2760482411034
Lucas Henrique Barbosa	2760482411045

### Requisitos Funcionais (RF):

**RF01.** O jogo deve permitir que o jogador controle a personagem principal (andar, pular, atacar, interagir com cenários e personagens).

- Possibilitar que o jogador consiga controlar o protagonista, nos eixos X e Y, através de comandos no teclado ou controle, incluindo movimentos, ataques, combinações de ataques e movimentos, dentre outros aspectos.

**RF02.** O jogo deve conter um sistema de combate (ataques, colisões, dano, defesa, esquiva).

- O combate entre inimigos e o protagonista deve ser possível, permitindo que ambos causem e recebam dano ao receber um ataque diretamente. Além do dano recebido por ataques, o protagonista deve receber dano ao colidir com inimigos.

**RF03.** O jogo deve conter um sistema de fases (com mundos e níveis).

- O jogo será dividido em diferentes mundos, que serão chamados de “Universos”, que contém fases diferentes dentro deles. Atualmente, a ideia principal é que o projeto inclua 3 mundos com 5 fases cada.

**RF04.** O jogador deve ser capaz de derrotar inimigos.

- Ao causar dano suficiente ao inimigo, o mesmo deve ser derrotado, podendo ou não incluir uma recompensa para o jogador (**Item RF06**).

**RF05.** O jogo deve conter obstáculos e objetos destrutíveis.

- Deve ser possível, ao atacar, destruir objetos do cenário, como caixas, barris, pedras, mobílias, paredes e portas falsas, dentre outros. Ao destruir certos objetos, é possível receber recompensas, como dinheiro, itens menores ou revelar uma passagem secreta.

**RF06.** O jogador deve ganhar recompensas ao completar objetivos ou derrotar inimigos.

---

- O progresso durante as fases do jogo deve existir com base nas recompensas de combate, onde o jogador receberá itens menores (itens ativos com número de usos limitado, como bombas, melhorias temporárias, etc) ou itens maiores (passivos, melhorias permanentes, itens ativos de usos infinitos, etc.) ao derrotar inimigos e completar fases.

**RF07.** O jogo deve permitir múltiplas tentativas (replayability) com variações na experiência.

- O sistema de mapas, itens e inimigos deve fazer uso de randomização para garantir que cada tentativa seja única.

- No aspecto do mapa, os caminhos a serem seguidos serão diferentes, podendo levar a uma ordem de Universos diferente da tentativa anterior, com obstáculos diferentes em lugares variados.

- No aspecto de itens, as recompensas devem ser variadas e escolhidas de forma aleatória, ou seja, o mesmo inimigo não tem garantia de dar ao jogador o mesmo item que lhe foi dado em tentativas anteriores.

- Da mesma forma, inimigos devem contar com um sistema de randomização para serem inseridos no mapa, fazendo com que, mesmo que o jogador esteja em um Universo que já esteve antes, os inimigos encontrados serão em números e ordens diferentes.

**RF08.** O jogo deve conter uma interface gráfica com HUD (vida, energia, inventário, etc.).

- A interface gráfica deve ser simples e amigável, permitindo que o usuário tenha acesso à informações chave, como vida e energia, sem nenhum esforço. Dessa forma, o usuário pode manter a atenção na jogabilidade. A HUD deve ser temática para cada Universo.

**RF09.** O jogo deve ter um sistema de narrativa com eventos ou diálogos (ex: com o mascote ou Hiki).

- Durante os acontecimentos do jogo, seja combate (vitórias ou derrotas), exploração ou interações, eventos de narrativa e diálogos podem ocorrer, onde a protagonista terá conversas com seu mascote, com NPCs, monólogos internos ou um narrador neutro descreve o cenário atual.

**RF10.** O jogo deve salvar os dados do jogador entre sessões (mesmo que simplificado).

- Dados como quantidade de abates, mortes, dano e quantidade de tempo jogado devem ser armazenados e podem ser acessados pelo jogador no menu do jogo.

**RF11.** O jogo deve apresentar uma introdução com a história inicial.

---

- Ao iniciar, o jogador deve ser apresentado, através de uma cutscene, a história e trama principal do jogo.

**RF12.** O jogador deve ser capaz de tomar decisões que influenciam o rumo da história (matar ou não matar um inimigo).

- O jogo apresentará caminhos e finais alternativos que refletirão sobre as escolhas feitas pelo jogador durante a exploração.

**RF13.** Ao final do jogo, o jogador deve receber conquistas.

- O jogo deve ter um sistema de conquistas, para ao completar fases específicas, missões, ou realizar certas ações, o jogador receba recompensas equivalente a conquista, que podem variar de itens que podem ser encontrados em futuras tentativas, até novos personagens, mascotes ou até Universos.

**RF14.** O jogo deve apresentar uma trilha sonora única.

- A parte musical do jogo deverá ter autoria própria, buscando ajuda de terceiros para sua composição.

**RF15.** O jogo deve apresentar artes próprias.

- A parte artística do jogo deve ser original, tanto feita pelos devs ou por comissões externas. Visando evitar direitos autorais.

**RF16.** O jogo deve apresentar a opção de desistir da fase.

- O jogador pode, ao abrir o menu, apertar o botão para desistir, consequentemente perdendo o jogo e começando uma nova jogatina.

**RF17.** O jogador deve ser capaz de morrer no jogo.

- Quando a barra de vida do jogador chegar a zero, uma pequena cena de morte será ativada e o jogador será notificado com uma “Janela de Morte”, onde serão apresentados os status da tentativa (Universo e fase em que chegou, qual inimigo o finalizou, tempo total da tentativa, etc).

**RF18.** O jogo deve apresentar um mascote, amigo da personagem principal.

- Ao iniciar o jogo, um mascote deverá ser introduzido ao jogador, ensinando comandos básicos do jogo e ajudando a introduzir a narrativa.

### **Requisitos Não Funcionais (RNF):**

---

**RNF01.** O jogo deve ser desenvolvido utilizando a engine **Godot** com scripts em **GDScrip**t.

**RNF02.** O jogo deve ser compatível com a plataforma **desktop (Windows/Linux)**.

**RNF03.** O tempo de carregamento de cada fase não deve ultrapassar 3 segundos.

**RNF04.** A interface do jogo deve ser intuitiva e responsiva.

**RNF05.** O desempenho do jogo deve ser fluido em computadores com configuração intermediária.

**RNF06.** O jogo deve utilizar assets próprios ou com licença livre.

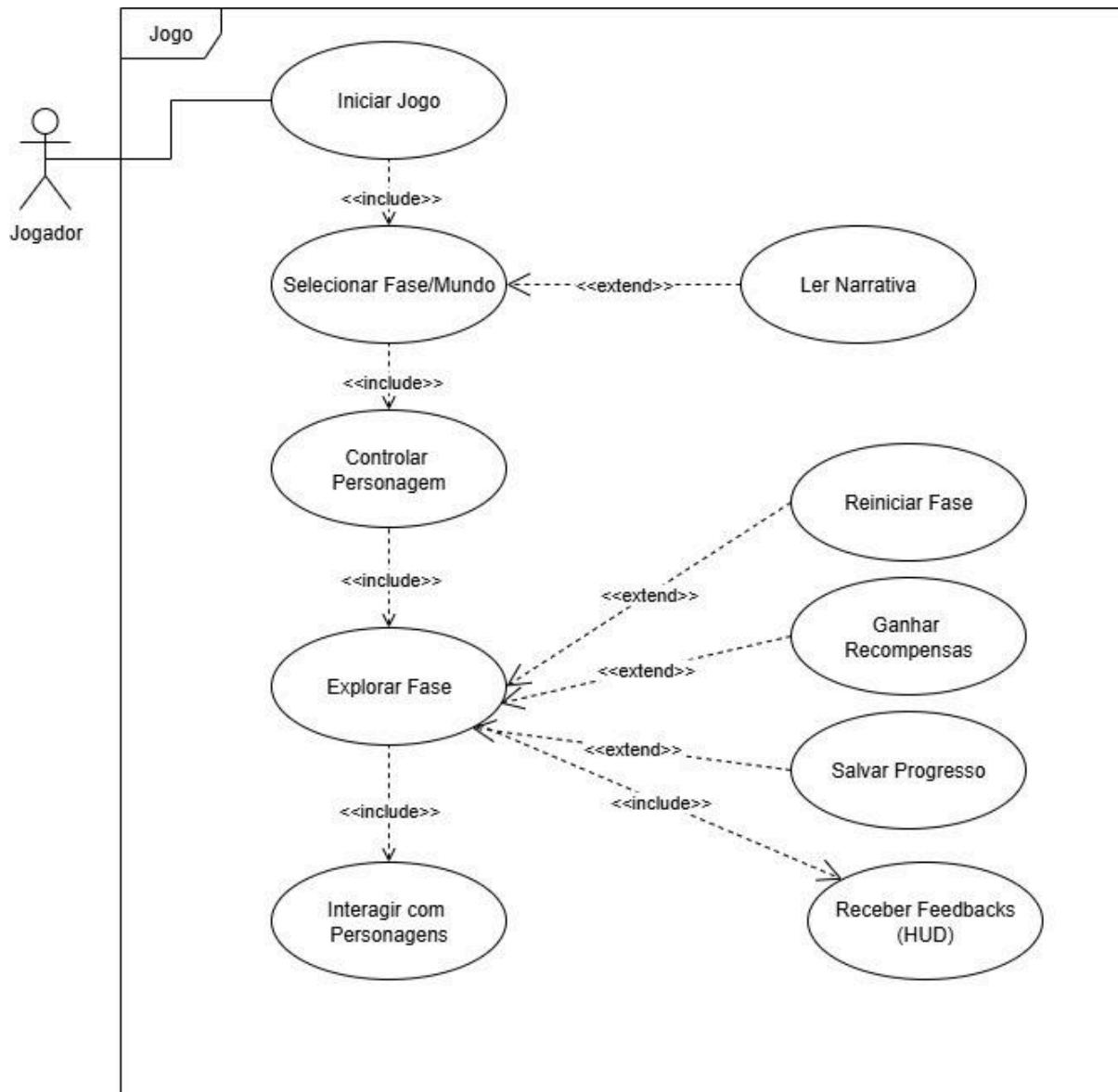
**RNF07.** O jogo deve apresentar uma taxa de quadros estável (mínimo 30 FPS).

**RNF08.** O código-fonte do jogo deve ser modular e de fácil manutenção.

**RNF09.** O jogo deve permitir fácil expansão de conteúdo (ex: novos mundos e fases).

## Diagrama de Caso de Uso

[https://drive.google.com/file/d/1rgwn6ny\\_y4DnEg-I22snBpYkzhw\\_7dNy/view](https://drive.google.com/file/d/1rgwn6ny_y4DnEg-I22snBpYkzhw_7dNy/view)



## Narrativas de Caso de Uso

<b>Nome do caso de uso</b>	Controlar Personagem
Requisitos Funcionais contemplados	RF01 RF02

Ator Principal	Jogador
Resumo	O jogador deve conseguir controlar a personagem para se mover, pular e atacar.
Pré-condições	O jogador deve ter um teclado funcional. O jogador deve ter iniciado o jogo. O jogador deve estar em uma fase. O jogador deve utilizar as teclas do teclado.
Pós-condições	O jogador deve visualizar o personagem realizando a movimentação.
Fluxo Principal	O jogador inicia a fase. O sistema habilita os controles. O jogador move-se pelas setas ou teclas direcionais. O jogador usa teclas específicas para pular ou atacar. O sistema responde aos comandos com a animação e movimentação adequadas.
Fluxo Alternativo	O jogador não utiliza as teclas e não se move.

<b>Nome do caso de uso</b>	Explorar Fase
Requisitos Funcionais contemplados	RF03 RF04 RF05 RF08
Autor Principal	Jogador
Resumo	O jogador explora a fase em busca de objetivos, enfrentando inimigos e evitando obstáculos.
Pré-condições	O jogador deve ter iniciado o jogo. O jogador deve estar em uma fase. O jogador deve utilizar as teclas do teclado para movimentação.
Pós-condições	O jogador ao se movimentar, deve entrar em contato com inimigos e obstáculos.
Fluxo Principal	A fase é carregada. O jogador inicia a exploração. O sistema apresenta obstáculos e inimigos.

	O jogador interage com o ambiente (destrói objetos, coleta itens, derrota inimigos). O jogador chega ao final da fase. O sistema exibe o resultado da fase.
Fluxo Alternativo	O jogador desiste de explorar a fase.

<b>Nome do caso de uso</b>	Ganhar Recompensas
Requisitos Funcionais contemplados	RF06 RF13
Autor Principal	Sistema do jogo
Resumo	O jogador recebe recompensas por concluir fases ou derrotar inimigos.
Pré-condições	O jogador deve ter iniciado o jogo. O jogador deve estar em uma fase ou ter concluído uma. O jogador deve ter eliminado um inimigo.
Pós-condições	O jogador recebe itens, moedas e/ou armas.
Fluxo Principal	O jogador derrota um inimigo ou cumpre um objetivo. O sistema verifica a condição de recompensa. O sistema entrega o item, ponto de experiência, ou melhoria. A HUD é atualizada com os novos dados do jogador.
Fluxo Alternativo	O jogador não captura a recompensa.

### Plano Inicial de Testes de Aceitação

#### Objetivo:

Verificar se o sistema (jogo) atende aos requisitos funcionais esperados do ponto de vista do usuário final (jogador), garantindo que as funcionalidades principais estejam operando corretamente.

ID	Requisito Relacionado	Caso de Teste	Critério de Aceitação	Resultado Esperado
CT01	RF01	Controlar a personagem para andar	A personagem deve se mover à esquerda e direita com fluidez	Movimentação responsiva e sem travamentos
CT02	RF01	Pular obstáculos	A personagem deve pular ao pressionar a tecla correspondente	A personagem realiza o salto de forma suave e previsível
CT03	RF02	Atacar um inimigo	Ao pressionar o botão de ataque, o personagem realiza a ação	Animação de ataque e dano aplicados corretamente
CT04	RF04	Derrotar inimigos	Inimigos devem perder vida ao serem atacados e desaparecer ao zerar vida	Inimigos somem após vida = 0
CT05	RF03	Avançar para a próxima fase após concluir a atual	Ao completar a fase, a próxima deve ser desbloqueada	Tela de transição aparece e nova fase é liberada

CT06	RF06	Receber recompensas após objetivos	Ao concluir uma fase ou derrotar inimigo, jogador recebe recompensa	Itens, moedas ou upgrades são exibidos na HUD
CT07	RF08	Aplicar melhorias ao personagem	Jogador deve conseguir aplicar melhorias no personagem	Personagem exibe status atualizado
CT08	RF09	Exibir informações na HUD	HUD deve mostrar vida, energia e inventário durante a partida	Informações visíveis e atualizadas em tempo real
CT09	RF10	Interagir com o mascote ou outros personagens	Personagem pode iniciar diálogo ou evento com outros personagens	Caixa de diálogo aparece com texto relacionado
CT10	RF11	Salvar progresso após completar uma fase	Jogo salva estado atual ao completar fase	Progresso pode ser carregado após reiniciar o jogo

CT11	RF12	Exibir introdução da história	Ao iniciar o jogo pela primeira vez, narrativa é apresentada	Cenas e textos de introdução são mostrados antes da gameplay
CT12	RF07	Jogar novamente com experiências diferentes	Cada nova tentativa gera uma variação de inimigos ou obstáculos	Fases ou desafios mudam entre sessões (randomizados)
CT13	RF13	Terminar uma jogatina, ou missão específica	A conquista aparece ao terminar o requisito	Conquista fica salva após ser concluída
CT14	RF14	Tocar a música planejada no cenário	A música toca normalmente durante a fase	A música toca na totalidade da fase, deixando o jogo mais imersivo
CT15	RF15	Implantar as artes	As artes ficam certas ao iniciar a fase	Artes ficam em resolução boas e nos devidos lugares denominados

CT16	RF16	Interagir com o botão de desistir	O botão for clicado e uma ação ocorrer	Aparecer a tela de derrota, com a opção de recomeçar ou ir ao menu
CT17	RF17	O jogador tomar dano letal	O protagonista faz a ação de “morrer” normalmente	O protagonista faz a ação de morrer, a tela de game over aparece com as opção de recomeçar ou ir ao menu
C18	RF18	Exibir o mascote no hub do jogo	O mascote aparece	O mascote se apresenta e mostra alguns tutoriais do jogo
C19	RNF03	Passar de fase	O personagem avança para outra fase	O tempo de passar de fase não deve ultrapassar 3 segundos
C20	RNF05	Iniciar o jogo em Computadores de média performance	O jogo apresentar consistência de comportamento	O jogo rodar fluidamente durante todo seu total

C21	RNF07	Iniciar o jogo na resolução máxima	O jogo apresentar consistência de comportamento	A taxa de frames por segundo for acima 30 durante a maioria do jogo
-----	-------	------------------------------------	---	---

Critérios para Aprovação do Sistema:

- Todos os casos de teste devem ser executados ao menos uma vez.
- Pelo menos 80% dos testes devem ser aprovados para considerar o sistema pronto para entrega da disciplina.
- Bugs críticos devem ser corrigidos antes da entrega final.