

1. Introdução

NERITH é um jogo de ação em plataforma 2D com arte em **pixel art 16-bit**, estilo **rogue lite** e foco em **combate, precisão e superação**. Desenvolvido em **Unity (URP)** com **linguagem C#**, o projeto foi executado via **arquivo .exe** para sistemas operacionais Windows. O jogo é fruto do esforço de uma equipe de estudantes do curso de **Engenharia de Software**, aplicando na prática princípios de design, arquitetura de software e metodologias ágeis como o **Scrum**. O projeto segue as boas práticas de desenvolvimento orientadas pelas **Normas Técnicas e Legais**:

- **ISO/IEC 12207**: Ciclo de vida do software.
- **ISO/IEC 25010**: Qualidade do produto de software.
- **ISO/IEC 29110**: Desenvolvimento para pequenas equipes.
- **ISO/IEC 23026**: Usabilidade e estrutura de software interativo.
- **LGPD (Lei 13.709/2018)**: Adotada preventivamente para segurança e transparência.

ISO/IEC 12207 – Ciclo de Vida do Software

Define um modelo completo para o ciclo de vida de sistemas e software. Ela estrutura as **atividades desde o planejamento até a manutenção**, incluindo desenvolvimento, testes, validação e entrega.

Aplicação no NERITH: A equipe seguiu etapas como concepção (GDD), planejamento (Trello), desenvolvimento (Unity), testes e entrega (.exe), dentro de um ciclo bem delimitado.

ISO/IEC 25010 – Qualidade de Produto de Software

Define as **características de qualidade** de um software, como:

- Usabilidade
- Desempenho
- Segurança
- Manutenibilidade
- Portabilidade

Aplicação no NERITH: NERITH foi planejado para ter jogabilidade fluida, fácil uso (controles claros), performance estável e código modular, seguindo os atributos da norma.

ISO/IEC 29110 – Desenvolvimento para Pequenas Equipes

Voltada para **pequenas organizações ou times educacionais**, como grupos universitários. Simplifica a ISO 12207, mantendo foco em **planejamento, rastreabilidade e entrega incremental**.

Aplicação no NERITH: A equipe de estudantes organizou tarefas por Sprints (Scrum), documentou as decisões e focou na entrega de um MVP funcional e testado.

ISO/IEC 23026 – Documentação e Interface Interativa

Estabelece diretrizes para **navegação, usabilidade e estrutura interativa** de sistemas com interface gráfica. Inclui boas práticas para menus, hierarquia de telas, feedback ao usuário e layout.

Aplicação no NERITH: A interface do jogo foi desenhada com foco em simplicidade, clareza da HUD e fluidez na navegação, respeitando os princípios de usabilidade definidos por essa norma.

Respeitamos os princípios da Lei Geral de Proteção de Dados (LGPD), neste projeto, ainda não se aplica, pois não há manipulação de dados sensíveis de usuários.

2. Enredo e Objetivo

O jogador assume o papel de **NERITH**, o último samurai de um clã extinto, herdeiro de habilidades marciais e poderes espirituais. Em um universo fictício corrompido por forças sombrias, seu objetivo é **restaurar o equilíbrio**, enfrentando inimigos em fases sucessivas com obstáculos únicos e desafios crescentes.

Elementos de gameplay:

- Ao sofrer danos, a **barra de vida** de NERITH diminui gradualmente.
- **Dano fatal reinicia o jogo e zera a pontuação.**
- A performance do jogador é avaliada com base em:
 - ✓ Velocidade de conclusão;
 - ✓ Número de inimigos derrotados;
 - ✓ Dano sofrido.

A pontuação é salva localmente (ranking interno) e exibida ao final das partidas, incentivando a rejogabilidade bem como estimulando a competitividade saudável entre jogadores.

3. Levantamento de Requisitos

Requisitos Funcionais:

- RF01: Controlar o personagem NERITH.
- RF02: Progredir por fases com obstáculos e inimigos.
- RF03: Reiniciar o jogo ao fim da vida e zerar pontuação.
- RF04: Gerar ranking interno baseado em performance.
- RF05: Exibir HUD com barra de vida.

Requisitos Não Funcionais:

- RNF01: Rodar via executável (.exe) em Windows.
- RNF02: Carregar fases em até 10 segundos.
- RNF03: Seguir padrões Clean Code e modularização.
- RNF04: Alinhar-se à norma ISO/IEC 25010.

4. Ferramentas Utilizadas

Ferramenta	Função
Unity (URP)	Engine de desenvolvimento
C#	Linguagem de scripts
GitHub	Versionamento
Trello	Organização ágil (Scrum)
Google Meet	Comunicação remota
PDF/Markdown	Documentação e registro

5. Estrutura de Diretórios (Assets)

Assets/

- |— BarraDeVida/=> HUD com barra de vida do jogador(HealthSystem.cs)
- |— Player/ => Sprites, animações e PlayerController.cs
- |— Scripts/ => lógica principal (GameManager.cs, EnemyAI.cs, ScoreSystem.cs, etc.)
- |— Scenes/ => Cenas jogáveis, tela inicial, fase 1, fase 2 e tela de derrota
- |— Resources/ => Prefabs, áudios, imagens e ScriptableObjects
- |— Monster_Creatures/ => Sprites e IA dos inimigos (Inimigos por fase)
- |— craftpix-* => Tilesets e assets visuais das fases (pixel art)

*Plataforma .net especializada em Assets confiáveis

6. Arquitetura do Jogo

Modelo Arquitetural: Híbrido (Componentes + MVC)

Principais Componentes:

- GameManager.cs: Gerencia o estado global do jogo
- HealthSystem.cs: Sistema modular de vida
- EnemyAI.cs: Lógica de perseguição e ataque inimigo
- ScoreSystem.cs: Cálculo e persistência de pontuação

Pipeline e Sistemas:

- Render: **Universal Render Pipeline (URP) - Unity**
- Input: **Unity Input System**
- Modularização com uso de **ScriptableObjects**

7. Mecânicas e Gameplay

- Estilo: Plataforma 2D com progressão linear
- Personagem principal: Samurai NERITH
- Fases: Diferentes ambientações, obstáculos e inimigos
- HUD: Vida, pontuação, ranking
- Ranking: Sistema interno baseado em performance

Controles do Jogo WAD ou Setas direcionais – Movimentação - Espaço – Ataque:

- ✓ Tecla “W” ou “Seta para cima” – Pular
- ✓ Tecla “P” – Pausa narrativa com caixa de diálogo reflexiva
- ✓ Tecla “ESC” – Menu de pausa com opções de controle, volume e comandos
- ✓ Enter – Retorna ao jogo após pausa reflexiva
- ✓ Barra de Espaço – Atacar / Executar animações de golpe com a espada



8. Equipe e Papéis

Nome	Função
Luan Brandão	Scrum Master, Game Designer, Desenvolvedor
Marcos Reis	Project Manager, Planejamento, Documentação
Nicolas Oliveira	Desenvolvedor: Criação, Scripts, Animações
Denis Bomfim	UI, Barra de Vida, Sonoplastia
Angélica Silva	UI, Suporte
Caio Cesar	UI, Barra de Vida
Paulo Victor	Criação de Telas, Interface
Yago Braga	Interface e Suporte

9. Testes Realizados

- **Funcionais:** Vida, ataques, colisões, fases
- **Exploratórios:** Jogabilidade, balanceamento, interface
- **Desempenho:** Build .exe testado com fluidez
- **Testes com jogadores internos:** Registros qualitativos

10. Roadmap e Sprints

- **Sprint final:**
 - ✓ Finalização de fases
 - ✓ Sistema de pontuação e ranking
 - ✓ Ajustes em documentação e apresentação (P2)
 - ✓ Testes e refinamento visual

11. Futuras Melhorias

- Melhorias gráficas: parallax, shaders, partículas
- Sistema de salvamento de progresso, permitindo continuar de onde parou
- Ranking online com integração real a banco de dados remoto (Firebase, Supabase ou MySQL)
- Novas fases e inimigos temáticos, com IA aprimorada e cenários variados
- Modo Infinito (Endless) ou Boss Rush, desafiando jogadores avançados
- Melhorias gráficas, incluindo efeitos parallax, shaders dinâmicos e camadas de fundo interativas
- Trilha sonora dinâmica e efeitos sonoros reativos à gameplay
- Sistema de habilidades especiais desbloqueáveis por performance ou tempo de jogo

12. Conclusão

NERITH representa não apenas um jogo, mas uma jornada educacional, cujo Educador, é o professor Fabrício Dias, de um time em formação em Engenharia de Software que aliou teoria e prática. A experiência com Unity, C#, Scrum e boas práticas de desenvolvimento permitiu à equipe entregar um **Produto Mínimo Viável** (MVP) sólido, funcional e promissor. A documentação, arquitetura e execução respeitam normas reconhecidas e demonstram o potencial da aplicação da engenharia de software no universo dos jogos digitais.