

**INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE  
SÃO PAULO – CAMPUS CAMPOS DO JORDÃO**

**ANDREY DE SOUZA SETÚBAL DESTRO**

**CAVALEIRO DE CAMPOS**

**CAMPOS DO JORDÃO  
2025**

## RESUMO

O projeto “Cavaleiro de Campos” apresenta um protótipo de jogo digital 2D desenvolvido com Phaser 3 e empacotado via Vite, no qual o jogador assume o papel de um cavaleiro responsável por defender as plantações da vila de Campos. O laço narrativo fundamenta-se na recuperação do cristal do sol, elemento vital para a subsistência da comunidade rural. O trabalho integra cenas independentes (título, lore e gameplay), HUD dinâmico com barra de vida vinculada ao personagem, pontuação acumulativa e controles responsivos (andar, pular, atacar e defender). A proposta enfatiza feedback audiovisual consistente (trilhas e FX sincronizados), interface padronizada com tipografia retrô e comentários didáticos espalhados pelo código, visando auxiliar estudantes na compreensão das responsabilidades de cada módulo. Foram aplicados princípios de design iterativo e boas práticas de organização de código, resultando em um protótipo funcional, rico em elementos técnicos e pedagógicos.

**Palavras-chave:** Desenvolvimento de Jogos 2D; Phaser 3; JavaScript.

## ABSTRACT

The project “Cavaleiro de Campos” presents a 2D digital game prototype developed with Phaser 3 and bundled via Vite, in which the player assumes the role of a knight responsible for defending the plantations of the village of Campos. The narrative premise is based on the recovery of the Sun Crystal, a vital element for the rural community's subsistence. The work integrates independent scenes (title, lore, and gameplay), a dynamic HUD with a health bar linked to the character, cumulative scoring, and responsive controls (walk, jump, attack, and defend). The proposal emphasizes consistent audiovisual feedback (synchronized soundtracks and FX), a standardized interface with retro typography, and didactic comments scattered throughout the code, aiming to assist students in understanding the responsibilities of each module. Iterative design principles and code organization best practices were applied, resulting in a functional prototype, rich in technical and pedagogical elements.

**Keywords:** 2D Game Development. Phaser 3; Didactic; JavaScript.

## SUMÁRIO

SUMÁRIO.....	4
1 INTRODUÇÃO .....	5
2 METODOLOGIA.....	5
3 RESULTADOS OBTIDOS.....	8
4 CONCLUSÃO .....	11
REFERÊNCIAS.....	13

## 1. INTRODUÇÃO

O desenvolvimento de “Cavaleiro de Campos” visa demonstrar competências na criação de jogos browser-based, contemplando desde a concepção narrativa até a implementação de mecânicas de combate e gerenciamento de estados. O objetivo principal é disponibilizar um protótipo que evidencie a integração harmoniosa entre enredo, jogabilidade e interface, permitindo sua apresentação em contexto acadêmico e posterior evolução para produto final.

A justificativa repousa na relevância de explorar ferramentas web modernas, possibilitando distribuição multiplataforma e fácil acesso para testes e avaliação. Além disso, o projeto reforça a identidade cultural ao ambientar o enredo em uma vila agrícola, valorizando temas locais adaptados à linguagem lúdica. Metodologicamente, adotou-se uma abordagem incremental: levantamento de requisitos narrativos e técnicos, definição de cenas, importação e animação de sprites, implementação das interações e posterior refinamento de áudio, HUD e comentários pedagógicos. O aporte teórico considera fundamentos de game design (estruturas de loop de jogo, curvas de dificuldade e feedback imediato) e referências em interface retro. Documentações oficiais do Phaser e literatura especializada (Rogers, Schell) serviram de guia para decisões de arquitetura e mecânicas, assegurando aderência às melhores práticas de mercado e adequação aos modelos acadêmicos de apresentação.

## 2. METODOLOGIA

O trabalho iniciou-se com a definição do escopo: três cenas principais (TitleScene, LoreScene e MainScene), mecânica de combate corpo a corpo, spawn periódico de inimigos Gorgon, controles de defesa e ataque, HUD com barra de vida e pontuação, além de tela de game over com feedback narrativo. As considerações iniciais incluíram a curadoria de assets: spritesheets do cavaleiro e da Gorgon (animações para idle, walk, attack, hurt, death), trilhas sonoras diferenciadas para título, gameplay e game over, e efeitos sonoros para ataques e danos.

As ferramentas utilizadas foram: Visual Studio Code para edição; Node.js 20.x e npm para gerenciamento de dependências; Git para versionamento local.

- Vite 7 como bundler e servidor de desenvolvimento, garantindo hot reload e feedback rápido durante os testes.
- Phaser 3.90.0 como engine principal, permitindo organização em cenas, uso de física arcade e manipulação de animações sprite-based.
- CSS com flexbox para centralizar o canvas, aplicar bordas pixeladas e manter responsividade em diferentes resoluções de janela.
- Fontes e efeitos: Google Fonts (“Press Start 2P”) e helper functions (applyTextEffects, createStyledText) para padronização tipográfica com sombras, strokes e opções de cor.

### **Estrutura do projeto:**

**main.js:** responsável por importar Phaser e entidades (Knight, Gorgon), definir constantes globais de resolução, padronizar texto, declarar e configurar as cenas, além de configurar o objeto config do jogo.

**Knight.js:** encapsula estado, física e animações do cavaleiro (movimentação, ataques, defesa, reações a dano).

**Gorgon.js:** controla IA simples do inimigo, incluindo janela de dano (damageWindowActive), disparo de efeitos sonoros e lógica de pontuação ao ser derrotado.

**style.css:** cuida da disposição do canvas, fontes, cores de fundo, borda temática e posicionamento global.

A implementação seguiu iterações semanais:

- Importações e testes de sprites, criação das cenas básicas.

- Ajuste de controles (andar/pular/ataque/defesa), centralização de HUD e cálculo dinâmico da barra de vida.
- Enriquecimento audiovisual (sword attack, gorgon attack, knight hurt, trilhas), balanceando volumes via playSfxQuiet.
- Inserção de auxiliares de texto, padronização de cores/sombras, responsividade e posicionamento dos botões.
- Documentação interna com comentários e reorganização do código, garantindo clareza para leitores iniciantes.

Os botões desempenham papel essencial no fluxo do projeto: a TitleScene oferece “Começar” (transição para LoreScene) e “Sair” (encerramento simbólico com mensagem de agradecimento); a MainScene encerra com “Reiniciar” (reinicia combate) e “Menu” (retorna ao título). Todos os botões estão encapsulados com setInteractive, incluem feedback visual (tint ao passar o mouse) e utilizam helpers para manter coerência visual e acessibilidade.

### 3. RESULTADOS OBTIDOS

O protótipo resultante reflete as decisões metodológicas através de funcionalidades concretas:

**Figura 1 – Tela Inicial:** o fundo “background\_start” dimensionado ao viewport cria ambientação imediata; o título “Cavaleiro de Campos” surge centralizado, com createStyledText aplicando cor dourada (#fcee09), sombras e contornos nítidos. O botão “Começar” destaca-se com cor semelhante e efeito de hover; o botão “Sair” reforça a navegação básica, disparando alerta agradecendo ao jogador. O rodapé “2025 · Desenvolvido por Andrey Destro” reforça autoria e contextualização temporal.

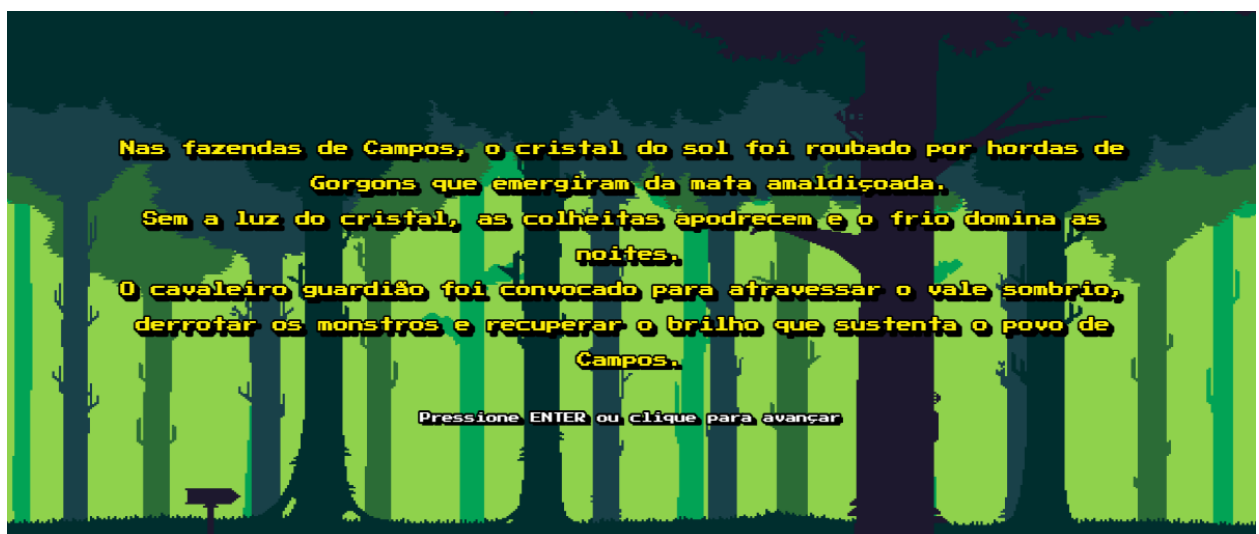
Figura 1 – Tela inicial



Fonte: O autor (2025)

**Figura 2 – Tela de Lore:** exibe narrativa em três parágrafos, com espaçamento e alinhamento central. A tipografia retro e efeito de sombra garantem legibilidade contra o fundo “background\_lore”. Um prompt textual (“Pressione ENTER ou clique para avançar”) funciona como botão implícito, aproveitando eventos de teclado e pointer para transição.

Figura 2 – Tela de Lore



Fonte: O autor (2025)



**Figura 3 – Gameplay:** apresenta o cavaleiro centralizado com animações responsivas (idle, walk, jump, attack, defend). A Gorgon surge em ondas com deslocamento horizontal e janela de dano controlada. A HUD inclui barra de vida centralizada na cabeça do cavaleiro (retângulos healthBarBg e healthBar) e texto “Pontuação: X” no canto superior esquerdo. A área respeita os limites do canvas, mantendo imersão e clareza.

Figura 3 – Gameplay



Fonte: O autor (2025)

**Figura 4 – HUD Detalhada:** close na barra de vida, demonstrando que healthBar.width reage proporcionalmente à variável health. O container acompanha o sprite do cavaleiro, reforçando a percepção de ligação entre personagem e status.

Figura 4 – HUD Detalhada



Fonte: O autor (2025)

**Figura 5 – Tela de Game Over:** overlay semitransparente evidencia transição de estado; texto “Fim de Jogo” em vermelho (#ff3b3b) destaca a derrota. A pontuação final e o lore de derrota reforçam a narrativa. Botões “Reiniciar” e “Menu” (mesma estética retro) garantem controle pós-derrota, reativando física ou navegando ao título conforme escolha.

Figura 5 – Tela de Game Over



Fonte: O autor (2025)

## 4. CONCLUSÃO

O protótipo alcança os objetivos de oferecer uma experiência consistente, tecnicamente sólida e didaticamente explicada. A estrutura modular, os helpers reutilizáveis e os comentários distribuídos fazem do projeto um recurso valioso para aulas práticas. A integração dos botões interativos assegura fluxo intuitivo entre cenas e evita rupturas na experiência do usuário, reforçando princípios de UX no contexto de jogos.

Sugestões técnicas para expansão futura incluem:

1. Introdução de novos inimigos com IA diferenciada (patrulha, projéteis) e telemetria de comportamento.
2. Implementação de sistema de power-ups e inventário, interagindo com o HUD (ex.: ícones adicionais, cooldowns).
3. Adoção de cutscenes interativas entre ondas, aproveitando a LoreScene para narrativa progressiva.
4. Portabilidade para dispositivos móveis com ajustes de controle (overlay de botões virtuais ou suporte a gamepad).
5. Otimização de performance (sprites atlases, lazy loading de áudio) e testes automatizados para manter estabilidade ao adicionar novos módulos.
6. Avaliação de acessibilidade: legendas para áudios críticos, opção de alto contraste e remapeamento de teclas.
7. Integração com placares online e modo cooperativo local para reforçar jogabilidade e engajamento comunitário.

## 7. REFERÊNCIAS

AMADOR, A. *Desenvolvimento de jogos digitais com Phaser*. São Paulo: Novatec, 2023.

PHASER.IO. *Phaser 3 API Documentation*. Disponível em: <https://newdocs.phaser.io/docs/3.90.0>. Acesso em: 24 nov. 2025.

ROGERS, S. *Level Up! The Guide to Great Video Game Design*. 2. ed. Wiley, 2014.

SANTOS, M. *Introdução ao Design de Jogos 2D*. Porto Alegre: Bookman, 2021.

SCHELL, J. *The Art of Game Design: A Book of Lenses*. 3. ed. CRC Press, 2019.

SILVA, R.; PEREIRA, T. Produção de jogos em HTML5 com Phaser. *Revista Brasileira de Jogos Digitais*, v. 5, n. 2, 2022.

VITEJS. *Vite – Next Generation Frontend Tooling*. Disponível em: <https://vitejs.dev>. Acesso em: 24 nov. 2025.