

Universidade Federal de Ouro Preto
Instituto de Ciências Exatas e Aplicadas
Departamento de Computação e Sistemas

Inteligência Artificial Generativa aplicada no design de jogos: uma análise exploratória e prática com LLMs e Modelos Multimodais

Guilherme Ribeiro Figueiredo

TRABALHO DE CONCLUSÃO DE CURSO

ORIENTAÇÃO:
Tiago França Melo de Lima

Julho, 2026
João Monlevade–MG

Guilherme Ribeiro Figueiredo

Inteligência Artificial Generativa aplicada no design de jogos: uma análise exploratória e prática com LLMs e Modelos Multimodais

Orientador: Tiago França Melo de Lima

Monografia apresentada ao curso de Engenharia de Computação do Instituto de Ciências Exatas e Aplicadas, da Universidade Federal de Ouro Preto, como requisito parcial para aprovação na Disciplina "Trabalho de Conclusão de Curso II".

Universidade Federal de Ouro Preto

João Monlevade

Julho de 2026

SISBIN - SISTEMA DE BIBLIOTECAS E INFORMAÇÃO

F475i Figueiredo, Guilherme Ribeiro.
Inteligência artificial generativa aplicada no design de jogos [manuscrito]: uma análise exploratória e prática com llms e modelos multimodais. / Guilherme Ribeiro Figueiredo. - 2026.
110 f.: il.: color., tab..

Orientador: Prof. Me. Tiago França Melo de Lima.
Monografia (Bacharelado). Universidade Federal de Ouro Preto. Instituto de Ciências Exatas e Aplicadas. Graduação em Engenharia de Computação .

1. Inteligência artificial generativa. 2. Jogos - Desenvolvimento. 3. Jogos - Designers. 4. Jogos eletrônicos - Desenvolvimento. 5. Jogos para computador - Projeto. I. Lima, Tiago França Melo de. II. Universidade Federal de Ouro Preto. III. Título.

CDU 004.96:004.89

Bibliotecário(a) Responsável: Flavia Reis - CRB6/2431

MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL DE OURO PRETO
REITORIA
INSTITUTO DE CIENCIAS EXATAS E APLICADAS
DEPARTAMENTO DE COMPUTACAO E SISTEMAS

FOLHA DE APROVAÇÃO

Guilherme Ribeiro Figueiredo

Inteligência Artificial Generativa aplicada no design de jogos: uma análise exploratória e prática com LLMs e modelos multimodais

Monografia apresentada ao Curso de Engenharia da Computação da Universidade Federal de Ouro Preto como requisito parcial para obtenção do título de Bacharelado em Engenharia da Computação.

Aprovada em 28 de julho de 2026

Membros da banca

Me. Tiago França Melo de Lima - Orientador (Universidade Federal de Ouro Preto)
Dr. Josemar Alves Caetano - Examinador (Universidade Federal de Ouro Preto)
Dr. Tadeu Moreira de Classe - Examinador (Universidade Federal do Estado do Rio de Janeiro)

Tiago França Melo de Lima, orientador do trabalho, aprovou a versão final e autorizou seu depósito na Biblioteca Digital de Trabalhos de Conclusão de Curso da UFOP em 29/09/2026

Documento assinado eletronicamente por **Tiago Franca Melo de Lima**, **PROFESSOR DE MAGISTERIO SUPERIOR**, em 29/09/2026, às 21:25, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do Decreto nº 8.539, de 8 de outubro de 2015.

A autenticidade deste documento pode ser conferida no site http://sei.ufop.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0 , informando o código verificador **1203877** e o código CRC **CF576C32**.

R. Diogo de Vasconcelos, 122, - Bairro Pilar Ouro Preto/MG, CEP 35402-163
Telefone: (31)3808-0819 - www.ufop.br

Criado por 05495948688, versão 2 por 05495948688 em 29/09/2026 21:24:49.

Este trabalho é dedicado a todos os amantes de jogos digitais que sempre sonharam em criar seus próprios universos, mas que enxergam a complexidade das múltiplas áreas do desenvolvimento como um grande desafio.

Agradecimentos

Agradeço inicialmente aos meus pais (Valéria e Maabson) e aos meus irmãos (Rodrigo e Camila) por sempre me apoiarem em minhas decisões, cada um à sua maneira, mas sempre lá quando foi preciso. Principalmente à minha mãe, por sempre estar presente ao meu lado e fazer o possível e o impossível para me apoiar em todos os aspectos da minha vida. E, não menos importante, ao meu pai, que, apesar da distância física, também sempre me apoiou da forma como era possível, e sempre oferecendo um ombro amigo para conversar quando preciso.

Agradeço de forma imensa a todos os meus amigos que não me deixaram desistir ao longo do processo, sempre se fazendo presentes para oferecer ajuda, independentemente da situação. Em especial ao Rafael Macedo, por se provar uma pessoa digna da palavra **AMIGO** e que serviu de inspiração para a escolha da minha formação; todos os conselhos e as oportunidades que me ofereceu me ajudaram fortemente a chegar até onde estou. Muito obrigado aos meus amigos de longa data, Guilherme S. e Victor B., por fazerem parte da minha vida e ajudarem a alegrar os meus dias durante tanto tempo.

Agradeço aos meus amigos, verdadeiros companheiros com os quais pude dividir meu tempo na graduação e compartilhar experiências, e que me ajudaram imensamente a seguir em frente e a finalizar esta etapa da minha vida: Roberto, Lucas, Francis, Pedro e Philipe.

Agradeço ao meu orientador, Tiago, por apoiar as minhas ideias desde o começo e me ajudar nesta grande empreitada, compartilhando o gosto pela área de jogos e ajudando a trazer um pouco mais desta incrível área para a universidade.

Sem vocês, nada disso teria sido possível! Muito obrigado!

E, por fim, agradeço aos leitores por dedicarem o seu tempo à leitura deste trabalho!

"O destino só te prende se você deixar. Faça o que é preciso, não porque está escrito."

— Kratos - Personagem Fictício (2018),
em: God of War (2018) - Santa Monica Studio - Los Angels - EUA.

Resumo

O desenvolvimento de jogos digitais exige competências em áreas distintas, como programação, arte visual e engenharia de áudio, o que impõe uma barreira de entrada significativa para desenvolvedores independentes. Este trabalho tem como objetivo desenvolver e documentar uma metodologia de apoio ao *design* e à produção de jogos digitais baseada em ferramentas de Inteligência Artificial Generativa (IA-Gen). A metodologia organiza a produção em cinco fases (concepção, planejamento da produção, seleção das ferramentas, geração dos ativos por *meta-prompting* e montagem na *game engine*), com refinamento iterativo dentro de cada uma. Para construí-la e verificá-la, a pesquisa adota uma abordagem exploratória organizada em etapas sequenciais, que vão da categorização dos ativos digitais necessários a um projeto de jogo, passando pelo teste comparativo de modelos generativos de texto, imagem, áudio e vídeo, até a aplicação desses recursos em três jogos elaborados como provas de conceito: *PinkBob J. e o Enigma do Rum*, *Hacking Survivors* e *Pafúncio vs. Desenhos*. Os testes comparativos, conduzidos entre novembro de 2025 e janeiro de 2026, apoiam-se em avaliação qualitativa feita exclusivamente pelo autor, com escala Likert ponderada para imagem e vídeo e análise comparativa para texto e áudio; após essa etapa, o escopo prático de geração restringe-se a texto, imagem e áudio, já que os modelos de vídeo apresentam instabilidade na permanência de objetos e custo elevado por geração. A seleção recai sobre o Claude 4.5 Sonnet para a estrutura narrativa (*story spine*), o Google Gemini 3 Pro para a expansão da ideia e o documento de *design*, o Flow Studio Nano Banana Pro para imagens, o Suno para trilhas sonoras e o Kling AI para efeitos sonoros. Na produção, o Gemini passa à versão 3.1 Pro. Nos três casos, o autor concebe, documenta e produz o jogo com os ativos gerados por essas ferramentas e registra os ajustes que cada projeto exige: dois jogos chegam à versão final, cada um com mais de uma centena de ativos gerados, e o terceiro, a uma versão de demonstração. A montagem e a programação na *game engine* permanecem trabalho autoral do desenvolvedor, com uso pontual de IA-Gen. Conclui-se que a IA-Gen se mostra um apoio viável para que um desenvolvedor independente produza os ativos de texto, imagem e áudio de um jogo, ainda que sujeito a custos recorrentes de uso, a instabilidades nas gerações e a incertezas quanto a direitos autorais. Os resultados dependem do julgamento de um único avaliador e não incluem linha de base de tempo nem validação sistemática com jogadores.

Palavras-chave: inteligência artificial generativa; *design* de jogos; geração de ativos digitais; jogos independentes; modelos generativos multimodais.

Abstract

Video game development demands expertise across distinct fields, such as programming, visual art, and audio engineering, which poses a significant barrier to entry for independent developers. This study aims to develop and document a methodology grounded in Generative Artificial Intelligence (GenAI) tools to support the design and production of digital games. The methodology organizes production into five phases (conception, production planning, tool selection, asset generation through meta-prompting, and assembly in the game engine), with iterative refinement within each one. To build and verify it, the research adopted an exploratory approach, organized into sequential stages that range from the categorization of the digital assets a game project requires, through the comparative testing of generative models for text, image, audio, and video, to the application of these resources in three games developed as proofs of concept: *PinkBob J. e o Enigma do Rum*, *Hacking Survivors*, and *Pafúncio vs. Desenhos*. The comparative tests, conducted between November 2025 and January 2026, relied on a qualitative assessment carried out by the author alone, using a weighted Likert scale for image and video and a comparative analysis for text and audio; after this stage, the practical scope of generation was narrowed to text, image, and audio, since the video models showed unstable object permanence and a high cost per generation. The tools selected were Claude 4.5 Sonnet for the story spine, Google Gemini 3 Pro for idea expansion and the design document, Flow Studio Nano Banana Pro for images, Suno for soundtracks, and Kling AI for sound effects. Production used Gemini 3.1 Pro. In all three cases, the author conceived, documented, and produced the game with the assets generated by these tools and recorded the adjustments each project required: two games reached a final version, each with more than a hundred generated assets, and the third reached a demo version. Assembly and programming in the game engine remained the developer's own work, with occasional use of GenAI. The study concludes that GenAI offers viable support for an independent developer to produce a game's text, image, and audio assets, although it remains subject to recurring usage costs, unstable generations, and copyright uncertainty. The results rely on the judgment of a single evaluator and include neither a time baseline nor systematic validation with players.

Keywords: generative artificial intelligence; game design; digital asset generation; independent games; multimodal generative models.

Lista de ilustrações

Figura 1 – Fluxo padrão de geração de ativos multimídia. 28
Figura 2 – Fluxo de geração automatizada com modelo agêntico. 28
Figura 3 – Comparativo da geração do personagem João entre diferentes modelos. 45
Figura 4 – Comparativo da geração do objeto Bússola do Capitão Barba Rosa entre diferentes modelos. 47
Figura 5 – Comparativo da geração de arte da capa entre diferentes modelos. . . . 47
Figura 6 – Personagem PinkBob conceitual usado para testes com vídeos. 53
Figura 7 – Árvore de ideias da ideia de jogo 32, que foi escolhida. 57
Figura 8 – Fluxograma do jogo PinkBob. 60
Figura 9 – Hierarquia de cenários, ações e personagens por episódios. 61
Figura 10 – Exemplo de protótipo da visão do jogador. 62
Figura 11 – Exemplo de protótipo de inventário. 62
Figura 12 – Exemplo de uso de item para solucionar *puzzle* de cenário. 63
Figura 13 – Exemplo de *sprite sheet* com todas as animações geradas do PinkBob. . 65
Figura 14 – Exemplo de *sprite sheet* com todas as animações da Capitã Maré Alta. 65
Figura 15 – Exemplo de *sprite sheet* com todas as animações da recepcionista do hotel. 65
Figura 16 – Exemplo de objeto gerado, “Anotações Misteriosas”, implementado no jogo. 66
Figura 17 – Exemplo de protótipo do interior da taverna desenhado pelo autor, à esquerda, e *asset* final gerado pela IA-Gen, à direita. 68
Figura 18 – Exemplo de protótipo do cenário de praia desenhado pelo autor, à esquerda, e *asset* final gerado pela IA-Gen, à direita. 68
Figura 19 – Exemplo de tela capturada da *engine* contendo o *plugin* de edição de diálogos. 69
Figura 20 – Exemplo de tela capturada do jogo contendo opções de diálogo. 70
Figura 21 – Exemplo de tela capturada da *engine* com o *plugin* de *cutscenes*. 70
Figura 22 – Exemplo de tela capturada do jogo finalizado com elementos gerados por IA-Gen. 71
Figura 23 – Exemplo de tela capturada do jogo finalizado ilustrando o cenário do bar no primeiro episódio. 72
Figura 24 – Exemplo de tela capturada do jogo finalizado ilustrando o cenário do quarto de hotel no primeiro episódio. 72
Figura 25 – Exemplo de tela capturada do jogo finalizado ilustrando o cenário do porto no segundo episódio. 73

Figura 26 – Exemplo de tela capturada do jogo finalizado ilustrando o cenário do naufrágio no terceiro episódio. 73
Figura 27 – Árvore de ideias do segundo estudo de caso, parte 1. 74
Figura 28 – Árvore de ideias do segundo estudo de caso, parte 2. 75
Figura 29 – Árvore de ideias do segundo estudo de caso, parte 3. 75
Figura 30 – Rascunho do protótipo desenvolvido para o segundo estudo de caso. . . 77
Figura 31 – Exemplo de *tiles* que não se conectam. 78
Figura 32 – Exemplo de objetos gerados para construção de *tiles* da fase 3. 79
Figura 33 – Exemplo de *tiles* da fase 3 criados usando os objetos gerados por IA-Gen. 80
Figura 34 – Exemplo de *sprite sheet* para a arma “Armadilha Zero-Day”. 80
Figura 35 – Exemplo de *sprite sheet* para o inimigo “Executivo Infectado” da primeira fase. 81
Figura 36 – Exemplo de um trecho da história em quadrinhos apresentada entre fases do jogo. 81
Figura 37 – Exemplo de tela capturada do jogo finalizado ilustrando a fase 1. . . . 83
Figura 38 – Exemplo de tela capturada do jogo finalizado ilustrando a fase 2. . . . 83
Figura 39 – Exemplo de tela capturada do jogo finalizado ilustrando a fase 3. . . . 84
Figura 40 – Exemplo de tela capturada do jogo finalizado ilustrando a fase 4. . . . 84
Figura 41 – Árvore de ideias do terceiro estudo de caso. 85
Figura 42 – Protótipo desenhado do jogo *Pafúncio vs. Desenhos*. 87
Figura 43 – Blocos gerados por IA-Gen utilizados na construção das fases. 88
Figura 44 – Exemplo de cena construída utilizando os blocos gerados por IA-Gen. . 89
Figura 45 – Exemplo da *sprite sheet* finalizada do personagem Pafúncio. 89
Figura 46 – Exemplo da *sprite sheet* finalizada do inimigo cachorro raivoso. 90
Figura 47 – Exemplo das camisas geradas para representar a vida do jogador. . . . 91
Figura 48 – Exemplo de tela capturada do jogo ilustrando o menu principal, em que o título aparece na grafia estilizada da arte (“Pafuncio Vs Desenhos”). . 92
Figura 49 – Exemplo de tela capturada do jogo ilustrando a interface de seleção de fases. 92
Figura 50 – Exemplo de tela capturada do jogo ilustrando as plataformas geradas por IA-Gen na construção da fase 1. 93
Figura 51 – Exemplo de tela capturada do jogo ilustrando as manchas de café e outros elementos gerados por IA-Gen na fase 1. 93

Lista de tabelas

Tabela 1 – Comparativo das áreas de aplicação de IA exploradas nos trabalhos relacionados e na presente pesquisa. 24
Tabela 2 – Métricas e pesos para análise qualitativa de imagens geradas por IA-Gen. 31
Tabela 3 – Métricas e pesos para análise qualitativa de vídeos gerados por IA-Gen. 32
Tabela 4 – Comparativo das ferramentas de IA-Gen para geração de texto (dezembro de 2025 a janeiro de 2026). 37
Tabela 5 – Exemplo do trecho “Todos os dias...” no texto de controle e em cada modelo. 42
Tabela 6 – Comparativo das ferramentas de IA-Gen para geração de imagens (novembro e dezembro de 2025). 43
Tabela 7 – Desempenho médio e métricas das ferramentas de imagem avaliadas. . 44
Tabela 8 – Especificações técnicas do computador utilizado para as gerações locais. 45
Tabela 9 – Comparativo das ferramentas de IA-Gen para geração de trilhas sonoras (novembro de 2025). 49
Tabela 10 – Comparativo das ferramentas de IA-Gen para geração de efeitos sonoros (novembro de 2025). 50
Tabela 11 – Comparativo das ferramentas de IA-Gen para geração de vídeos (novembro de 2025). 52
Tabela 12 – Desempenho médio e métricas das ferramentas de vídeo avaliadas. . . . 53
Tabela 13 – Conteúdo do repositório da pesquisa e seções que o citam. 108

Lista de abreviaturas e siglas

API *Application Programming Interface*

CFPO *Content Format Integrated Prompt Optimization*

GDD *Game Design Document*

GDF *Game Design File*

GIF *Graphics Interchange Format*

IA Inteligência Artificial

IA-Gen Inteligência Artificial Generativa

LLM *Large Language Model*

NPC *Non Playable Character*

SFX *Sound Effects*

Sumário

1 INTRODUÇÃO 15
1.1 Objetivos 16
1.2 Organização do trabalho 17

2 REVISÃO BIBLIOGRÁFICA 18
2.1 O processo de *design* e desenvolvimento de jogos 18
2.2 Inteligência Artificial Generativa e modelos multimodais 19
2.2.1 Engenharia de *prompt* 20
2.3 Aplicações práticas da IA-Gen na indústria 21
2.4 Trabalhos relacionados 22

3 METODOLOGIA 25
3.1 O método proposto 25
3.2 Etapa 1: categorização de ativos digitais em jogos 29
3.3 Etapa 2: pesquisa e testes exploratórios de ferramentas 29
3.4 Etapa 3: processo de ideação e seleção 32
3.5 Etapa 4: primeiro estudo de caso — *PinkBob J. e o Enigma do Rum* 33
3.6 Etapa 5: segundo estudo de caso — *Hacking Survivors* 34
3.7 Etapa 6: terceiro estudo de caso — *Pafúncio vs. Desenhos* 35

4 RESULTADOS 36
4.1 Etapa 1: categorização de ativos digitais em jogos 36
4.2 Etapa 2: pesquisa e testes exploratórios de ferramentas 36
4.2.1 Pesquisa e testes com modelos textuais 37
4.2.2 Pesquisa e testes com modelos de imagens 42
4.2.3 Pesquisa e testes com modelos de áudio 48
4.2.3.1 Pesquisa e testes com trilhas sonoras 48
4.2.3.2 Pesquisa e testes com efeitos sonoros 50
4.2.4 Pesquisa e testes com modelos de vídeo 51
4.3 Etapa 3: processo de ideação e seleção 54
4.3.1 Ideação inicial e filtragem qualitativa 54
4.4 Etapa 4: primeiro estudo de caso — *PinkBob J. e o Enigma do Rum* 57
4.4.1 Definição do estudo de caso 57
4.4.2 Expansão da história 59
4.4.3 Protótipos de baixa fidelidade 60
4.4.4 Desenvolvimento e gerações com IA-Gen 62

4.4.4.1 Geração de personagens . . . 63
4.4.4.2 Geração de objetos . . . 66
4.4.4.3 Geração de cenários . . . 66
4.4.4.4 Geração de plugins . . . 68
4.4.4.5 Geração de áudios . . . 69
4.4.4.6 Considerações sobre o estudo de caso 1 . . . 71
4.5 Etapa 5: segundo estudo de caso — *Hacking Survivors* . . . 72
4.5.1 Definição do estudo de caso e expansão da história . . . 73
4.5.2 Desenvolvimento e gerações com IA-Gen . . . 77
4.5.2.1 Geração de cenários . . . 78
4.5.2.2 Geração de personagens e armamentos . . . 79
4.5.2.3 Geração de quadrinhos . . . 80
4.5.2.4 Considerações sobre o estudo de caso 2 . . . 81
4.6 Etapa 6: terceiro estudo de caso — *Pafúncio vs. Desenhos* . . . 82
4.6.1 Definição do estudo de caso e expansão da história . . . 83
4.6.2 Desenvolvimento e gerações com IA-Gen . . . 87
4.6.2.1 Geração de cenários . . . 87
4.6.2.2 Geração de personagens . . . 88
4.6.2.3 Geração de elementos de interface . . . 90
4.6.2.4 Considerações sobre o estudo de caso 3 . . . 90
4.7 Síntese dos estudos de caso . . . 94

5 CONSIDERAÇÕES FINAIS . . . 95
5.1 Sumarização da pesquisa e resultados . . . 95
5.2 Limitações da pesquisa . . . 96
5.3 Impacto para a área de jogos e IA-Gen . . . 98
5.4 Implicações e trabalhos futuros . . . 98

REFERÊNCIAS . . . 100

APÊNDICES 106

APÊNDICE A – REPOSITÓRIO DA PESQUISA . . . 107

APÊNDICE B – DECLARAÇÃO SOBRE O USO DE INTELIGÊNCIA ARTIFICIAL GENERATIVA . . . 109

1 Introdução

O avanço na qualidade e na popularização de ferramentas de IA-Gen tem causado uma transformação visível em diversas áreas da sociedade, incluindo a do *design* de jogos. Desde o lançamento dos grandes modelos de linguagem, *Large Language Model*s (LLMs), como o ChatGPT (OPENAI, 2025), o Gemini (GOOGLE, 2026a) e o Claude (ANTHROPIC, 2025), a indústria de jogos tem buscado integrar a IA-Gen ao processo de *design* e desenvolvimento de jogos, incluindo as etapas de criação de histórias, revisão de ideias e inserção de diálogos interativos (GALLOTTA et al., 2024).

Além dos LLMs, também começaram a se popularizar os modelos multimodais, que combinam a facilidade de interação de um modelo de linguagem com a capacidade de "entender" e gerar outros tipos de conteúdo não textual, como imagens e áudios de alta fidelidade (AL-SURAYHI et al., 2025). Assim, tarefas que antes exigiam grandes ciclos de produção e equipes especializadas, como a confecção de ativos digitais (*assets*), tanto visuais (*sprites* de personagens e objetos, artes conceituais e cenários) quanto de áudio (como trilhas sonoras), agora podem ser feitas por uma única pessoa ou por uma equipe reduzida (PANCHANADIKAR et al., 2026).

Grandes estúdios de jogos já começaram a incorporar essas tecnologias em seus *pipelines* de desenvolvimento, e a IA-Gen tornou-se tópico de intensa pesquisa e de aplicação prática para otimizar a criação de conteúdo, aprimorar a jogabilidade e democratizar o acesso à produção de jogos. Exemplificando essa tendência, a desenvolvedora Ubisoft tem avançado em pesquisas voltadas à geração de materiais técnicos para jogos 3D (UBISOFT LA FORGE, 2025; YING et al., 2025). Nesse cenário, como apontam Ternar et al. (2025), a forma como artistas, desenvolvedores e *designers* de jogos interagem com seu trabalho é alterada pela necessidade de atuar como curadores ou cocriadores junto à IA-Gen.

Esses casos indicam a tendência do mercado de jogos de que a IA-Gen não atue como uma substituta, mas sim como uma ferramenta de suporte que abrange desde a fase de ideação até a elaboração de materiais técnicos (TERNAR et al., 2025). Este trabalho parte da seguinte questão: **de que forma e com quais ferramentas a IA-Gen pode apoiar desenvolvedores independentes na produção de ativos multimídia e na documentação de jogos?** A pergunta se justifica porque os trabalhos levantados na Seção 2.4 tratam essas ferramentas de forma isolada e não estruturam uma metodologia que integre múltiplas modalidades de geração em um único fluxo de produção.

Para isso, foi realizada uma análise exploratória e prática da integração de LLMs e modelos multimodais no *design* de jogos. O procedimento adotado consiste na documentação completa do processo de desenvolvimento de três jogos elaborados como provas

de conceito. O primeiro deles, *PinkBob J. e o Enigma do Rum*, foi desenvolvido junto à disciplina Design e Desenvolvimento de Jogos. Ele foi utilizado como laboratório para testar diversas ferramentas de IA-Gen capazes de gerar os ativos digitais do jogo e, em seguida, para selecionar os modelos que seguiriam auxiliando no desenvolvimento do próprio jogo e das demais provas de conceito.

A investigação buscou, portanto, desenvolver e documentar uma metodologia de apoio ao processo de *design* de jogos na forma de um método de produção (*pipeline*) híbrido, que combina ferramentas de IA-Gen multimodais ao trabalho autoral do desenvolvedor. Os testes comparativos que fundamentaram a seleção dessas ferramentas foram conduzidos entre novembro de 2025 e janeiro de 2026, recorte que delimita os modelos e as versões comparados; a produção dos três estudos de caso, a partir de fevereiro de 2026, usou a versão seguinte do modelo de linguagem selecionado (Google Gemini 3.1 Pro).

1.1 Objetivos

O trabalho tem como objetivo desenvolver e documentar uma metodologia de apoio ao *design* e à produção de jogos digitais baseada em ferramentas de Inteligência Artificial Generativa (IA-Gen), voltada ao desenvolvimento independente de jogos, abrangendo atividades de ideação, *storytelling*, documentação técnica e a geração de ativos digitais. Fazem parte do escopo do trabalho os seguintes objetivos específicos:

- listar e categorizar os tipos de ativos digitais necessários ao desenvolvimento de um jogo digital, tais como *sprites*, animações, efeitos sonoros, músicas e enredo;
- levantar os modelos de IA-Gen mais adequados à geração de cada categoria de ativo digital;
- realizar testes comparativos para cada categoria e modelo de IA-Gen, analisando os resultados obtidos;
- selecionar os modelos e elaborar a metodologia a ser seguida na construção dos jogos;
- gerar os ativos finais, desenvolver e documentar a primeira prova de conceito, o jogo *PinkBob J. e o Enigma do Rum*;
- conceber, projetar, desenvolver e documentar a segunda prova de conceito, o jogo *Hacking Survivors*, aplicando a metodologia elaborada;
- conceber, projetar, desenvolver e documentar a terceira prova de conceito, a versão de demonstração do jogo *Pafúncio vs. Desenhos*, aplicando a metodologia já revisada a partir dos dois casos anteriores;

- verificar a aplicabilidade do método ao longo dos três casos, registrando os ajustes que cada projeto exigiu.

A condução do trabalho seguiu as etapas sequenciais detalhadas no Capítulo 3: a categorização dos tipos de ativos digitais que compõem um jogo; a pesquisa e os testes exploratórios das ferramentas de IA-Gen capazes de gerar cada categoria; o processo de ideação e seleção das propostas; e o desenvolvimento dos três estudos de caso, nos quais a metodologia foi aplicada e revisada a cada projeto. Dessas etapas resulta o método proposto, apresentado na Seção 3.1, que organiza a produção em cinco fases: concepção, planejamento da produção, seleção das ferramentas, geração dos ativos por *meta-prompting* e montagem na *game engine*, esta última a cargo do próprio desenvolvedor.

1.2 Organização do trabalho

O restante do trabalho está estruturado em quatro capítulos. O Capítulo 2 apresenta a revisão bibliográfica, abordando o processo de *design* e desenvolvimento de jogos, a importância e a elevada complexidade da criação dos componentes de um jogo, além de fundamentar como a IA-Gen tem sido utilizada para apoiar essas etapas e de reunir os trabalhos relacionados que situam a lacuna tratada aqui. O Capítulo 3 apresenta o método proposto e as etapas sequenciais que conduziram à sua construção, da categorização dos ativos digitais e dos testes exploratórios de ferramentas até os três estudos de caso. O Capítulo 4 apresenta os resultados de cada etapa: a categorização dos ativos digitais, a avaliação comparativa das ferramentas, o funil de ideação e as três provas de conceito elaboradas ao longo da pesquisa, ilustrando a aplicação prática das ferramentas de IA-Gen na concepção dos jogos propostos. Por fim, o Capítulo 5 apresenta a conclusão do trabalho, sumarizando os resultados obtidos, apontando as limitações encontradas, discutindo o impacto do trabalho para as áreas de jogos e de Inteligência Artificial Generativa e sugerindo possíveis temas para pesquisas futuras.

2 Revisão bibliográfica

Este capítulo apresenta a fundamentação teórica que sustenta a pesquisa. Inicialmente, são detalhadas as particularidades e a complexidade do processo de *design* e desenvolvimento de jogos, com foco na produção dos aspectos que não envolvem a programação direta de código. Em seguida, são introduzidos os conceitos fundamentais sobre Inteligência Artificial Generativa, modelos multimodais e engenharia de *prompts*, seguidos de uma breve análise de como a indústria de jogos aplica estas tecnologias. Por fim, são discutidos os trabalhos relacionados que investigam as aplicações da IA-Gen na criação de jogos.

2.1 O processo de *design* e desenvolvimento de jogos

O *design* e desenvolvimento de jogos é uma área criativa complexa, normalmente estruturada em três estágios fundamentais: pré-produção, produção e pós-produção. No estágio inicial, a pré-produção, os desenvolvedores concentram os seus esforços na ideação e no planejamento: são definidos o universo da narrativa, as regras de interação e a estética, e todas essas decisões são documentadas. Com esta base estabelecida, o projeto avança para o estágio de produção, que engloba a programação do jogo, a escrita da narrativa, a criação de artes e a composição sonora (músicas e efeitos). Por fim, no estágio de pós-produção, o jogo é refinado para garantir que a experiência final seja livre de problemas e bem equilibrada (CALHABEU; LIMA, 2025).

No estágio de pré-produção, o foco principal consiste na elaboração do documento de *design* do jogo, ou *Game Design Document* (GDD). Este documento formaliza todos os conceitos, as mecânicas, o estilo visual, os personagens e o universo da narrativa da obra em um único local. O material serve como referência para todos os envolvidos no projeto, funcionando como um guia prático para assegurar que a equipe esteja alinhada com as mesmas ideias (PEDERSEN, 2003).

Uma vez finalizado o GDD, dá-se início à produção propriamente dita do jogo. É neste estágio que a maior parte do tempo de *design* e de desenvolvimento é empregada. Nele, o código-fonte é programado para construir as mecânicas, a jogabilidade e as funcionalidades técnicas, em ciclos de prototipação que validam os sistemas antes de sua implementação na versão final. Em paralelo, ocorrem as produções relacionadas aos demais aspectos do jogo: a criação visual de personagens, de cenários e de objetos, além da confecção dos elementos de interface gráfica, das trilhas musicais e dos efeitos sonoros (ADAMS, 2014).

Embora a estruturação do código represente um dos pontos mais críticos de um

jogo, o processo geral de concepção vai além da programação. A imersão e a qualidade estética do produto final dependem também da produção de elementos que não são código. Segundo Adams (2014), áreas como arte, animação, música, engenharia de áudio e roteiro constituem disciplinas separadas e essenciais, exigindo papéis específicos dentro de uma equipe, tais como diretores de arte, *designers* de áudio, roteiristas e *designers* de níveis. Por consequência, a elaboração de artes conceituais, a modelagem de interface, a animação de *sprites* e a composição de trilhas sonoras originais requerem um alto nível de especialização técnica e artística.

Em estúdios comerciais de grande porte, este fluxo produtivo é conduzido por equipes multidisciplinares compostas por profissionais dedicados de forma exclusiva a cada segmento de especialidade. Em contrapartida, ao analisar o cenário do desenvolvedor independente, Adams (2014) destaca que lidar com toda a complexidade da obra de maneira solitária exige um perfil versátil e dotado de conhecimentos profundos em múltiplas áreas. A necessidade de concentrar tantas funções para produzir artes de alta fidelidade, enredos complexos e áudio imersivo atua como uma barreira técnica e orçamentária, a qual sobrecarrega os criadores solitários e frequentemente inviabiliza a conclusão de seus projetos.

Para concluir o ciclo de desenvolvimento, o jogo avança para o estágio de pós-produção. Nela, o projeto é submetido a baterias de testes práticos com o objetivo de identificar e de corrigir falhas técnicas de execução. O estágio concentra os esforços da equipe no polimento das mecânicas, no balanceamento geral do sistema e na resolução de quaisquer pendências que tenham restado do estágio de produção, assegurando que a experiência final entregue ao jogador seja coesa e funcional (CALHABEU; LIMA, 2025).

2.2 Inteligência Artificial Generativa e modelos multimodais

A IA-Gen tem sido incorporada de forma crescente à indústria de jogos (AL-SURAYHI et al., 2025; KULTIMA, 2025; SWEETSER, 2024). Historicamente, as abordagens tradicionais de Inteligência Artificial concentravam-se em tarefas discriminativas orientadas a classificações e previsões. Contudo, desenvolvimentos recentes impulsionaram uma mudança de paradigma, expandindo as capacidades computacionais para a execução de tarefas criativas complexas por meio da IA-Gen (BANH; STROBEL, 2023).

Fundamentada nos avanços do Aprendizado Profundo (*Deep Learning*), a IA-Gen utiliza modelos generativos profundos, em inglês *Deep Generative Models*. Diferentemente da Inteligência Artificial (IA) discriminativa, que busca determinar fronteiras de decisão em um conjunto de dados, esses modelos aprendem, a partir de grandes volumes de dados de treinamento, a distribuição de probabilidade desses dados para gerar novas amostras que se assemelham à sua base original (BANH; STROBEL, 2023). Essa capacidade abriu

o caminho para a criação de sistemas ainda mais versáteis, os modelos multimodais.

Os modelos multimodais caracterizam-se por combinar a geração e o processamento de texto com, pelo menos, mais uma modalidade de dados, como imagens, vídeos, áudio ou modelos 3D. A interação do usuário com esses sistemas ocorre frequentemente por meio de *prompts*, que são instruções descritivas em linguagem natural utilizadas para guiar e especificar a criação do conteúdo desejado (BANH; STROBEL, 2023).

A evolução desses sistemas multimodais, a exemplo do ChatGPT (OPENAI, 2025) e do Google Gemini (GOOGLE, 2026a), ampliou o acesso a recursos criativos, permitindo a indivíduos sem experiência artística prévia gerar narrativas e imagens a partir de descrições em linguagem natural. Além disso, ferramentas focadas em áudio, como o Suno (SUNO, 2025), permitem a criação de músicas completas com apenas uma instrução de estilo e letras. Outras plataformas recentes, como o Kling AI (KLING AI, 2025), elevam esta integração ao combinarem a geração de imagens, vídeos e efeitos sonoros em uma única ferramenta.

2.2.1 Engenharia de *prompt*

A engenharia de *prompts* (*Prompt Engineering*) tornou-se uma etapa essencial para a obtenção de respostas consistentes e de alta qualidade em LLMs. Essa prática consiste no desenvolvimento sistemático de instruções capazes de orientar o modelo na execução de tarefas específicas, considerando aspectos como clareza, contexto, estrutura e formato do *prompt*. Segundo Boonstra (2025), a elaboração de *prompts* é um processo iterativo, no qual diferentes formulações são testadas e refinadas até que se alcance o desempenho desejado, uma vez que pequenas alterações na redação ou na organização das instruções podem influenciar significativamente a qualidade das respostas geradas.

Entre as técnicas mais difundidas está o *zero-shot prompting*, no qual o modelo recebe apenas a descrição da tarefa, sem exemplos de entrada e saída, explorando o conhecimento previamente adquirido durante seu treinamento para realizar a atividade solicitada (BOONSTRA, 2025).

Além da técnica *zero-shot*, a literatura registra o refinamento iterativo de *prompts* com o auxílio dos próprios modelos de linguagem. Essa estratégia é respaldada pelo estudo de Liu et al. (2025), que demonstra que a eficácia dos modelos depende não apenas do conteúdo do *prompt*, mas também de sua estrutura e formatação. Os autores destacam que o desempenho dos LLMs pode ser significativamente aprimorado por meio do refinamento iterativo dos *prompts*, considerando simultaneamente ajustes no conteúdo e na organização das instruções, abordagem denominada *Content Format Integrated Prompt Optimization* (CFPO).

Uma variação desse refinamento transfere ao próprio modelo a redação do *prompt*.

Boonstra (2025) descreve essa prática como engenharia automática de *prompts* (*Automatic Prompt Engineering*): um modelo é instruído a escrever *prompts* para uma tarefa, e as versões obtidas são avaliadas e refinadas em ciclos sucessivos. A técnica foi proposta por Zhou et al. (2023), cujos *prompts* gerados por LLMs alcançaram, na maior parte das tarefas avaliadas, desempenho comparável ou superior ao das instruções escritas por pessoas. Na geração de imagens, Hao et al. (2023) aplicam a mesma ideia a modelos de difusão: um modelo de linguagem adapta automaticamente a descrição do usuário para o *prompt* que o modelo gerador interpreta melhor. O termo *meta-prompting* aparece na literatura com outro sentido: em Suzgun e Kalai (2024), designa um modelo de linguagem que coordena instâncias de si mesmo, cada uma encarregada de uma parte da tarefa.

2.3 Aplicações práticas da IA-Gen na indústria

A adoção da IA-Gen tem transformado profundamente os fluxos de trabalho criativos em diversas áreas. No desenvolvimento de jogos, um setor conhecido historicamente por ser pioneiro na aplicação de novas tecnologias, essa transformação é ainda mais evidente, sendo apontada por Kultima (2025) como a maior tendência contemporânea da área.

O impacto prático da IA-Gen é quantificável no mercado comercial. Segundo o levantamento de Lambe (2025), que contou os jogos com declaração de uso de IA-Gen na Steam (VALVE CORPORATION, 2026), esses jogos passaram de cerca de 1% para 7% de todo o catálogo da plataforma no intervalo de um ano, até julho de 2025, e o número de títulos com a declaração ficou quase oito vezes maior. Entre os jogos lançados em 2025 até aquele mês, a proporção chegava a quase 20%. O mesmo levantamento registra que a geração de ativos visuais, como personagens, cenários e texturas, aparece em cerca de 60% das declarações, dado que leva Ternar et al. (2025) a apontar a produção de arte como uma das principais portas de entrada da IA-Gen no desenvolvimento de jogos. Como as declarações são inconsistentes, os autores ressalvam que esses números provavelmente subestimam a escala real da adoção.

A mesma adoção traz uma questão ainda em aberto no plano jurídico: a dos direitos autorais sobre o que os modelos geram. Modelos generativos são capazes de criar obras que se assemelham fortemente a conteúdos originais, o que levanta questionamentos complexos sobre a titularidade dos direitos autorais, a qual fica em uma zona cinzenta entre o usuário, o desenvolvedor da ferramenta e a própria IA-Gen (CARNEIRO; CHAI, 2025; SANTOS; PENDIUK, 2026). Segundo Carneiro e Chai (2025), a legislação brasileira atual (Lei nº 9.610/1998) define o autor exclusivamente como uma pessoa física, criando uma lacuna jurídica evidente para conteúdos gerados por máquinas. Além disso, como os modelos são frequentemente treinados com vastas bases de dados que incluem materiais protegidos, existe um risco real de plágio inadvertido e violação de propriedade intelectual

(CARNEIRO; CHAI, 2025). Um caso recente que ilustra a controvérsia ocorreu em março de 2025, quando usuários passaram a usar o gerador de imagens do GPT-4o, recém-integrado ao ChatGPT, para recriar fotografias pessoais no estilo artístico do Studio Ghibli, gerando ampla polêmica sobre o uso não autorizado de obras protegidas para o treinamento dos modelos e a consequente geração de novas imagens (CNN BRASIL, 2025). A interseção entre a IA-Gen e os direitos autorais exige, portanto, uma investigação aprofundada, a exemplo do estudo conduzido por Santos e Pendiuk (2026), que buscam analisar o arcabouço jurídico vigente e os desafios da propriedade intelectual na produção artística contemporânea.

2.4 Trabalhos relacionados

A integração da IA-Gen no *design* e desenvolvimento de jogos tem sido objeto de crescente interesse na comunidade acadêmica. Uma revisão preliminar da literatura feita por Sweetser (2024) evidencia que os LLMs estão sendo aplicados em diversas frentes da indústria de jogos, abrangendo desde a criação de narrativas e diálogos até o desenvolvimento de agentes autônomos e ferramentas de apoio à produção. Os trabalhos discutidos a seguir foram obtidos a partir de buscas por artigos científicos usando termos relacionados combinados (por exemplo, LLM, *game design*, *asset generation*) em ferramentas genéricas (Google) e especializadas (Google Scholar, Semantic Scholar, SciELO), publicados entre 2023 e 2025. Considerando o objetivo exploratório do estudo, é importante ressaltar que não foram adotados procedimentos de maior rigor metodológico, como a condução de uma revisão sistemática ou um mapeamento sistemático da literatura.

Nesse contexto de mapeamento da área, a pesquisa conduzida por Gallotta et al. (2024) busca sumarizar as principais aplicações que os LLMs podem ter no ecossistema de jogos. Os autores apontam diferentes papéis para esses modelos: como jogador, que deve completar o jogo; como personagens não jogáveis, que interagem com o usuário; como assistentes, oferecendo tutoriais e dicas específicas adaptadas às dificuldades do jogador; como comentaristas ou narradores, a exemplo de jogos de esportes; como analistas de dados, processando e sumarizando o comportamento do jogador para pesquisas; e, por fim, como *designers* autônomos ou assistentes de desenvolvimento, utilizando ferramentas como o ChatGPT para auxiliar o profissional a conceber o jogo.

No escopo de agentes interativos e jogabilidade, pesquisas têm demonstrado o potencial dos modelos generativos para simular comportamentos complexos de personagens não jogáveis. A arquitetura proposta por Park et al. (2023) utiliza modelos de linguagem para simular interações sociais humanas realistas e de longo prazo em ambientes virtuais, permitindo que os agentes armazenem memórias e planejem ações de forma autônoma. Em cenários tridimensionais complexos, o agente autônomo VOYAGER demonstrou a

capacidade de explorar o mundo do jogo *Minecraft* (MOJANG STUDIOS, 2011) e adquirir novas habilidades continuamente por meio de IA-Gen, sem a necessidade de intervenção humana e com suporte a memória de longo prazo (WANG et al., 2023).

No campo da geração de narrativas, estudos investigam como a IA-Gen pode gerar caminhos não determinísticos e emergentes a partir da interação do jogador em tempo real. Em testes com *Dejaboom!*, um jogo de aventura textual cujos personagens não jogáveis são conduzidos pelo GPT-4, Peng et al. (2024) observaram que as interações livres dos jogadores geravam "nós emergentes" na história original. Essa abordagem proporciona narrativas colaborativas e inesperadas que servem, inclusive, como um mecanismo de *feedback* criativo e direto para os *designers* do jogo.

A pesquisa de Amresh (2023) propõe a criação de jogos educacionais utilizando ferramentas generativas. Entre os exemplos, destaca-se um jogo de perguntas e respostas sobre ciências, história e estudos sociais, que utiliza o ChatGPT para gerar as questões e soluções dinamicamente. O estudo também apresenta um jogo de química no qual o ChatGPT gera os elementos químicos em tempo real e o DALL-E cria a imagem dos compostos; assim, o jogador recebe os elementos e deve formar a equação correta para progredir. Por fim, o artigo descreve um gerador de histórias infantis, em que a interação do usuário com um botão aciona a IA-Gen para gerar uma narrativa inédita.

Em uma abordagem focada na etapa de implementação técnica, Serra et al. (2025) apresentam o Prompting GameCraft, uma ferramenta que utiliza LLMs para automatizar as fases iniciais do desenvolvimento de jogos. A arquitetura da solução recebe um GDD como entrada e o traduz automaticamente para um arquivo estruturado em JSON, denominado *Game Design File* (GDF). A partir dessa formalização semântica, o sistema infere diagramas de classe modulares, cria a estrutura de diretórios e gera o código-fonte executável do jogo, com o objetivo de otimizar a transição do *design* abstrato para a implementação em cenários de prototipagem rápida.

Apesar de alguns trabalhos combinarem pontualmente texto e imagem para gerar conteúdos em tempo real durante a *gameplay*, os trabalhos levantados não apresentam a preocupação de estruturar uma metodologia de *design* de jogos ampla que integre múltiplas modalidades de geração de conteúdo em um único *pipeline* centralizado de produção.

Este trabalho tem como objetivo desenvolver e documentar uma metodologia de apoio ao *design* e à produção de jogos digitais baseada em ferramentas de IA-Gen. Em vez de analisar as ferramentas de forma isolada, esta pesquisa apresenta o diferencial de avaliar de maneira simultânea modelos geradores de texto, imagem, áudio e vídeo, e aplicar estas tecnologias em conjunto para o *design* e o desenvolvimento de três jogos distintos como estudos de caso, com gêneros e estéticas próprios. Esta abordagem permite não apenas examinar a viabilidade do uso de IA-Gen no apoio à produção, mas também mapear os desafios reais enfrentados por um único desenvolvedor ao aplicar múltiplas ferramentas de

IA-Gen em um mesmo fluxo de produção, da concepção do jogo à geração de seus ativos. A Tabela 1 apresenta uma breve comparação entre as modalidades de geração por IA-Gen abordadas nos trabalhos mencionados.

Tabela 1 – Comparativo das áreas de aplicação de IA exploradas nos trabalhos relacionados e na presente pesquisa.

Trabalho	**TXT**	**IMG**	**MUS**	**SFX**	**VID**	**3D**	**HIS**	**TR**	**COD**
Sweetser (2024)	X					X	X	X	X
Gallotta et al. (2024)	X						X	X	X
Park et al. (2023)	X							X	
Wang et al. (2023)	X							X	X
Peng et al. (2024)	X						X	X	
Amresh (2023)	X	X					X	X	
Serra et al. (2025)	X								X
Ying et al. (2025)		X				X			X
Este trabalho	X	X	X	X	*		X		*

Legenda: TXT: Texto; **IMG**: Imagens; **MUS**: Músicas; **SFX**: Efeitos Sonoros; **VID**: Vídeos; **3D**: Modelos 3D; **HIS**: História e Narrativa; **TR**: Aplicações em Tempo Real; **COD**: Desenvolvimento de Código. (*): parcialmente explorado (vídeos gerados apenas nos testes exploratórios, sem uso na produção dos jogos; código gerado de forma pontual, sem avaliação), conforme detalhado nos Capítulos 3 e 4.
Fonte: Elaborada pelo autor (2026).

O panorama reunido nesta seção mostra que os trabalhos levantados cobrem aplicações isoladas da IA-Gen no desenvolvimento de jogos, mas não descrevem um fluxo que as articule da concepção do jogo à geração de seus ativos. O Capítulo 3 apresenta essa articulação na forma do método proposto e relata as etapas que conduziram à sua construção, com os critérios usados para avaliar as ferramentas em cada modalidade de geração.

3 Metodologia

Este capítulo apresenta a metodologia, as ferramentas e o fluxo de trabalho adotados para a estruturação do método de produção (*pipeline*) de IA-Gen e a implementação dos artefatos práticos. A pesquisa adotou uma abordagem exploratória, organizada em etapas sequenciais com refinamento iterativo dentro de cada uma. A Seção 3.1 apresenta o método proposto, que é o produto dessa pesquisa; as Seções 3.2 a 3.7 relatam as seis etapas que o construíram.

Cabe ressaltar que as etapas iniciais desta metodologia ocorreram em conjunto com a disciplina Design e Desenvolvimento de Jogos (2025/2). O processo simplificado adotado na disciplina é constituído das seguintes etapas: ideação, concepção (*design* do jogo), prototipação, desenvolvimento e testes. Esse processo foi informalmente adotado na produção dos jogos utilizados como prova de conceito. Ele é distinto do método proposto na Seção 3.1: as etapas da disciplina auxiliaram a produção dos jogos, mas diferem em relação aos objetivos e fases propostos neste trabalho.

3.1 O método proposto

Esta seção apresenta o método de apoio proposto para a produção independente de jogos: a partir de uma ideia e do acesso às ferramentas de IA-Gen, o que fazer e em que ordem. As seções seguintes relatam como ele foi construído e aplicado na produção de três jogos como provas de conceito; os resultados estão no Capítulo 4.

O método pressupõe um único desenvolvedor, ou uma equipe reduzida, produzindo um jogo 2D de escopo curto, e organiza a produção em cinco fases, com refinamento iterativo dentro de cada uma. Neste trabalho, ele foi construído e aplicado por um único desenvolvedor; o uso por equipes reduzidas não foi verificado. Considera-se que o método foi bem executado quando o jogo foi concebido, documentado e produzido com os ativos gerados pelas ferramentas de IA-Gen selecionadas, e os ajustes que o projeto exigiu ficaram registrados.

Fase 1: concepção. Parta de uma ideia curta, de um parágrafo, que declare a mecânica central ou a premissa narrativa. Expanda a ideia sobre os três pilares do *design* de jogos (ADAMS, 2014) (*gameplay*, *storytelling* e experiência do usuário), definindo aspectos como perspectiva, câmera, plataforma, mecânicas, arco dramático e resposta emocional pretendida. Elabore então o documento de *design* de página única e, a partir dele, a *story spine*. Antes de pedir a *story spine*, desenhe. Faça rascunhos de baixa fidelidade dos elementos principais e das telas e anexe-os ao *prompt*. No segundo e no terceiro estudos de

caso, os rascunhos entraram como insumo da narrativa, e não como consequência dela. Recomenda-se essa ordem porque o texto gerado já nasce compatível com o que o jogo mostrará na tela. Os comandos aplicados para a expansão da ideia, o documento de *design* e a *story spine* estão exemplificados na Seção 4.2.1.

Fase 2: planejamento da produção. Do documento de *design* e da *story spine*, derive um diagrama das fases do jogo que liste, na ordem em que aparecerão, os cenários, os itens e os personagens com os quais o jogador poderá interagir. Esse diagrama guia as gerações: cada fase do jogo é implementada individualmente e os ativos são gerados sob demanda, conforme a necessidade de cada momento do projeto. A geração sob demanda evita produzir ativos (*assets*) para fases que o projeto ainda pode descartar, e mantém a consistência visual, porque cada novo pedido é feito com os *assets* já aprovados como referência.

Fase 3: seleção das ferramentas. Para cada categoria de *asset* que o diagrama da Fase 2 exige, escolha uma ferramenta de IA-Gen como geradora principal. A escolha dos modelos de linguagem vem antes de todas as outras, porque é com eles que se faz a Fase 1. O procedimento de comparação adotado neste trabalho está descrito na Seção 3.3. Adote, em cada categoria, a ferramenta mais bem avaliada; quando o desempenho for equivalente, ou quando a vantagem vier de uma característica contornável, decida pelas condições de acesso (limites do plano gratuito e custo) e pelos recursos de edição da plataforma, como se fez na escolha do Flow Studio para imagens (Seção 4.2.2). Dada a rápida evolução dos modelos, recomenda-se refazer a análise com os lançamentos mais recentes. Caso não queira repetir a comparação, é possível adotar diretamente as ferramentas selecionadas neste trabalho, com a ressalva de que o recorte é de novembro de 2025 a janeiro de 2026: Claude 4.5 Sonnet para a *story spine*, Google Gemini 3 Pro para a expansão da ideia e o documento de *design*, Flow Studio Nano Banana Pro para imagens, Suno para trilhas sonoras e Kling AI para efeitos sonoros, nas versões mais recentes de cada uma.

Fase 4: geração dos ativos. Todas as gerações seguem o mesmo fluxo, aqui denominado *meta-prompting*: em vez de solicitar o ativo diretamente à ferramenta de destino, instrua um modelo de linguagem a atuar como engenheiro de *prompts*, fornecendo a ele a concepção narrativa, o documento de *design* e os ativos já aprovados como referência. Inclua também, quando houver, imagens ou vídeos que sirvam de referência artística. O comando refinado e detalhado que ele devolve é o que se submete à ferramenta geradora. Essa abordagem foi adotada para que cada pedido à ferramenta geradora carregue o contexto completo do projeto (a narrativa, o documento de *design* e os ativos já aprovados) sem que o desenvolvedor precise redigir à mão um *prompt* longo e específico de cada ferramenta. É a engenharia automática de *prompts* descrita na Seção 2.2.1, aplicada entre dois modelos: o de linguagem escreve o *prompt*, e a ferramenta geradora o executa. Por categoria, segue a recomendação pela experiência prática obtida nos três estudos de caso:

- Personagens. Gere primeiro uma imagem base, de corpo inteiro, a partir da descrição na *story spine*. Reintroduza essa imagem no modelo de linguagem junto com o pedido da folha de animação, especificando o número de quadros, a progressão do movimento, o espaçamento uniforme e o fundo limpo, e descrevendo o traje peça a peça para travar a identidade do personagem entre os quadros (ver exemplo aplicado na Seção 4.4.4.1). As animações geradas para os jogos produzidos usaram três quadros por estado.

- Objetos. Use a imagem base do protagonista como referência de estilo e acrescente a descrição do item. É o que mantém a paleta e o traço coerentes sem precisar descrever tudo novamente a cada pedido.

- Cenários. Anexe o rascunho de baixa fidelidade ao comando, junto com a *story spine*, e descreva o ambiente elemento a elemento (ver exemplo aplicado na Seção 4.4.4.3). Para jogos montados com *tiles*, não peça o cenário inteiro: peça os elementos isolados e monte-os na *game engine* escolhida. No segundo estudo de caso, a tentativa de gerar o cenário completo falhou porque as extremidades das peças não encaixavam entre si.

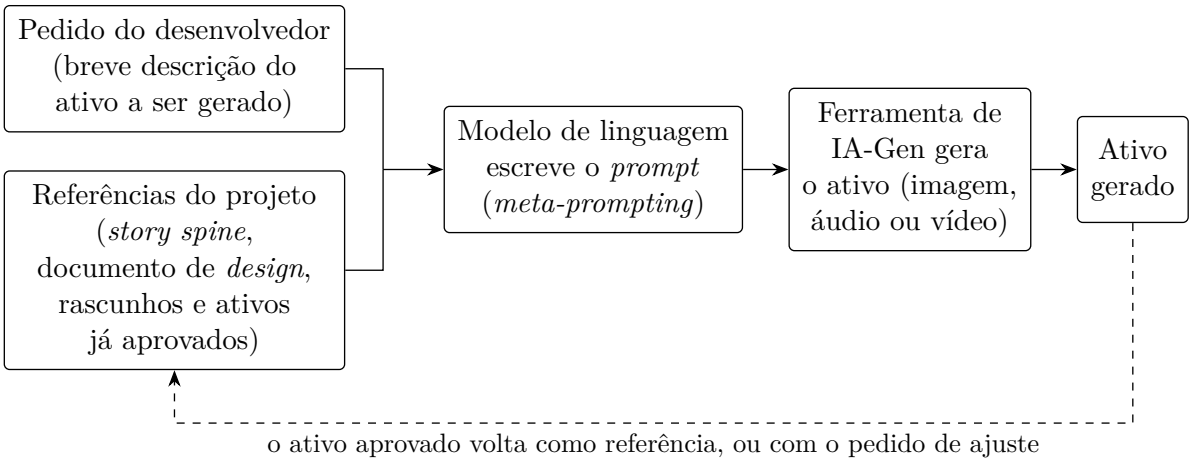
- Imagens para a *game engine*. Salve cada imagem em formato *.jpg*. Como o comando pede fundo limpo, chegue ao fundo transparente utilizando uma ferramenta removedora de fundo, como o *Microsoft Paint*, selecionando toda a área a ser limpa e salve a imagem em formato *.png*. Para a montagem (Fase 5), recorte as folhas de animação quadro a quadro com a ferramenta *Sprite Editor* da *Unity*. Se a resolução nativa não bastar, amplie a imagem antes de importá-la, com a ferramenta de ampliação integrada do Flow Studio.

- Áudio. Derive o tom da *story spine* e peça ao modelo de linguagem que escreva os comandos de trilha e de efeitos. Para efeitos, exija som claro, sem cortes abruptos e capaz de rodar em repetição contínua.

A padronização desse fluxo é o que favorece a reprodutibilidade do processo: o mesmo procedimento vale para as imagens e os áudios produzidos nos três jogos e se apoia na utilização integrada de diferentes ferramentas de IA-Gen, conforme ilustrado na Figura 1.

Variante agêntica. Quando a ferramenta de geração opera dentro do repositório do projeto, o fluxo é encurtado: reúna o documento de *design*, os rascunhos e a *story spine* em um diretório único e acessível ao agente. O pedido passa a ser feito em um só lugar: o agente lê o contexto, gera o ativo e o salva no diretório correspondente, restando apenas a integração com a *game engine*, como ilustra a Figura 2. Essa prática começou no segundo

Figura 1 – Fluxo padrão de geração de ativos multimídia.

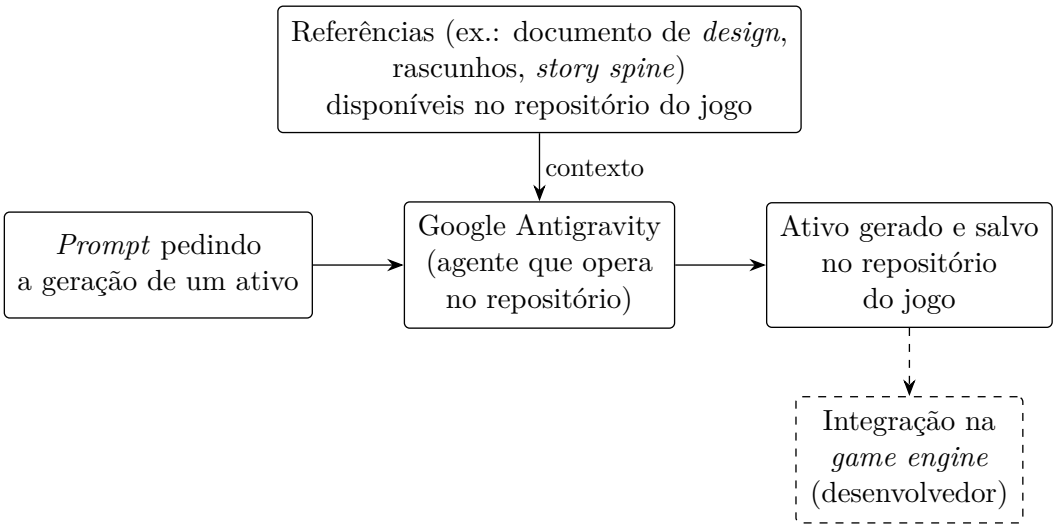


Fonte: Elaborada pelo autor (2026).

estudo de caso, a partir da quarta fase do jogo, e foi adotada integralmente no terceiro estudo de caso, com o Google Antigravity (GOOGLE, 2026b).

Fase 5: montagem e integração na *game engine*. A montagem do jogo é trabalho autoral do desenvolvedor: importar os ativos, recortar as folhas de animação, construir os cenários a partir das peças geradas, programar as mecânicas e integrar tudo na *engine*. Neste trabalho a IA-Gen não atuou como geradora do jogo, e sim de forma pontual e como ferramenta. Por exemplo, no primeiro estudo de caso um modelo agêntico conectado ao ambiente de desenvolvimento Unity ajudou a produzir dois *plugins* de editor (um de diálogos e um de *cutscenes*) que não estavam previstos no método e foram criados sob demanda.

Figura 2 – Fluxo de geração automatizada com modelo agêntico.



Fonte: Elaborada pelo autor (2026).

Por fim, vale destacar que a geração de código-fonte não fez parte do foco desta

pesquisa: os resultados obtidos nas etapas de programação não foram catalogados nem avaliados, o que delimita o escopo do trabalho. Isso não impede o seu emprego: a geração de código-fonte por IA-Gen pode auxiliar, por exemplo, em tarefas de programação e de testes.

3.2 Etapa 1: categorização de ativos digitais em jogos

A primeira etapa consistiu em definir o que poderia ser gerado por IA-Gen dentro do contexto de *design* e desenvolvimento de jogos. Esta categorização teve por base os formatos suportados pelos modelos generativos disponíveis na data da elaboração do trabalho e estruturou o escopo da pesquisa nas seguintes categorias: texto, imagem, áudio, vídeo, *Graphics Interchange Format* (GIF) e modelo 3D.

Embora tenham sido identificados como possíveis *assets* em jogos digitais, os modelos 3D e as animações em formato GIF não foram incluídos no escopo dos testes de geração para escolha dos modelos, pela maior exigência de poder computacional para executar modelos locais e pela limitação de tempo para realização do trabalho. Já a categoria de vídeo foi abordada nos testes exploratórios, mas não seguiu para a etapa de produção dos jogos principalmente pela instabilidade na permanência de objetos e custo por geração.

3.3 Etapa 2: pesquisa e testes exploratórios de ferramentas

A segunda etapa envolveu a pesquisa e os testes exploratórios de ferramentas de IA-Gen capazes de gerar os ativos mapeados na Etapa 1. Foi construída uma planilha eletrônica para o registro dos comandos de entrada (*prompts*), dos modelos testados e dos resultados obtidos, disponível no repositório indicado no Apêndice A.

Para a análise dos ativos gerados nos testes exploratórios, foram estipulados critérios específicos para cada categoria: comparação qualitativa, sem escala nem pesos, para texto e áudio, e avaliação qualitativa estruturada, com pontuação ponderada, para imagens e vídeos:

Texto: A análise dos *assets* textuais foi estruturada em três passos sequenciais. O primeiro passo englobou a expansão de uma ideia conceitual fundamentada nos pilares de mecânicas de jogo (*gameplay*), narrativa (*storytelling*) e experiência do usuário (*user experience*), os três componentes de projeto que Adams (2014) trata em capítulos próprios. O segundo passo abrangeu a elaboração de um documento de *design* de jogo de página única. Por fim, o terceiro passo consistiu na criação da *story spine*, a estrutura narrativa base proposta por Adams (2007). Para esta etapa de concepção narrativa, foi conduzida uma comparação qualitativa entre diferentes LLMs, com o mesmo comando de entrada

submetido a todos, sem ajuste por modelo. O critério de comparação adotado priorizou os modelos que seguiram a premissa original de forma fiel e criativa, sem apresentar desvios conceituais excessivos. A validação do método consistiu em comparar os textos gerados pelas ferramentas de IA-Gen com um material de controle correspondente, redigido manualmente pelo autor do trabalho. Os textos não receberam escala nem pesos porque a comparação tomou esse material como referência: observou-se como cada narrativa gerada divergia da história de controle, considerando que todas partiram da mesma premissa conceitual.

Imagens: As amostras visuais foram submetidas a uma comparação estruturada em uma escala Likert de cinco pontos (sendo 1 para "Muito Ruim" e 5 para "Muito Bom"). A opção por avaliação humana, em vez de métricas automatizadas, apoia-se na literatura: a percepção humana continua sendo fundamental para a avaliação de gerações visuais, uma vez que métricas automatizadas apresentam limitações para capturar falhas semânticas complexas (HARTWIG et al., 2025). Cabe ressaltar que a atribuição das notas foi realizada exclusivamente pelo autor deste trabalho, com conhecimento da ferramenta de origem de cada amostra gerada e sem replicação das avaliações. Por se tratar de um processo de análise qualitativa humana individual, a pesquisa registra formalmente a presença de um viés comparativo e subjetivo.

Para a composição final da nota, o modelo de análise adotou pesos matemáticos específicos, detalhados na Tabela 2. A atribuição de pesos maiores (peso 3) para métricas como o alinhamento com a diretriz de estilo e a ausência de alucinações reflete a importância desses atributos para a utilidade do ativo. Deficiências de resolução nativa (peso 1) são contornáveis via algoritmos de escalonamento (*upscaling*), enquanto artefatos estruturais ou desvios narrativos degradam a coerência visual de maneira irreversível, caracterizando um desafio nos modelos generativos modernos (ELASRI et al., 2022). Os pesos foram mantidos desiguais para que especificações de menor impacto não compensassem graves falhas estéticas no ranqueamento das ferramentas mais aptas. A nota final de cada ferramenta é a soma, em todas as métricas, da nota atribuída multiplicada pelo peso, com as métricas binárias valendo 1 ponto quando presentes, expressa em porcentagem do máximo possível: 71 pontos para imagens e 106 para vídeos. As métricas incluem, além da qualidade do ativo gerado, características das próprias ferramentas, porque a avaliação serviu para escolher as ferramentas do fluxo de produção: o tamanho suportado do comando textual, nas imagens e nos vídeos, e a utilidade prática (acesso via plataforma ou *Application Programming Interface* (API)), nos vídeos, avaliados uma vez por ferramenta. Essas métricas respondem por 10 dos 71 pontos possíveis nas imagens e por 25 dos 106 nos vídeos.

Áudio (Trilhas Sonoras): Foi adotada uma abordagem de análise qualitativa subjetiva. As composições musicais foram analisadas sob parâmetros qualitativos de ruído, continuidade e clareza, com a observação da naturalidade vocal (quando presente) e da

Tabela 2 – Métricas e pesos para análise qualitativa de imagens geradas por IA-Gen.

Métrica	**Tipo de Análise**	**Peso**
Resolução nativa da imagem	Escala Likert	1
Qualidade visual (nitidez e ausência de borrões)	Escala Likert	2
Preservação do enquadramento delimitado	Escala Likert	1
Alinhamento com a diretriz de estilo proposta	Escala Likert	3
Tamanho suportado do comando textual	Escala Likert	2
Presença de texto na imagem	Sim ou Não	1
Legibilidade textual (caso aplicável)	Escala Likert	2
Ausência de alucinações visuais involuntárias	Escala Likert	3

Fonte: Elaborada pelo autor (2026).

aderência ao estilo solicitado.

Áudio (Efeitos Sonoros): De forma análoga às trilhas, os efeitos sonoros, *Sound Effects* (SFX), foram analisados quanto à presença de ruídos ou cortes abruptos, à clareza sonora e à coerência com o comando solicitado. As trilhas e os efeitos sonoros não receberam escala nem pesos, e a avaliação se fundamentou na percepção de cada música e de cada efeito gerados, por dois motivos. As plataformas testadas apresentam níveis de controle e funcionalidades muito distintos, como a adição de vocais a uma música depois da geração instrumental, que só algumas oferecem. Além disso, as restrições dos planos gratuitos impediram um número uniforme de gerações por ferramenta: na ElevenLabs, os créditos de uso esgotaram-se depois de uma única música.

Vídeo: Inicialmente, os vídeos foram gerados e analisados sob a mesma escala Likert de cinco pontos, sendo também classificados exclusivamente pelo autor da pesquisa, implicando o mesmo viés avaliativo subjetivo reportado anteriormente. Os critérios combinaram as métricas visuais e sonoras com parâmetros específicos de movimento, tais como estabilidade temporal (fluidez do movimento e ausência de cintilação), permanência de objetos e sincronia audiovisual, de acordo com a Tabela 3.

O critério para os pesos adotados segue a mesma lógica da categoria de imagens. Parâmetros críticos para a coesão das animações, como a estabilidade temporal e a manutenção geométrica das formas, receberam carga máxima (peso 3), pois a falta de integridade ao longo dos quadros anula o valor prático da animação (ELASRI et al., 2022). Os pesos foram mantidos desiguais para que a presença de áudio e a resolução não compensassem falhas de renderização no ranqueamento. Foi justamente a análise das pontuações obtidas nesta etapa prática que fundamentou a decisão de não levar a categoria de vídeo à etapa de produção.

Por fim, a documentação final incluiu o levantamento dos direitos de uso comercial dos ativos gerados, os limites de volume de geração impostos pelos planos gratuitos e pagos, as restrições de tamanho para os *prompts* e as resoluções máximas suportadas

Tabela 3 – Métricas e pesos para análise qualitativa de vídeos gerados por IA-Gen.

Métrica	**Tipo de Análise**	**Peso**
Presença de áudio nativo	Sim ou Não	1
Sincronia e qualidade sonora	Escala Likert	2
Resolução visual nativa	Escala Likert	1
Qualidade visual e nitidez	Escala Likert	2
Permanência de objetos (manutenção de formas)	Escala Likert	3
Preservação do enquadramento delimitado	Escala Likert	1
Estabilidade temporal (movimento fluido e contínuo)	Escala Likert	3
Alinhamento com a diretriz de estilo proposta	Escala Likert	3
Utilidade prática (acesso via plataforma ou API)	Escala Likert	3
Capacidade de reprodução contínua (*loop*)	Escala Likert	1
Tamanho suportado do comando textual	Escala Likert	2

Fonte: Elaborada pelo autor (2026).

nativamente por cada ferramenta mapeada.

3.4 Etapa 3: processo de ideação e seleção

A terceira etapa concentrou-se na ideação dos projetos, por meio de um funil criativo progressivo. O autor gerou, por *brainstorming* (OSBORN, 1953), uma listagem espontânea de cem ideias curtas de jogos. Na sequência, dez foram selecionadas por afinidade temática e viabilidade de escopo e expandidas com base em três pilares do *design* de jogos:

- ***Gameplay*:** define a perspectiva espacial (2D ou 3D), o posicionamento da câmera (fixa ou móvel), a plataforma-alvo e as mecânicas básicas, estabelecendo as regras e interações fundamentais do jogo;
- ***Storytelling*:** apresenta a premissa narrativa e o arco dramático nos quais a experiência do jogador se baseará;
- ***User Experience* (UX):** delimita a resposta emocional pretendida, especificando as sensações que o jogo busca evocar no usuário e como isso será feito.

Dentre as dez, cinco propostas foram escolhidas pelo que a expansão revelou do escopo de cada uma (viabilidade de produção no prazo da pesquisa e variedade de ativos a gerar) e submetidas a dinâmicas de *brainstorming* estruturado com a participação de cinco voluntários (estudantes do curso de Engenharia de Computação). Na primeira, três voluntários leram as cinco propostas e listaram a maior quantidade possível de palavras associadas a cada uma delas, em dois minutos por proposta. Cada um escreveu a sua lista no chat da conversa pela plataforma Discord. A partir dessas listas, as palavras foram sendo conectadas pelas suas afinidades semânticas gerando árvores de ideias. Na segunda,

essas árvores foram apresentadas a dois outros voluntários, a quem se pediu uma opinião livre sobre a relação entre a ideia original e as árvores elaboradas para refinamento da ideia; as opiniões foram registradas em um bloco de anotações do autor durante a conversa e analisadas qualitativamente. Os resultados obtidos contribuíram para fechar a versão expandida das cinco propostas, com base na qual se confirmou o primeiro estudo de caso, pela maior afinidade temática.

Os voluntários eram colegas do autor na disciplina Design e Desenvolvimento de Jogos, convidados pessoalmente, e a participação não foi remunerada. Antes das dinâmicas, foram informados do objetivo da pesquisa e de que as palavras e as opiniões registradas seriam usadas neste trabalho, e concordaram verbalmente em participar. Nenhum dado que permita identificá-los foi registrado: as dinâmicas coletaram apenas as listas de palavras e as opiniões sobre as ideias. Por se tratar de consulta pontual, em que os participantes expressaram o sentido que atribuíam a propostas de jogos e a sua avaliação sobre elas, sem possibilidade de identificação, a atividade enquadra-se no que a Resolução nº 510/2016 do Conselho Nacional de Saúde denomina pesquisa de opinião pública com participantes não identificados, que dispensa o registro e a avaliação por comitê de ética em pesquisa (BRASIL, 2016).

3.5 Etapa 4: primeiro estudo de caso — *PinkBob J. e o Enigma do Rum*

A quarta etapa consistiu no desenvolvimento do primeiro estudo de caso, *PinkBob J. e o Enigma do Rum*, um jogo de aventura 2D com mecânica focada na exploração interativa do tipo apontar e clicar (*point and click*).

A implementação deste projeto teve início com a elaboração de um documento técnico, conhecido como documento de *design* do jogo (GDD), a definição da *story spine* e a criação de protótipos de baixa fidelidade para verificar o encadeamento entre as fases. É importante salientar que ocorreu uma seleção das ferramentas levantadas na Etapa 2, empregando, em cada categoria, a ferramenta mais bem avaliada, com as exceções justificadas na Seção 4.2 por condições de acesso e recursos de edição.

O motor gráfico Unity (UNITY TECHNOLOGIES, 2026), na versão 6000.3.8f1, foi utilizado para a construção do ambiente interativo. As gerações executadas localmente utilizaram o computador descrito na Tabela 8. As particularidades deste desenvolvimento envolveram o uso de IA-Gen não apenas para a produção dos ativos digitais do jogo, mas também para o refinamento dos *prompts* utilizados na respectiva geração. Embora este trabalho não implemente integralmente o método CFPO proposto por Liu et al. (2025), adota o mesmo princípio de aperfeiçoamento contínuo, realizando sucessivas revisões dos *prompts* com auxílio dos próprios modelos de linguagem até obter resultados mais

consistentes e adequados às necessidades do processo de desenvolvimento. Dessa forma, a combinação entre o *meta-prompting* e o refinamento iterativo de *prompts* permitiu explorar o potencial dos LLMs na geração de conteúdo utilizado ao longo desta pesquisa.

O processo de desenvolvimento do jogo seguiu uma abordagem sequencial que foi replicada para os demais artefatos. Após a elaboração do GDD e com a *story spine* do jogo definida, foi elaborado um diagrama das fases do jogo listando, na ordem em que apareceriam, os cenários, os itens e os personagens com os quais o jogador poderia interagir. Em seguida, cada fase do jogo foi implementada de forma individual no projeto, executando a geração de novos ativos de forma dinâmica, conforme a demanda pontual de cada segmento do planejamento.

3.6 Etapa 5: segundo estudo de caso — *Hacking Survivors*

A quinta etapa abordou o desenvolvimento do segundo estudo de caso, visando iniciar a verificação da transferibilidade do método. Intitulado *Hacking Survivors*, o jogo pertence ao gênero de sobrevivência em arena (*survivors-like*) com elementos de progressão entre partidas (*roguelite*), utilizando visão superior (*top down*) e ambientação temática de ficção científica urbana.

Embora não fizesse parte do grupo de cinco jogos refinados inicialmente, esta proposta constava na lista original de cem ideias. A escolha foi motivada pelo objetivo de introduzir gêneros diversificados, visando expandir a variedade de ativos digitais (*assets*) que poderiam ser gerados conforme a estrutura do jogo.

A concepção seguiu o fluxo estabelecido: expansão da proposta inicial utilizando modelos generativos atuando como parceiros de *design* e formulação das árvores conceituais, desta vez sem a participação de voluntários e com o apoio dos próprios modelos generativos. Seguiram-se a elaboração do GDD e a estruturação da *story spine* no modelo clássico de jornada sequencial (baseada nas premissas “era uma vez, todo dia, mas um dia, por causa disso, até que, desde então”). Por fim, foram listadas as mecânicas, objetos e inimigos para cada fase do jogo, e os *assets* foram gerados sob demanda: *sprites* do jogador, inimigos, objetos, cenários e armamentos.

Na implementação técnica deste jogo, a pesquisa combinou o uso dos modelos selecionados anteriormente com uma seleção direcionada de modelos agênticos operando dentro do repositório do projeto. Tal abordagem permitiu a manipulação de arquivos diretamente no computador local de desenvolvimento.

3.7 Etapa 6: terceiro estudo de caso — *Pafúncio vs. Desenhos*

Por fim, a sexta etapa englobou o desenvolvimento do terceiro estudo de caso, *Pafúncio vs. Desenhos*. O projeto consiste em um título de plataforma 2D de rolagem lateral orientado à resolução de desafios, adotando uma estética visual que simula rascunhos de desenhos no papel.

De maneira análoga ao jogo descrito na Etapa 5, esta proposta não fazia parte do grupo dos cinco jogos selecionados no recorte original; contudo, estava presente na listagem original de cem ideias. A sua escolha ocorreu com o propósito de verificar a aplicação da metodologia em um gênero distinto dos anteriores.

A particularidade do fluxo de trabalho neste artefato foi a consolidação plena da ferramenta Google Antigravity (GOOGLE, 2026b). O desenvolvimento operou de forma centralizada em um único ambiente, o qual interagia diretamente com o computador local. Esta abordagem integrou, em um único ecossistema, o refinamento textual de comandos e a confecção de *assets* visuais. A montagem final no motor gráfico reuniu os ativos gerados pelos modelos selecionados nas etapas anteriores. As seis etapas descritas neste capítulo conduziram da categorização dos ativos à produção dos três artefatos, e delas resultou o método apresentado na Seção 3.1. Em relação aos estudos de caso desenvolvidos como provas de conceito: a geração de vídeo, GIF e modelos 3D ficou fora da produção dos jogos; a geração de código ficou fora da avaliação; a etapa de testes do jogo prevista no processo da disciplina ficou fora do método proposto. O Capítulo 4 apresenta as provas de conceito resultantes, os ativos gerados em cada uma e a avaliação das ferramentas empregadas.

4 Resultados

Este capítulo apresenta os resultados obtidos na pesquisa a partir da aplicação prática da metodologia. Os dados estão estruturados de acordo com cada uma das etapas definidas no Capítulo 3. O escopo das análises engloba a documentação e teste das ferramentas de IA-Gen, a concepção e o *design* dos jogos, a elaboração dos documentos técnicos, a geração dos ativos digitais (*assets*) finais, além da prototipagem e da versão consolidada dos jogos desenvolvidos.

4.1 Etapa 1: categorização de ativos digitais em jogos

O exercício de categorização foi realizado com foco na análise dos tipos de conteúdos necessários para o *design* e o desenvolvimento de jogos. O resultado desta etapa consistiu no mapeamento detalhado das categorias e dos tipos de conteúdos que poderiam ser concebidos utilizando ferramentas de IA-Gen disponíveis publicamente. Todo o levantamento foi catalogado e organizado em uma planilha Google Sheets, que se encontra disponível no repositório indicado no Apêndice A.

A estruturação final dos ativos digitais (*assets*) passíveis de geração e as suas subcategorias são listadas a seguir:

- **Texto:** documentação técnica do jogo (GDD), história e mecânicas de jogo;
- **Imagem:** *sprites* de personagens (incluindo quadros múltiplos para a geração de animações), *sprites* de itens e objetos, cenários, texturas, artes de menu (capa e ícone do jogo) e elementos de interface gráfica (botões e ícones);
- **Áudio:** trilha sonora e efeitos sonoros (SFX);
- **Vídeo:** *cutscenes*, *trailers* e animações cinematográficas;
- **GIF:** animações;
- **Modelo 3D:** personagens em diferentes perspectivas, itens e objetos tridimensionais.

4.2 Etapa 2: pesquisa e testes exploratórios de ferramentas

Esta etapa ocorreu em conjunto com a disciplina Design e Desenvolvimento de Jogos. Por esse motivo, já existia o conceito de que o jogo possivelmente eleito como estudo de caso seria *PinkBob J. e o Enigma do Rum*. Com essa premissa estabelecida, foram

realizados os testes preliminares de geração para cada tipo de conteúdo que poderia ser necessário ao desenvolvimento do projeto. Os resultados obtidos para cada categoria são discutidos a seguir.

4.2.1 Pesquisa e testes com modelos textuais

A pesquisa foi conduzida de forma exploratória. Em especial, a etapa de geração de texto foi avaliada de forma comparativa, contrastando a estrutura narrativa base (*story spine*) confeccionada manualmente pelo autor (como artefato de entrega da referida disciplina) com o resultado obtido mediante a utilização de modelos de IA-Gen. Os modelos selecionados para este teste foram os principais LLMs disponíveis publicamente durante o período compreendido entre dezembro de 2025 e janeiro de 2026, sendo eles: Claude 4.5 Sonnet (ANTHROPIC, 2025), ChatGPT 5.2 (OPENAI, 2025) e Google Gemini 3 Pro (GOOGLE, 2026a). A Tabela 4 apresenta o compilado das especificações de cada modelo mapeado, detalhando os limites e as funcionalidades do plano gratuito, os recursos do plano pago, os direitos comerciais e outras observações relevantes.

Tabela 4 – Comparativo das ferramentas de IA-Gen para geração de texto (dezembro de 2025 a janeiro de 2026).

Ferramenta e Operação	**Limites e Funcionalidades (Plano Gratuito)**	**Limites e Funcionalidades (Plano Pago)**	**Direitos Comerciais e Observações**
Claude 4.5 Sonnet (Nuvem) Limite de 200 mil *tokens*.	Sessões de cinco horas (sem limite quantitativo fixo, variando conforme a demanda dos servidores). Acesso aos modelos Sonnet e Haiku. Permite escrita, análise, *upload* de imagens, geração de código e busca na *web*.	Custo de R$ 110,00 mensais. Oferece cinco vezes mais capacidade de uso. Projetos ilimitados, raciocínio estendido, integração com e-mail e documentos, além de acesso avançado via Claude Code.	Direitos autorais do usuário. Uso comercial liberado. A plataforma utiliza dados para treinamento, salvo quando a configuração de privacidade for ativada pelo usuário.
ChatGPT 5.2 (Nuvem) Limite de 128 mil *tokens*.	Limite de cerca de dez mensagens por janela de cinco horas. Permite escrita, edição, análise de texto e imagens, geração de código e busca integrada na *web*.	Custo de R$ 39,99 mensais. Limite ampliado para cerca de cem mensagens a cada janela de cinco horas. Mesmas funcionalidades base do plano gratuito.	Direitos autorais totais do usuário em ambos os planos. Uso comercial permitido. A empresa não reivindica direitos sobre o conteúdo gerado.
Google Gemini 3 Pro (Nuvem) Limite de um milhão de *tokens*.	Acesso a modelos otimizados (como o Gemini Flash). Permite escrita, análise, edição de texto, geração de código e busca integrada.	Custo de R$ 24,99 mensais. Oferta de 200 créditos. Acesso a modelos avançados de vídeo (Veo 3.1) e de imagem (Nano Banana Pro). Conexão nativa com os aplicativos do ecossistema da empresa.	Direitos autorais retidos pelo usuário (condicionado à aceitação dos termos de serviço). Uso comercial liberado em ambos os planos.

Fonte: Elaborada pelo autor (2026).

A partir da premissa narrativa de um jogo de aventura e comédia focado na temática de piratas, foi elaborada uma ideia preliminar (detalhada nos *prompts* discutidos no próximo tópico). Esta ideia foi expandida de forma manual pelo autor para uma

estrutura narrativa inicial (*story spine*). O enredo acompanha um pirata aficionado por rum que, após um episódio de embriaguez extrema, esquece os eventos da noite anterior e desperta munido apenas de um pedaço de papel com anotações confusas, as quais o conduzem a uma grande aventura. O documento integral encontra-se no repositório indicado no Apêndice A.

Utilizando esta mesma premissa narrativa aliada a um comando textual estruturado, a pesquisa solicitou às três ferramentas de IA-Gen a execução de três passos sequenciais (com os mesmos *prompts*). O primeiro passo consistiu na expansão da ideia fundamentada nos pilares *gameplay* (mecânicas de jogo), *storytelling* (narrativa) e *user experience* (experiência do usuário), utilizando o seguinte comando:

Prompt: Refinamento de Ideação

Pensando na etapa de ideação de um jogo ***point and click*** de aventura e comédia, faça o refinamento da ideação do jogo "PinkBob J. e o Enigma do Rum". Inclua três tópicos principais:

- ***Gameplay***: O que o jogador faz? Quais são as restrições para o jogador?
- ***Storytelling***: Onde o jogo ocorre? Quais as características do ambiente? Quais os personagens e como eles se relacionam?
- ***User Experience***: Como o jogo é apresentado e percebido? Como se dá a interação do jogador? Quais emoções o jogo está tentando provocar no jogador?

O jogo tem base principalmente em outros títulos como "The Secret Of Monkey Island" da Lucasfilm Games (LUCASFILM GAMES, 1990) e outras obras de ***point and click*** da mesma época. O principal tema do jogo engloba piratas e uma sátira com o mundo antigo e atual, mesclando elementos da modernidade com a antiguidade. Além disso, apresenta uma paródia com a história clássica de João e o Pé de Feijão. Na referida adaptação, o protagonista PinkBob escuta do próprio João uma história absurda sobre um pé de feijão que levava a uma terra onde havia um gigante, galinhas que davam ovos de ouro e uma vaca que produzia rum no lugar de leite. PinkBob, sendo um pirata que adora rum mais do que tudo, ignora a parte das galinhas e decide ir atrás do pé de feijão para conseguir a vaca que produz rum. Simultaneamente, o seu tio (Capitão Barba Rosa, o pirata mais temível, bem vestido e cheiroso dos sete mares, além de antagonista da história) também escuta o relato e decide perseguir o mesmo objetivo, mas com foco nas galinhas de ouro. O problema principal da trama é que PinkBob ouviu essa história durante uma festa, na qual bebeu demais e

fez apenas algumas anotações confusas ("Rum, João, bambu, pé de feijão, vaca de rum, rum"). No outro dia, devido à grande festança, ele acorda em uma viela qualquer, nu, dentro de um barril de rum e munido apenas de suas anotações. PinkBob precisa unir os pontos de suas notas, roubar uma bússola mágica de seu tio (a qual aponta para aquilo que a pessoa mais quer) e descobrir a relação entre o João, o bambu, o pé de feijão e o rum, enquanto compete contra seu tio para ver quem chegará ao destino primeiro.

O segundo passo envolveu a elaboração do GDD de página única abordando os principais pontos necessários para o desenvolvimento do jogo (LUGOSSIAN, 2026; MOTTA; TRIGUEIRO JUNIOR, 2013). Esta etapa foi executada com o seguinte comando:

Prompt: Elaboração do Game *Design* Document (GDD)

Agora, atue como um *Game Designer* Sênior. Com base na ideia de jogo que desenvolvemos na etapa anterior, preciso que você compile e detalhe as informações preenchendo o modelo de documento de *design* (*Game Design Document*) abaixo. Seja direto, técnico e criativo. Preencha todos os campos a seguir:

VISÃO GERAL

TÍTULO: (Nome do jogo)

ID: (Uma frase de alto conceito ou *Logline*)

TEMA: (Temática principal narrativa e estética)

GÊNERO: (Exemplo: RPG, Plataforma, FPS)

AUDIÊNCIA: (Público alvo)

PLATAFORMA: (PC, Mobile, Console)

ESTILO E ARTE

ESTILO VISUAL: (Descreva a direção de arte, exemplo: *Pixel Art*, *Low Poly*, Realista)

MÚSICA E ESTILO SONORO: (Atmosfera do áudio)

MECÂNICA E *GAMEPLAY*

PRINCIPAL OBJETIVO: (O que o jogador deve alcançar no final)

PRINCIPAL MÉTODO PARA ATINGIR O OBJETIVO: (A mecânica chave)

NÚCLEO DE *GAMEPLAY* (*Core Loop*):

Ativo: (Ações diretas do jogador)

Passivo: (Sistemas automáticos ou progressão indireta)

PILARES PRINCIPAIS: (Três a quatro conceitos fundamentais que sustentam o *design*)

CARACTERÍSTICAS DO JOGO: (*Unique Selling Points*, ou seja, o que o torna único)

ELEMENTOS DE JOGO

FACILITADORES: (*Power ups*, itens ou aliados que ajudam o jogador)
BLOQUEADORES: (Obstáculos e Inimigos)
Ativo: (Inimigos ou perigos móveis)
Passivo: (Barreiras ambientais, enigmas lógicos ou limitações de recursos)
CONDIÇÕES DE VITÓRIA: (Como se ganha a fase ou o jogo)
CONDIÇÕES DE DERROTA: (Como se perde)
CONTEXTO
HISTÓRIA: (Sinopse breve do enredo)
TIME: (Recursos humanos necessários estimados para criar este escopo)
META PESSOAL: (Qual o objetivo de desenvolvimento ou aprendizado com este projeto)

Por fim, o terceiro passo requereu a criação da espinha dorsal narrativa (*story spine*) propriamente dita, originada a partir da seguinte instrução:

Prompt: Estruturação Narrativa (Story Spine)

Agora vamos aprofundar a narrativa. Atue como um *Narrative Designer*.
Com base no GDD que acabamos de criar e na etapa de ideação, preciso que você desenvolva a estrutura narrativa do jogo utilizando a técnica Story Spine (Espinha Dorsal da História).
O OBJETIVO: Expandir o campo 'História' do GDD, transformando-o em uma jornada estruturada.
INSTRUÇÃO IMPORTANTE: Você tem liberdade criativa para modificar ou evoluir personagens e eventos específicos se isso tornar a trama mais engajante ou se encaixar melhor nas mecânicas de jogo definidas anteriormente. O foco é a qualidade dramática e a coerência com o gameplay.
Por favor, apresente a resposta no seguinte formato:
1. Resumo das Alterações (Liste brevemente quais personagens ou eventos você modificou em relação à ideia original e porquê).
2. *Story Spine* Preencha a estrutura abaixo:
Era uma vez... (Introdução do mundo e do protagonista em seu estado normal).
Todos os dias... (A rotina e o status quo do protagonista).
Mas, um dia... (O Incidente Incitante que quebra a rotina).
Por causa disso... (Primeira consequência/ação do jogador - Início da jornada).
Por causa disso... (O desenrolar da ação, obstáculos e aprendizado das mecânicas, repita quantas vezes for necessário).
Por causa disso... (O clímax ou ponto de virada principal).

```
Até que finalmente... (A resolução do conflito/batalha final).
E desde então... (O novo normal/conclusão).
3. Conexão com o Gameplay Explique brevemente como os pontos 'Por causa
disso' se conectam com o Core Loop (Núcleo de Gameplay) definido no GDD.
```

Os resultados completos possuem uma extensão muito grande e não foram incluídos de maneira integral no corpo do texto principal. Contudo, todos os artefatos gerados podem ser consultados no repositório indicado no Apêndice A.

Em relação ao passo de expansão da ideia e à posterior elaboração do documento técnico (*One Page GDD*) (LUGOSSIAN, 2026; MOTTA; TRIGUEIRO JUNIOR, 2013), todos os modelos apresentaram desempenho satisfatório. As três ferramentas entregaram estruturas organizadas que respeitaram os comandos de entrada de forma equivalente, não havendo destaque isolado para nenhuma delas nesta tarefa específica. Com o desempenho equivalente, decidiu o acesso: o modelo Google Gemini 3 Pro (GOOGLE, 2026a) foi escolhido para conduzir estes dois primeiros passos nos estudos de caso subsequentes, porque a plataforma disponibilizava a sua versão mais avançada (com limites superiores de capacidade) de maneira gratuita para estudantes durante o período de realização da pesquisa. Por essa via, a produção dos três estudos de caso empregou o Google Gemini 3.1 Pro, liberado antecipadamente para contas de estudante e lançado publicamente em versão de prévia em fevereiro de 2026, enquanto a comparação relatada nesta seção foi conduzida com o Google Gemini 3 Pro.

Para a criação da estrutura narrativa (*story spine*), os resultados divergiram qualitativamente. A Tabela 5 apresenta um recorte comparativo do segmento "Todos os dias..." no texto de controle, redigido pelo autor, e em cada modelo, evidenciando as diferenças de abordagem conceitual de cada ferramenta.

A análise das gerações demonstra que a ferramenta ChatGPT 5.2 (OPENAI, 2025) produziu uma espinha dorsal muito resumida e com pouca clareza de detalhes, resultando em uma entrega superficial, embora a linha do enredo tenha permanecido relativamente próxima ao material concebido pelo autor.

O modelo Google Gemini 3 Pro (GOOGLE, 2026a) adotou uma postura semelhante. O principal fator desmotivador para a sua utilização, contudo, foi o exagero na aplicação do elemento anacrônico: o modelo alterou drasticamente o período histórico no qual o jogo se ambienta (conforme evidenciado pela menção a redes sociais na Tabela 5), transformando o anacronismo, que deveria ser apenas um artifício cômico pontual, em uma intervenção excessiva e descaracterizante.

Por fim, a ferramenta Claude 4.5 Sonnet (ANTHROPIC, 2025) apresentou o melhor balanço entre criatividade e respeito à história original projetada pelo autor. O modelo inseriu tons de anacronismo na medida exata e detalhou com riqueza cada ponto da

Tabela 5 – Exemplo do trecho “Todos os dias...” no texto de controle e em cada modelo.

Origem	**Resultado**
Autor	Todo dia, Pinbob J., um pirata malandro, porém inocente, sobrinho do terrível Barba Rosa, busca uma forma de conseguir aquilo que todo pirata que se preze mais cobiça... aquilo que é mais importante que qualquer quantidade de ouro ou riquezas... mais importante que uma boa aventura ou a amizade... ele busca uma forma de conseguir mais rum!
Google Gemini 3 Pro	Todos os dias... PinkBob tenta, sem sucesso, provar seu valor para conseguir seu próprio navio e suprimento vitalício de rum, enquanto seu tio o humilha publicamente, postando seus fracassos no “Pombo-Gram” (rede social via pombos).
ChatGPT 5.2	Todos os dias, PinkBob bebia, contava histórias exageradas em tavernas, evitava responsabilidades e vivia à sombra de seu tio, Barba Rosa, o pirata mais temido, elegante e respeitado dos mares — alguém que PinkBob admirava e desprezava em igual medida.
Claude 4.5 Sonnet	TODOS OS DIAS... PinkBob acordava em alguma taverna diferente, evitava o tio Barba Rosa (que frequentemente o criticava em público), fazia bicos para conseguir rum, e sonhava com uma grande aventura que o tornaria respeitável - mas sem muito esforço de sua parte. Sua rotina envolvia: acordar com ressaca → procurar trabalho fácil → ser rejeitado por incompetência → conseguir rum de alguma forma questionável → beber em alguma taverna → repetir. Outros piratas zombavam dele, especialmente por suas roupas cor-de-rosa, mas PinkBob não se importava muito - afinal, ele tinha rum. Barba Rosa, enquanto isso, acumulava riquezas através de empreendimentos questionáveis mas lucrativos, sempre esfregando seu sucesso na cara do sobrinho, mas nunca o abandonando completamente (havia afeto familiar distorcido ali).

Fonte: Elaborada pelo autor (2026).

narrativa, descrevendo claramente como os eventos se conectam com as mecânicas de jogo e quais são as interações pretendidas. Diante destes resultados, o Claude 4.5 Sonnet foi eleito como a ferramenta padrão para a estruturação narrativa (*story spine*) em todos os estudos de caso da pesquisa.

4.2.2 Pesquisa e testes com modelos de imagens

A etapa de geração de imagens foi avaliada por meio de uma avaliação qualitativa estruturada, com pontuação ponderada, na escala Likert apresentada na Tabela 2 da Seção 3.3. Os modelos selecionados para esta etapa combinaram opções de código aberto que executam diretamente na própria máquina e soluções baseadas em nuvem. Em especial,

as alternativas de código aberto visaram utilizar interfaces de execução local de modelos de difusão, de fácil acesso e amplamente documentadas, como o Stable Diffusion WebUI (AUTOMATIC1111, 2025) e o ComfyUI (COMFYUI, 2025a). Ambas as opções são de acesso gratuito e utilizam a capacidade de processamento do computador do usuário para executar os modelos de IA-Gen localmente. As ferramentas selecionadas, bem como suas especificações, limitações e direitos de uso, estão consolidadas na Tabela 6.

Tabela 6 – Comparativo das ferramentas de IA-Gen para geração de imagens (novembro e dezembro de 2025).

Ferramenta e Operação	**Limites e Funcionalidades (Plano Gratuito)**	**Limites e Funcionalidades (Plano Pago)**	**Direitos Comerciais e Observações**
Flow Studio Nano Banana Pro (GOOGLE, 2025a) (Nuvem)	Cota de 100 créditos mensais (geração de imagens gratuita). Acesso aos modelos Veo 3.1.	Custo de R$ 24,99 mensais para 200 créditos. Geração de imagens gratuita. Acesso expandido aos modelos Veo 2 e Veo 3.1.	Direitos autorais do usuário (condicionado aos termos de serviço). **Resolução:** 1365x768 (suporta ampliação para 2752x1536).
Kling AI (KLING AI, 2025) (Nuvem)	Cota de 166 *tokens* mensais (cerca de 44 gerações). Acesso às funcionalidades base mediante saldo.	Custo de US$ 6,99 mensais para 660 *tokens*. Geração de imagens, vídeos, efeitos sonoros, avatares e sobreposição virtual de vestuário.	Termos vagos de propriedade intelectual. Exige citação direta da plataforma. **Resolução:** 1536x2720 (suporta ampliação).
Gemini Nano Banana Pro (GOOGLE, 2026a) (Nuvem)	Geração limitada, dependente da demanda atual dos servidores (recurso restrito no plano gratuito).	Custo de R$ 24,99 mensais para 200 créditos. Geração de imagens gratuita no plano Pro. Suporta comandos extensos (cerca de um milhão de *tokens*).	Direitos autorais do usuário (condicionado aos termos de serviço). **Resolução:** Otimizada dinamicamente pela ferramenta.
Juggernaut XL (CIVITAI, 2025b) (Local)	Solução aberta e gratuita. Permite a geração irrestrita de imagens de acordo com a capacidade do computador do usuário.	Não aplicável. O desempenho depende exclusivamente do processamento local.	Uso comercial exige solicitação formal de licença via correio eletrônico institucional. **Resolução:** Recomendada 832x1216 (máximo de 2048x2048).
NetaYume Lumina (DUONGVE, 2025) (Local)	Solução aberta e gratuita. Permite a geração irrestrita de imagens de acordo com a capacidade do computador do usuário.	Não aplicável. O desempenho depende exclusivamente do processamento local.	Uso comercial liberado sob as diretrizes da Licença Apache 2.0. **Resolução:** Recomendada 1024x1024 (máximo de 2048x2048).
CHEYENNE (CIVITAI, 2025a) (Local)	Solução aberta e gratuita. Permite a geração irrestrita de imagens de acordo com a capacidade do computador do usuário.	Não aplicável. O desempenho depende exclusivamente do processamento local.	Direitos integrais das imagens concedidos ao usuário. Proibida a distribuição do arquivo fonte do modelo. **Resolução:** Sem recomendação fixa (máximo de 2048x2048).

Fonte: Elaborada pelo autor (2026).

Foram realizados 36 testes para cada um dos modelos, englobando a geração de três

personagens do jogo (PinkBob, Capitão Barba Rosa e João Feijão) em três perspectivas distintas (frente, perfil e costas), três objetos (bússola, mesa e espada) em três perspectivas diferentes, e três cenários (Cabine do Barba Rosa, Bar Pirata e Centro da cidade) em três visualizações de câmera (2D de frente, 2D isométrico e 2D de cima para baixo). O escopo incluiu também três texturas para a composição de cenários (madeira, mar e papel de parede), além da arte da capa, da arte da tela de configurações, da arte da tela de ajuda e de elementos de interface gráfica (botão, ícone de inventário e cursor). Vale ressaltar que, durante esta etapa do trabalho, o escopo definitivo do jogo ainda não havia sido fechado. Contudo, devido ao alinhamento com a disciplina Design e Desenvolvimento de Jogos, a temática de piratas já se mostrava como uma forte candidata. Por esse motivo, os ensaios foram projetados com tais elementos temáticos, embora não tenham sido necessariamente aproveitados no artefato final por se tratar exclusivamente de testes exploratórios.

Todos os testes receberam avaliação conforme os critérios definidos na Tabela 2, e os registros foram organizados em uma planilha Google Sheets. Os resultados individuais detalhados encontram-se disponíveis no repositório indicado no Apêndice A. A partir das notas atribuídas, calculou-se a média de cada métrica nos 36 testes e, com os pesos da Tabela 2, a pontuação ponderada de cada ferramenta, expressa também em percentual da pontuação máxima, de 71 pontos. Os dados consolidados estão apresentados na Tabela 7.

Tabela 7 – Desempenho médio e métricas das ferramentas de imagem avaliadas.

Modelo Analisado	**Nota (%)**	**Pontos**	**Resolução**	**Visual**	**Enquadramento**	**Alinhamento**	**Texto**	**Legibilidade**	**Alucinação**
Peso			*1*	*2*	*1*	*3*	*1*	*2*	*3*
Gemini Nano Banana Pro	83,1	59,0	3,94	4,97	5,00	4,44	Sim	1,64	5,00
Flow Studio Nano Banana Pro	81,7	58,0	3,00	4,94	5,00	4,75	Sim	1,47	5,00
Kling AI	76,1	54,0	4,00	5,00	5,00	4,36	Sim	0,83	4,00
NetaYume Lumina (ComfyUI)	65,5	46,5	3,00	4,33	4,75	3,64	Sim	0,36	3,00
Juggernaut XL	54,2	38,5	3,00	3,89	5,00	2,44	Sim	0,11	2,00
CHEYENNE	47,9	34,0	3,00	3,36	3,81	1,92	Não	0,00	2,00

Fonte: Elaborada pelo autor (2026).

Conforme pode ser observado nos resultados estruturados na Tabela 7, os modelos que operam em nuvem obtiveram um desempenho consideravelmente superior em comparação às opções de código aberto executadas localmente. Este comportamento era esperado. Os modelos oferecidos em nuvem são, em geral, maiores e mais recentes e rodam em infraestrutura dedicada, enquanto os modelos locais ficam limitados ao que cabe na memória da placa de vídeo do computador usado nos testes (ver Tabela 8). A vantagem aparece sobretudo na qualidade visual, no alinhamento com a diretriz de estilo

e na ausência de alucinações; na resolução, só o Gemini Nano Banana Pro e o Kling AI superaram os modelos locais.

Tabela 8 – Especificações técnicas do computador utilizado para as gerações locais.

Componente	**Especificação**
Processador	AMD Ryzen 5 5600X
Memória RAM	32 GB DDR4 (3200 MHz)
Placa de Vídeo	NVIDIA GeForce RTX 2060 (12 GB VRAM)
Armazenamento	1 TB SSD NVMe PCIe 4.0
Sistema Operacional	Windows 11

Fonte: Elaborada pelo autor (2026).

É importante destacar que, embora o modelo Gemini Nano Banana Pro (GOOGLE, 2026a) tenha alcançado a maior pontuação geral ponderada, a ferramenta Flow Studio Nano Banana Pro (GOOGLE, 2025a) foi a eleita para conduzir as etapas subsequentes do projeto. O critério que favoreceu o Gemini Nano Banana Pro na pontuação foi a sua resolução nativa, de 2816x1536 pixels, ao passo que o Flow Studio gera os arquivos originais em 1365x768 pixels. No entanto, o Flow Studio oferece a funcionalidade integrada de ampliar posteriormente as imagens geradas para 2752x1536 pixels, sem a cobrança de custos adicionais. Com as duas notas praticamente empatadas (83,1 e 81,7), esse fator prático, somado ao maior controle sobre os parâmetros de geração, à interface de usuário mais intuitiva e à presença de ferramentas dedicadas exclusivamente à edição de imagens, decidiu a escolha do Flow Studio Nano Banana Pro para as etapas seguintes.

Figura 3 – Comparativo da geração do personagem João entre diferentes modelos.



Fonte: Elaborada pelo autor com auxílio de IA-Gen (2026).

Entre os resultados, é possível observar na Figura 3 a geração do personagem João Feijão de frente, concebido como um dramaturgo de proporções exageradas e com traços de um aristocrata rico, empregando o estilo artístico de desenho. O mesmo comando de entrada foi submetido às seis ferramentas de IA-Gen avaliadas. Muito embora a instrução textual tenha sido idêntica, cada modelo produziu uma imagem consideravelmente diferente. Tal variação decorre das diferenças entre as ferramentas (o modelo, os dados com que foi treinado e a configuração de cada interface) e do processo de randomização, o qual é parte das arquiteturas generativas e faz com que até um mesmo modelo gere imagens diferentes a cada execução do mesmo comando. Nos testes desta etapa, foram mantidos os parâmetros padrão de cada ferramenta, com semente (*seed*) aleatória, e cada comando foi executado uma vez, sendo avaliada a primeira imagem gerada.

Nesta comparação, é possível constatar que até mesmo o modelo Gemini Nano Banana Pro (GOOGLE, 2026a) apresentou um nível perceptível de alucinação visual, gerando o personagem em diversas poses simultâneas, incluindo uma figura representada apenas por sua silhueta. Em contrapartida, modelos como o CHEYENNE (CIVITAI, 2025a) e o Juggernaut XL (CIVITAI, 2025b) produziram imagens com estética de realismo, a despeito da instrução explícita no comando para a adoção de um estilo ilustrado. Esta divergência estrutural é possivelmente justificada pelo viés da base de dados utilizada pelos criadores das ferramentas no treinamento destes modelos específicos.

Além da ferramenta NetaYume Lumina (DUONGVE, 2025), apenas os modelos operados em nuvem alcançaram um resultado convergente com a estética de ilustração desejada. Em destaque, o ativo digital (*asset*) concebido pelo modelo Flow Studio Nano Banana Pro (GOOGLE, 2025a) nos ensaios obteve o melhor desempenho visual e foi posteriormente integrado à versão final do jogo.

Padrões análogos foram observados em outras categorias de geração, como na concepção visual da Bússola do Capitão Barba Rosa. O mesmo comando foi submetido a todas as ferramentas de IA-Gen, contendo a instrução explícita para produzir o referido *asset* adotando um estilo artístico de desenho.

Novamente, é possível constatar que os modelos CHEYENNE (CIVITAI, 2025a) e Juggernaut XL (CIVITAI, 2025b) produziram imagens com estética realista, ao passo que os modelos hospedados em nuvem mantiveram maior fidelidade à ideia original. O modelo Gemini Nano Banana Pro (GOOGLE, 2026a) manifestou a mesma alucinação visual observada nos testes de personagens, gerando o objeto em múltiplas perspectivas simultâneas, em contraste com a instrução que limitava a renderização estritamente ao ângulo frontal. A totalidade dos comandos de entrada e dos resultados gerados nos testes exploratórios pode ser consultada no repositório indicado no Apêndice A.

Por fim, na Figura 5 é possível observar os resultados referentes à geração da arte da capa conceitual do jogo. Novamente, todos os modelos receberam o mesmo comando, o

Figura 4 – Comparativo da geração do objeto Bússola do Capitão Barba Rosa entre diferentes modelos.



Fonte: Elaborada pelo autor com auxílio de IA-Gen (2026).

Figura 5 – Comparativo da geração de arte da capa entre diferentes modelos.



Fonte: Elaborada pelo autor com auxílio de IA-Gen (2026).

qual continha referências de personagens icônicos da obra, letreiro com o título do projeto e a diretriz para a adoção de um estilo ilustrado.

Neste teste específico, todas as ferramentas convergiram para uma aparência visual

de desenho. No entanto, ainda persiste uma disparidade nítida: as imagens produzidas pelos modelos CHEYENNE (CIVITAI, 2025a) e Juggernaut XL (CIVITAI, 2025b) apresentam aspectos mais voltados ao realismo e significativamente menos cartunizados.

Em contrapartida, é possível notar que apenas quatro dos seis modelos conseguiram incluir elementos textuais na composição da capa. Dentre eles, somente as ferramentas Gemini Nano Banana Pro (GOOGLE, 2026a) e Flow Studio Nano Banana Pro (GOOGLE, 2025a) conseguiram escrever o letreiro de forma legível e exata, registrando com precisão o título *PinkBob J. e o Enigma do Rum*.

4.2.3 Pesquisa e testes com modelos de áudio

A seção de pesquisa e testes com modelos de áudio foi segmentada em duas frentes: a primeira destinada à geração de trilhas sonoras e a segunda voltada para a geração de efeitos sonoros.

4.2.3.1 Pesquisa e testes com trilhas sonoras

Para a geração de trilhas sonoras, foram selecionadas as seguintes plataformas e modelos: Suno (SUNO, 2025), ElevenLabs (ELEVENLABS, 2025), Udio (UDIO, 2025), Stable Audio Open 1.0 (STABILITY AI, 2025) e ACE Step Model (ACE, 2025). O levantamento aprofundado sobre as capacidades, as limitações, os custos e os direitos autorais de cada ferramenta foi realizado com base nos dados disponíveis no período da pesquisa, referente ao dia 12 de novembro de 2025. A Tabela 9 resume as principais características comparativas entre as soluções avaliadas.

A geração de trilhas sonoras para os testes contou com o modelo Google Gemini 3 Pro (GOOGLE, 2026a) atuando como refinador de comandos de entrada. Para a elaboração das instruções musicais, foram criados *prompts* contendo a descrição do estilo da música e, em alguns casos, a respectiva letra. Posteriormente, foi solicitado à ferramenta que realizasse o refinamento dos comandos, visando a sua utilização nos modelos de geração de áudio. Em seguida, os comandos otimizados foram submetidos às ferramentas de IA-Gen para a execução das gerações. Para cada música, as ferramentas receberam o mesmo comando otimizado, sem ajuste por modelo.

A avaliação dos resultados desta etapa ocorreu de forma qualitativa, buscando identificar elementos como a presença de ruídos, cortes abruptos na música, coesão da letra, harmonia e o alinhamento da ferramenta com o comando de entrada (ou seja, a verificação se o modelo seguiu o estilo e as letras propostos).

Nesses testes, alguns resultados específicos chamaram a atenção. A ferramenta ElevenLabs (ELEVENLABS, 2025) esgotou o seu limite de uso com a geração de apenas uma música no plano gratuito e, por este motivo, ficou fora da comparação final das trilhas;

Tabela 9 – Comparativo das ferramentas de IA-Gen para geração de trilhas sonoras (novembro de 2025).

Ferramenta e Operação	**Limites e Funcionalidades (Plano Gratuito)**	**Limites e Funcionalidades (Plano Pago)**	**Direitos Comerciais e Observações**
Suno (Nuvem)	50 *tokens* diários (cerca de cinco músicas). Permite gerar faixas e letras, adicionar vocais e baixar o áudio. Ferramenta utiliza dados para treinamento.	Custo de R$ 50,00 mensais para 500 faixas. Acesso ao editor, criação de identidade artística e gerações simultâneas.	Uso comercial exclusivo para assinantes (direito mantido após cancelamento).
Udio (Nuvem)	10 *tokens* diários (cerca de duas músicas) somados a 100 *tokens* únicos de boas vindas. Permite gerar, aplicar *remix* e recortar a duração. Ferramenta utiliza dados para treinamento.	Custo de US$ 10,00 mensais para 2400 *tokens* (cerca de 480 músicas). Permite usar músicas base como inspiração e realizar gerações simultâneas.	Uso comercial gratuito exige citação (atualmente inviabilizado pela ausência de *download*).
ElevenLabs (Nuvem)	10 mil *tokens* mensais (cerca de três músicas). Permite gerar letras, prolongar a duração e mesclar faixas. Possui opção de exclusão de treinamento.	Custo de US$ 11,00 mensais para cerca de 62 minutos de áudio. Permite edição avançada, dublagem, efeitos sonoros e clonagem vocal.	Uso comercial restrito ao período de assinatura ativa. Sem assinatura, o uso requer citação.
ACE Step Model (Local)	Solução de código aberto e totalmente gratuita. Permite a geração de composições instrumentais e vocais.	Não aplicável. O desempenho depende exclusivamente da capacidade de *hardware* do usuário.	Uso comercial permitido e garantido sob a licença Apache 2.0.
Stable Audio Open 1.0 (Local)	Solução de código aberto e totalmente gratuita. Permite a geração de composições instrumentais e de efeitos sonoros.	Não aplicável. O desempenho depende exclusivamente da capacidade de *hardware* do usuário.	Uso comercial permitido mediante solicitação, restrito a entidades com receita inferior a um milhão de dólares.

Fonte: Elaborada pelo autor (2026).

ela permanece na Tabela 9 como ferramenta levantada, e nos efeitos sonoros foi avaliada normalmente (Tabela 10). Apesar deste contratempo, a música gerada apresentou uma harmonia interessante, embora a pronúncia do idioma português tenha se aproximado consideravelmente do sotaque europeu de Portugal.

Também foi realizado um teste com o objetivo de gerar uma música exclusivamente instrumental para, em seguida, utilizar a própria ferramenta para acrescentar os vocais. Neste teste, ficou constatado que soluções como o Udio (UDIO, 2025) e o Stable Audio Open 1.0 (STABILITY AI, 2025) não permitiam a adição de elementos vocais após a conclusão da geração da música original.

Em suma, o modelo que mais se destacou em todos os testes foi o Suno (SUNO, 2025). A plataforma apresentou um desempenho notável no uso de vocais (respeitando a pronúncia tanto em português quanto em inglês), na harmonia, no alinhamento com o *prompt* e na facilidade de uso das ferramentas internas de edição. O ambiente permite a edição posterior da música, possibilitando a adição de vocais ou a alteração de trechos

específicos. Outro fator decisivo para a escolha foi o limite generoso de cinco músicas por dia no plano gratuito, permitindo que as faixas sejam baixadas e utilizadas para fins não comerciais. Caso as obras tenham sido geradas durante o período em que o usuário possuía um plano pago, os direitos autorais são integralmente concedidos a ele, autorizando o uso comercial. Diante de todos estes atributos favoráveis, este foi o modelo eleito para a geração das trilhas sonoras nas etapas subsequentes da pesquisa. Todos os *prompts* e os respectivos arquivos resultantes podem ser consultados no repositório indicado no Apêndice A.

4.2.3.2 Pesquisa e testes com efeitos sonoros

Para a geração específica de efeitos sonoros, os modelos selecionados englobaram as seguintes ferramentas: ElevenLabs (ELEVENLABS, 2025), Stable Audio Open 1.0 (STABILITY AI, 2025), Kling AI (KLING AI, 2025), SFX Engine (SFX ENGINE, 2025) e Adobe Firefly (ADOBE, 2025). Os dados correspondentes ao comparativo de funcionalidades e limitações foram levantados no dia 13 de novembro de 2025. A Tabela 10 sintetiza as especificações de cada ferramenta, ordenadas conforme a avaliação qualitativa feita nos testes do projeto, sem escala nem pesos.

Tabela 10 – Comparativo das ferramentas de IA-Gen para geração de efeitos sonoros (novembro de 2025).

Ferramenta e Operação	**Limites e Funcionalidades (Plano Gratuito)**	**Limites e Funcionalidades (Plano Pago)**	**Direitos Comerciais e Observações**
Kling AI (Nuvem) Duração de três a dez segundos.	Cota de 166 *tokens* mensais (cerca de 44 efeitos). Acesso integral às funcionalidades do plano pago mediante saldo.	Custo de US$ 6,99 mensais para 660 *tokens*. Geração de imagens, vídeos, efeitos sonoros, avatares e sobreposição virtual de vestuário.	Termos de propriedade intelectual vagos. Exige citação direta indicando o uso de IA-Gen.
Stable Audio Open 1.0 (Local) Limite de mil segundos.	Solução de código aberto e totalmente gratuita. Permite gerar composições instrumentais e efeitos sonoros.	Não aplicável. O desempenho depende exclusivamente da capacidade de processamento do computador do usuário.	Uso comercial permitido mediante solicitação, restrito a entidades com receita inferior a um milhão de dólares.
Adobe Firefly (Nuvem) Duração de um a quinze segundos.	Cota gratuita aproximada de dez *tokens* mensais. Mesmas funcionalidades base do plano pago.	Custo de R$ 47,00 mensais para dois mil *tokens*. Edição de vídeo e imagem, aprimoramento de fala, avatares dinâmicos e traduções.	Uso comercial liberado nativamente, salvo indicação contrária especificada na própria plataforma.
ElevenLabs (Nuvem) Limite de trinta segundos.	Cota de dez mil *tokens* mensais. Permite gerar efeitos sonoros e realizar a transferência dos arquivos de áudio.	Custo de US$ 11,00 mensais para 2500 segundos. Geração de efeitos, clonagem vocal, dublagem e edições musicais.	Uso comercial restrito aos assinantes ativos. Usuários do plano gratuito devem obrigatoriamente incluir citação.
SFX Engine (Nuvem) Limite de 47 segundos.	Cota de trinta *tokens* mensais (equivalente a dez efeitos sonoros completos).	Custo de US$ 10,00 mensais para cinquenta *tokens*. Geração de áudio, música e conversão direta de texto para fala.	Licença para uso comercial garantida e isenta de *royalties* em ambos os planos. Ferramenta voltada para desenvolvedores.

Fonte: Elaborada pelo autor (2026).

A avaliação ocorreu de forma bastante semelhante à metodologia das trilhas sonoras, orientada por uma abordagem qualitativa. O diferencial consistiu nos critérios de observação: nesta etapa, a análise focou em verificar se o som gerado é claro e reconhecível, se apresenta interrupções bruscas e se os áudios possuem a capacidade de operar em reprodução contínua (*loop*), característica essencial para a sonorização repetitiva de ambientes e de movimentações.

Os testes consistiram na geração de efeitos hipotéticos aplicáveis ao projeto temático de piratas, como por exemplo sons de um barco navegando no mar, ruídos de batalhas de espadas e a ambientação sonora de uma taverna (pessoas conversando, móveis sendo movimentados e o tilintar de talheres). Para cada efeito, as cinco ferramentas receberam o mesmo comando de entrada, sem ajuste por modelo. Durante a execução dos testes, os modelos que mais se destacaram foram o Kling AI (KLING AI, 2025) e o Stable Audio Open 1.0 (STABILITY AI, 2025), pois alcançaram uma alta qualidade sonora e respeitaram os critérios estipulados.

Consequentemente, o modelo Kling AI (KLING AI, 2025) foi o eleito para conduzir a geração de efeitos sonoros nas etapas seguintes. Os testes realizados, os *prompts* e os áudios resultantes podem ser consultados no repositório indicado no Apêndice A.

4.2.4 Pesquisa e testes com modelos de vídeo

Para a geração de vídeos, foi adotada a técnica de conversão de imagem para vídeo (*image to video*), utilizando a imagem gerada previamente pela ferramenta Flow Studio Nano Banana Pro (GOOGLE, 2025a) em conjunto com os seguintes modelos de vídeo: Google Gemini Veo 3.1 (GOOGLE, 2026a), Flow Studio Veo 3.1 Fast (GOOGLE, 2025b), Wan2.2 5B (COMFYUI, 2025b), Kling AI (KLING AI, 2025), Hailuo AI (HAILUO AI, 2025), Runway (RUNWAY, 2025) e OpenArt (OPENART, 2025). Em cada teste, as sete ferramentas receberam a mesma imagem de partida e o mesmo comando de entrada, sem ajuste por modelo. O levantamento aprofundado sobre as capacidades, as limitações, os custos e os direitos autorais de cada ferramenta foi realizado com base nos dados disponíveis no período da pesquisa, referente ao dia 24 de novembro de 2025. A Tabela 11 resume as principais características comparativas entre as soluções avaliadas, ordenadas conforme o ranqueamento de qualidade aferido durante os testes.

Todos os testes receberam avaliação conforme os critérios estabelecidos na Tabela 3, e os registros foram organizados em uma planilha Google Sheets. Os resultados detalhados estão disponíveis no repositório indicado no Apêndice A. A partir das notas atribuídas, calculou-se a média de cada métrica nos testes e, com os pesos da Tabela 3, a pontuação ponderada de cada ferramenta, expressa também em percentual da pontuação máxima, de 106 pontos. Os dados consolidados estão apresentados na Tabela 12.

Tabela 11 – Comparativo das ferramentas de IA-Gen para geração de vídeos (novembro de 2025).

Ferramenta e Operação	**Limites e Funcionalidades (Plano Gratuito)**	**Limites e Funcionalidades (Plano Pago)**	**Direitos Comerciais e Observações**
Kling AI (Nuvem) Duração variável.	Cota de 166 créditos mensais. Permite acesso integral às funcionalidades do plano pago mediante o consumo de saldo.	Custo de US$ 6,99 mensais para 660 créditos. Geração de vídeos (com áudio nativo), efeitos sonoros, imagens e sobreposição virtual de vestuário.	Termos de propriedade intelectual vagos. Exige citação direta indicando o uso da plataforma.
Flow Studio Veo 3.1 Fast (Nuvem) Duração de oito segundos.	Cota de 100 créditos mensais. Permite a criação de vídeos a partir de texto ou imagem utilizando os modelos da família Veo 3.1.	Custo de R$ 24,99 mensais para 200 créditos. Acesso aos modelos nas versões Fast e Quality. Áudio requer instrução específica no comando.	Direitos mantidos ao usuário sob os termos de serviço da provedora.
OpenArt (Nuvem) Duração de cinco a dez segundos.	Cota de noventa créditos iniciais. Acesso amplo às ferramentas de edição, criação de histórias animadas e locução.	Custo de US$ 14,00 mensais para quatro mil créditos. Ferramentas avançadas de reestilização, sincronização labial e ampliação de resolução. Áudio opcional.	Obras em domínio público conforme diretrizes vigentes, com uso comercial liberado (exige citação caso utilize gerações de terceiros).
Gemini Veo 3.1 (Nuvem) Duração de oito segundos.	Indisponível no plano base.	Custo de R$ 24,99 mensais. Criação de vídeos a partir de texto ou imagem. Gera vídeos com áudio de forma nativa e automática.	Direitos mantidos ao usuário condicionados à aceitação das políticas de uso corporativas.
Hailuo AI (Nuvem) Duração de seis a dez segundos.	Cota única de 200 créditos com validade de três dias. Duração restrita a seis segundos e ausência de áudio.	Custo de US$ 9,99 mensais para mil créditos. Geração a partir de texto ou imagem.	Uso comercial liberado. O plano gratuito concede licença irrevogável à empresa sobre o conteúdo gerado.
Wan2.2 5B (Local) Duração flexível.	Solução de código aberto. Permite a geração de animações sem áudio. O limite de tempo depende da capacidade do computador do usuário.	Não aplicável. O desempenho e a qualidade dependem exclusivamente do processamento de *hardware* local.	Uso comercial garantido sob a Licença Apache 2.0 (exige manutenção do aviso de direitos autorais).
Runway (Nuvem) Duração de cinco a dez segundos.	Cota única de 125 créditos (aproximadamente cinco vídeos). Ausência de áudio nativo nas gerações.	Custo de US$ 15,00 mensais para 625 créditos. Permite modificação de clima e iluminação, criação de imagens e extensão de quadros.	Uso comercial liberado. A empresa utiliza as gerações para treinamento. Exige citação explícita caso integre a interface de programação externa.

Fonte: Elaborada pelo autor (2026).

A condução dos testes revelou-se lenta, podendo exigir vários minutos para uma única geração. Além disso, os custos operacionais são elevados e há uma grande dificuldade dos modelos de IA-Gen no quesito de permanência de objetos, além de uma alta complexidade envolvida na integração dos *assets* de vídeo com o motor de jogo (*engine*). A imagem base utilizada para os testes retrata um personagem pirata com uma perna de madeira, apresentada na Figura 6. Uma das maiores dificuldades observadas foi a manutenção consistente de seus adereços, incluindo a perna de pau. Em diversos casos, o personagem

Tabela 12 – Desempenho médio e métricas das ferramentas de vídeo avaliadas.

Modelo Analisado	**Nota (%)**	**Pontos**	**Som**	**Áudio**	**Resolução**	**Visual**	**Objetos**	**Enquadramento**	**Estabilidade**	**Alinhamento**	**Utilidade**	***Loop***	**Comandos**
Peso			*1*	*2*	*1*	*2*	*3*	*1*	*3*	*3*	*3*	*1*	*2*
Kling AI	77,4	82,0	Sim	3,0	3,1	3,9	3,6	4,6	3,9	4,0	5,0	3,0	3,0
Flow Studio Veo 3.1 Fast	76,9	81,5	Sim	3,1	3,0	3,8	3,3	4,5	4,1	3,8	4,0	2,9	5,0
OpenArt	68,9	73,0	Sim	1,4	1,6	2,4	3,9	4,0	3,9	3,9	3,0	3,0	5,0
Gemini Veo 3.1	67,0	71,0	Sim	3,3	3,0	3,1	2,5	3,8	3,5	3,0	4,0	3,0	5,0
Hailuo AI	66,0	70,0	Não	0,0	2,6	3,4	3,0	3,9	3,1	4,1	5,0	3,0	4,0
Wan2.2 5B	50,0	53,0	Não	0,0	3,0	2,4	1,3	3,1	2,0	2,0	5,0	3,0	5,0
Runway	47,2	50,0	Não	0,0	2,7	2,1	1,9	4,0	3,0	2,7	3,0	3,0	3,0

Fonte: Elaborada pelo autor (2026).

adquiria uma perna humana comum e, repentinamente, a prótese de madeira ressurgia. Outros artefatos visuais indesejados incluíram a fusão de dedos e o desaparecimento abrupto de detalhes da vestimenta.

Figura 6 – Personagem PinkBob conceitual usado para testes com vídeos.



Fonte: Elaborada pelo autor com auxílio de IA-Gen (2026).

Estas instabilidades refletiram de forma direta nos resultados avaliativos, visto que nenhum dos modelos alcançou nota próxima da pontuação máxima (cinco pontos) no critério de permanência de objetos. Adicionalmente, as ferramentas apresentaram perdas

significativas em outras qualidades, evidenciadas por áudios distorcidos e irreais, ou até mesmo pela total incapacidade de gerar sons.

Tendo em vista os custos altos por geração, somados à dificuldade de contínua produção e integração com o motor gráfico Unity (UNITY TECHNOLOGIES, 2026), a pesquisa deliberou pela não inclusão da categoria de vídeos nas etapas subsequentes do projeto. Todos os resultados e *prompts* utilizados nesta etapa podem ser consultados no repositório indicado no Apêndice A.

A categoria de modelos em três dimensões (3D) não chegou a ser testada: ficou fora do escopo desde o início, pela alta complexidade técnica e pelo limite de tempo estipulado para a realização da pesquisa.

4.3 Etapa 3: processo de ideação e seleção

Esta seção apresenta os resultados obtidos a partir do processo de ideação e filtragem conceitual dos projetos de jogos. O texto detalha o desenvolvimento sequencial das propostas concebidas, o refinamento fundamentado nos pilares de *design* e a estruturação lógica das ideias geradas por meio das dinâmicas de *brainstorming* colaborativas.

4.3.1 Ideação inicial e filtragem qualitativa

Nesta etapa, cem conceitos de jogos foram listados de forma espontânea. Cada proposta consistiu em um curto parágrafo representativo da ideia central da obra (focando na história ou nas mecânicas principais). As premissas catalogadas pertenceram a múltiplos gêneros e estilos, englobando ambientes 2D e 3D, com focos variados entre a narrativa e a jogabilidade. A partir deste levantamento geral, foram selecionados dez conceitos para o processo de refinamento estruturado. A filtragem teve fundamentação na afinidade temática e na viabilidade de escopo, resultando no isolamento das seguintes premissas:

32. Depois de acordar nu em um barril vazio de rum, um pirata chamado PinkBob J. precisa descobrir a relação entre um bago de feijão, um poste de bambu, e o que isso tudo tem a ver com ele achar que ontem deveria ser sábado. O jogador controla PinkBob J. em uma aventura maluca pelos sete mares atrás das respostas.

93. Depois de acordar de madrugada e perceber que está sem papel higiênico, Pedro decide sair para comprar mais. Porém o trajeto até a loja de conveniência não é seguro, bandidos e um assassino em série estão à solta. O jogador precisa chegar à loja de conveniência e voltar para casa em segurança, enfrentando os bandidos e fugindo do assassino.

10. Bob, o paralelepípedo. O jogador é um paralelepípedo em um estranho mundo habitado apenas por formas redondas, onde seres com formas retas são abominados como monstros. Ele precisa explorar o ambiente, resolvendo desafios lógicos para encontrar seu lugar nessa sociedade.

55. Um operador de telégrafo durante a Primeira Guerra Mundial precisa desvendar um mistério usando as chamadas de código Morse e documentos a sua disposição antes que seja tarde demais.

37. Zlipzorb 7249-b é um agente fiscal da fronteira de Andrômeda II e deve verificar as naves dos alienígenas em busca de contrabando.

74. Perspectiva. O jogador acorda em um local completamente maluco, onde a perspectiva muda tudo, e resolve *puzzles* mudando a posição da câmera, sons ou movimento.

26. O jogador assume o papel de "O Panificador", um justiceiro mascarado com vestes de padeiro, que durante o dia tem uma padaria e à noite usa sua *baguette* francesa em nome da justiça. Durante o dia o jogador faz pães e à noite combate o crime.

19. O jogador assume o papel de presidente, criando sua própria república de bananas. O jogador deve equilibrar a sua corrupção com o crescimento da sua "república" sem deixar que o sistema colapse.

53. O jogador é um despachante policial, e deve atender aos chamados dos cidadãos, priorizando aqueles que são uma emergência de verdade e tentando dar instruções para que eles sejam salvos.

21. Se eu tenho uma arma é para atirar: O jogador aparece em meio a uma confusão sideral, sem nenhum contexto, apenas uma arma e alienígenas correndo/atirando em sua direção. O objetivo é derrotar todos os aliens.

Com as dez propostas selecionadas, procedeu-se à expansão conceitual estruturada sobre três pilares fundamentais de desenvolvimento: mecânicas de jogo (*gameplay*), narrativa (*storytelling*) e experiência do usuário (*user experience*). Este desdobramento permitiu a visualização do escopo geral dos projetos, facilitando o planejamento de desafios de programação, mapeamento de tecnologias e demandas de ativos digitais (*assets*). A título de exemplificação do resultado obtido neste passo, o refinamento da ideia de jogo 32 estruturou-se da seguinte forma:

- ***Gameplay*:** Consiste na mecânica *point and click*. O jogador utiliza o *mouse* para clicar em objetos e personagens do cenário, podendo coletar itens, combiná-los e interagir com o ambiente e com cada personagem não jogável, ou *Non Playable Character* (NPC). O avanço exige a resolução de desafios lógicos envolvendo a ordem

e o local de uso de cada objeto, sendo possível obter dicas e objetivos dialogando com os personagens.

- ***Storytelling*:** A obra é ambientada no Caribe, na famosa ilha de Tortuga, lar de uma grande vila que serve de refúgio para piratas escaparem das autoridades. A ilha é extremamente perigosa, habitada pelos piratas mais terríveis de todos os tempos. PinkBob é um pirata inocente, porém malandro, expulso de seu antigo barco por cobiçar a namorada do capitão, vagando pelas ruas em busca de novas oportunidades. Nossa história, no entanto, começa quando PinkBob não se recorda de sua última noite, ele acorda nu em um barril de rum vazio abandonado em uma viela qualquer, claramente bêbado, munido apenas de um bago de feijão, um papel com desenho de um poste de bambu e uma estranha sensação de que ontem com certeza tinha sido sábado.
- ***User Experience* (UX):** Uma mescla de aventura com comédia, inspirada em clássicos como *The Secret of Monkey Island* e *Indiana Jones and the Fate of Atlantis*. O jogador deve sentir o apelo exploratório para desvendar o mistério absurdo do protagonista.

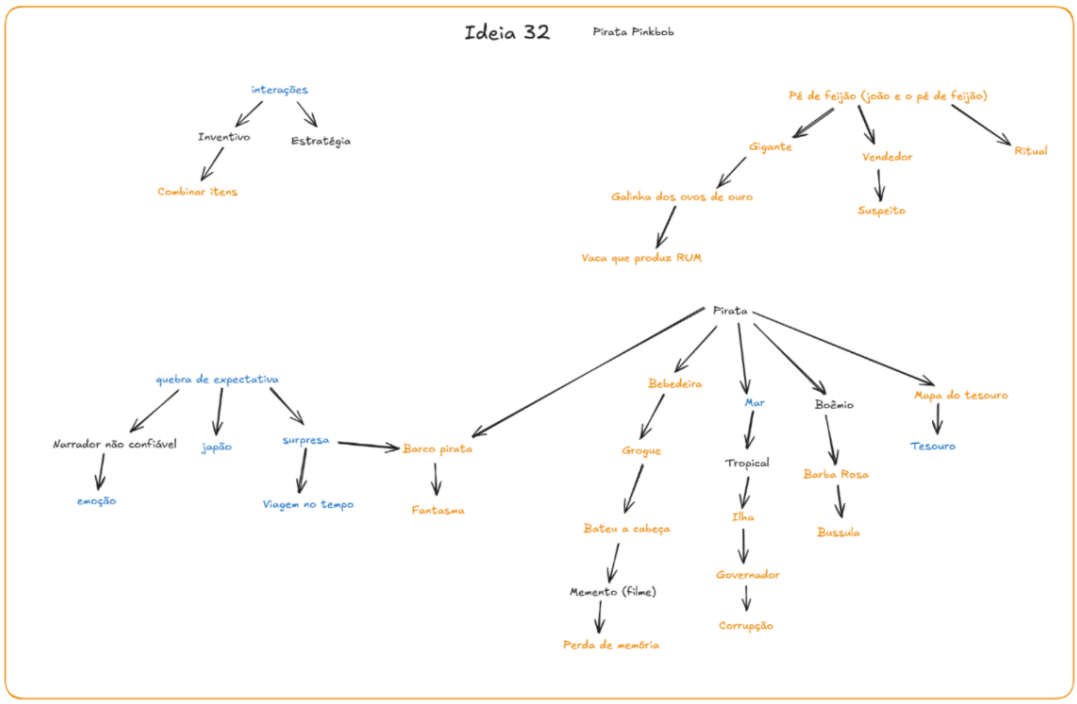
Com os dez conceitos consolidados, foi realizado um segundo corte qualitativo, preservando apenas cinco ideias de jogo: 32, 93, 10, 55 e 53. Estas propostas foram submetidas a uma sessão de *brainstorming* estruturado. Durante a dinâmica, os voluntários receberam a instrução de listar a maior quantidade possível de palavras associadas à ideia lida dentro de um limite de tempo de dois minutos.

Na sequência, utilizando o vocabulário coletado, foram elaboradas as árvores de ideias. Este processo conectou as ramificações de palavras de acordo com afinidades semânticas ligadas aos mesmos três pilares do refinamento — jogabilidade (*gameplay*), narrativa (*storytelling*), nela incluída a composição de personagens, e experiência do usuário, gerando a base conceitual para o aprofundamento dos jogos. A Figura 7 ilustra o resultado gráfico gerado para a ideia de jogo 32, onde cada tonalidade indica a contribuição colaborativa de um participante distinto.

As demais árvores de ideias geradas estão disponibilizadas no repositório indicado no Apêndice A, todas adotando a mesma padronização visual e estrutural.

Por fim, as árvores resultantes foram apresentadas a um segundo grupo de voluntários para a coleta de *feedback* qualitativo. As avaliações obtidas foram combinadas às estruturas das árvores para fundamentar a redação da versão final do refinamento conceitual de cada um dos cinco jogos. Esta síntese confirmou o *PinkBob J. e o Enigma do Rum*, candidato desde a disciplina (Seção 4.2), como primeiro estudo de caso. As versões completas destas cinco propostas refinadas estão disponíveis para consulta no repositório indicado no Apêndice A.

Figura 7 – Árvore de ideias da ideia de jogo 32, que foi escolhida.



Fonte: Elaborada pelo autor (2026).

4.4 Etapa 4: primeiro estudo de caso — *PinkBob J. e o Enigma do Rum*

Esta seção apresenta os resultados obtidos no desenvolvimento do primeiro jogo estudo de caso, incluindo *design* do jogo, documentação, elaboração de protótipos e criação da versão final do jogo.

4.4.1 Definição do estudo de caso

Após o refinamento das cinco ideias finais, o projeto confirmado como primeiro estudo de caso foi *PinkBob J. e o Enigma do Rum*, o candidato desde a disciplina, que o refinamento manteve por apresentar maior afinidade temática. A escolha justificou-se principalmente pela abordagem cômica do universo pirata e pela vasta quantidade de ativos digitais (*assets*) para possível geração, incluindo cenários altamente distintos, personagens e objetos.

Para consolidar a escolha e nortear todo o processo de desenvolvimento, o primeiro artefato elaborado foi o documento de *design* do jogo de página única (*One Page GDD*), o qual pode ser consultado de maneira integral no repositório indicado no Apêndice A. Em resumo, os principais pontos definidos no documento incluem:

- **Tema**: Piratas, Caribe e Caça ao Tesouro.
- **Gênero**: Comédia, Aventura e Quebra-cabeça.
- **Pilares principais**: Protagonista, rum, Barba Rosa, quebra-cabeças, quebra de expectativa, paródia, personagens malucos e mentiras.
- **Estilo visual**: Visual Cartoon, deve passar o tom engraçado do jogo, de preferência que faça referência ao jogo *The Secret of Monkey Island: Special Edition*.
- **História**: Aventura de comédia ambientada na lendária ilha pirata de Tortuga, Caribe, inspirada em clássicos como *The Secret of Monkey Island* e a história de "João e o pé de feijão". O jogador assume o papel de PinkBob J., um pirata malandro e que odeia sabonete, que acorda de uma grande ressaca, nu dentro de um barril vazio de rum. Sem memória da noite anterior, sua única pista é uma anotação enigmática que envolve um cara chamado João, um feijão mágico, um japonês, bambu, e uma vaca que produz rum.
- **Audiência**: Adultos.
- **Núcleo de *gameplay***: O ciclo de *gameplay* básico do jogo envolve interagir com um personagem/ cenário, encontrar o item correto, resolver um quebra-cabeça, retornar ao personagem/cenário e utilizar o item obtido. Ou, pode ser apenas descobrir uma informação que precisa ser dita ao personagem.
- **Principal Objetivo**: Encontrar a famigerada vaca que produz rum.
- **Condições de vitória**: Chegar ao final da história e encontrar a vaca de rum.
- **Condições de derrota**: Não há um método para perder, no máximo o jogador pode ficar preso em uma fase por não saber o que fazer.

As principais inspirações para a definição do tema do jogo incluem títulos do estúdio LucasArts, como *The Secret of Monkey Island* (LUCASFILM GAMES, 1990), *Indiana Jones and the Fate of Atlantis* (LUCASARTS, 1992) e *Full Throttle* (LUCASARTS, 1995). Todas estas obras pertencem ao gênero apontar e clicar (*point and click*) e introduzem mecânicas de coleta de itens, diálogos com personagens não jogáveis e interação com o cenário, sempre permeadas por um forte tom de comicidade e paródia. Destaca-se, em especial, o título *The Secret of Monkey Island*, por também se tratar de uma sátira do universo dos piratas, servindo como referência direta para a composição do estilo artístico do projeto *PinkBob J. e o Enigma do Rum*.

4.4.2 Expansão da história

Com o GDD, a próxima etapa englobou o refinamento das ideias para uma estrutura narrativa (*story spine*). Seguindo a ordem de criação de histórias ("Era uma vez...", "Todos os dias...", "Mas, um dia...", "Por causa disso...", "Até que finalmente..." e "E desde então..."), a primeira versão da história do jogo foi redigida de forma manual. Posteriormente, utilizando a ideia refinada e o GDD como referência, a ferramenta Claude 4.5 Sonnet (ANTHROPIC, 2025) foi encarregada de gerar a sua versão da estrutura narrativa.

As duas abordagens foram combinadas em uma estrutura final unificada, a qual agrega as ideias originais do autor, as contribuições dos voluntários das dinâmicas de *brainstorming* e as sugestões da IA-Gen. Ambas as versões originais encontram-se disponíveis no repositório indicado no Apêndice A. O enredo resumido é apresentado a seguir.

A história acompanha PinkBob J., um pirata preguiçoso e aficionado por rum, que embarca em uma corrida contra o pirata mais temido dos sete mares, que também é seu tio, o Capitão Barba Rosa, após descobrir a lenda de um pé de feijão gigante que guarda galinhas de ouro e uma mística vaca produtora de rum. Com a ajuda da Capitã Maré Alta, PinkBob decifra pistas, rouba uma bússola mágica do próprio tio e cumpre tarefas para um botânico japonês excêntrico chamado Dr. B. Chen em troca das sementes, percebendo durante a jornada que seu verdadeiro desejo não é apenas a bebida infinita, mas provar o seu valor. Após plantar o feijão e escalar até um castelo nas nuvens, a dupla de piratas se depara com um gigante reformado e adepto de terapia, que decide a posse do tesouro através de um inusitado quiz de conhecimentos piratas transmitido ao vivo. PinkBob vence o desafio, cede a galinha de ouro ao tio para ficar apenas com a vaca de rum e, após finalmente conquistar o respeito de Barba Rosa, inaugura uma taverna lendária, provando que o maior tesouro foi quem ele se tornou durante essa aventura.

As principais divergências entre esta versão final, enriquecida pelas contribuições da ferramenta de IA-Gen, e a versão redigida originalmente pelo autor residem na relação entre o protagonista PinkBob e o Capitão Barba Rosa. Na concepção original, a relação é puramente cômica e antagônica. O desentendimento do tio e PinkBob é devido à sua falta de higiene e, no desfecho da trama, após capturar a vaca de rum, o protagonista rouba o navio do Barba Rosa e foge para um local distante, deixando a ilha em estado de caos absoluto.

Em contrapartida, na versão complementada pela IA-Gen, a rixa entre os personagens é motivada pela preguiça de PinkBob, o qual recusa o trabalho e demonstra vício em rum. Esta premissa cria uma corrida entre sobrinho e tio para definir quem alcançará o pé de feijão primeiro e reivindicará o tesouro. O desfecho dessa versão introduz uma lição de moral, de que o verdadeiro tesouro é a jornada e quem você se torna no final.

O escopo original da trama era contido em uma única ilha, com o progresso dividido

por seções do local. A versão final, influenciada pelas sugestões da ferramenta, expandiu a exploração para ilhas adicionais. Esta ampliação de cenário contribuiu diretamente para a estruturação do jogo em formato episódico. Neste novo formato, cada capítulo acompanha PinkBob na exploração de uma ilha distinta em busca de um item específico ou de um personagem-chave essencial para o avanço da narrativa.

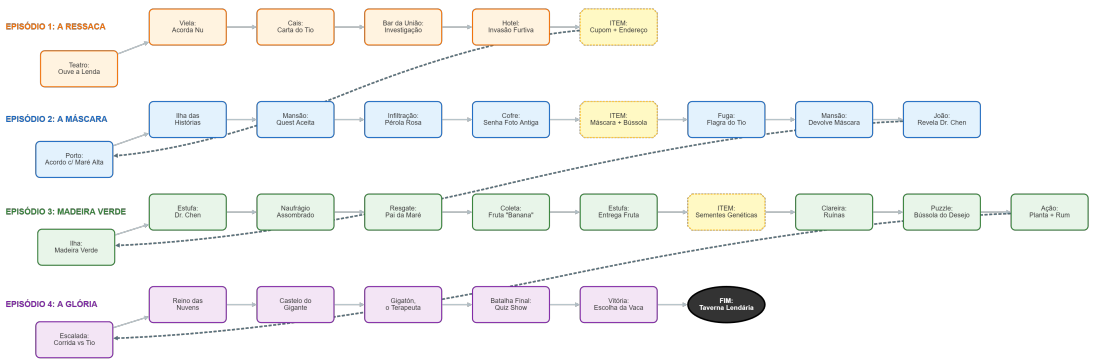
Além das modificações citadas, a versão final expandiu significativamente as interações narrativas e introduziu novos membros no elenco de personagens. O principal destaque é a inclusão da Capitã Maré Alta, que acompanha PinkBob durante a jornada, conduzindo-o de ilha em ilha e narrando histórias absurdas sobre a pirataria ao longo do trajeto. O conceito fundamental é de que a história do jogo seja transmitida por meio da interação ativa do jogador com os personagens integrantes do universo, mediante as escolhas de diálogos que direcionam o fluxo da narrativa.

4.4.3 Protótipos de baixa fidelidade

Nesta etapa, foram concebidos o fluxograma responsável por apresentar a progressão da narrativa, um diagrama hierárquico de cenários por episódios e os protótipos de baixa fidelidade desenvolvidos para ilustrar as mecânicas projetadas para o jogo.

O fluxograma apresenta a progressão da narrativa em um formato resumido com o objetivo de guiar o processo de desenvolvimento. O documento lista apenas os acontecimentos principais que são estritamente necessários para o avanço à próxima etapa do enredo. A estrutura pode ser conferida na Figura 8.

Figura 8 – Fluxograma do jogo PinkBob.

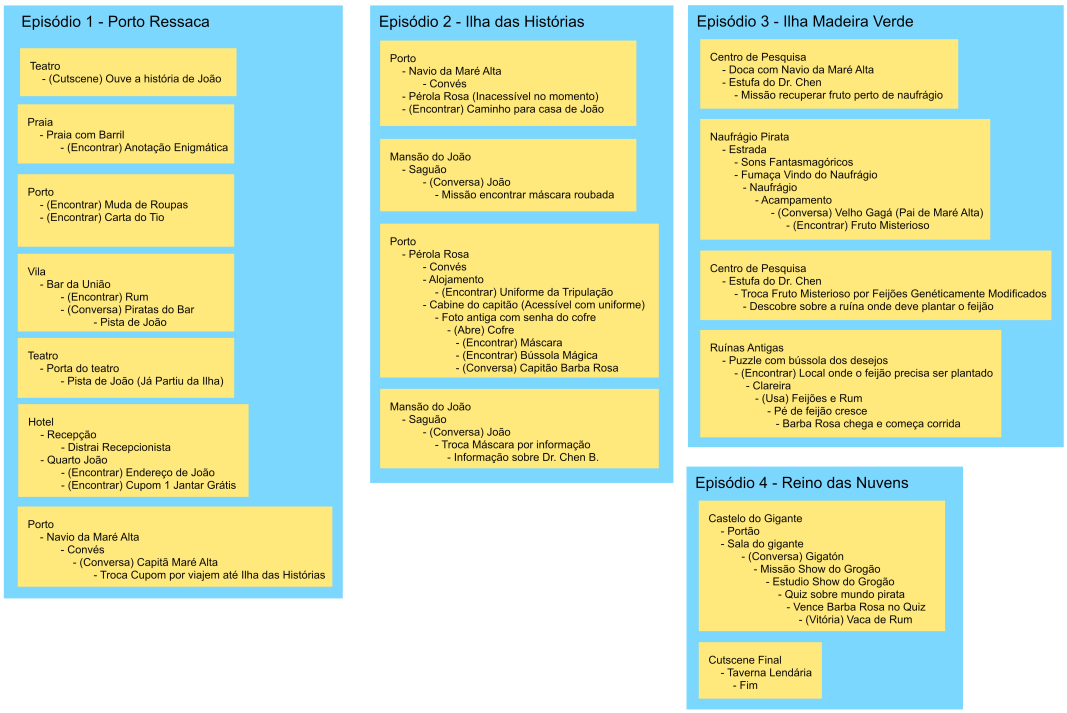


Fonte: Elaborada pelo autor (2026).

Posteriormente, foi elaborado um diagrama hierárquico, dividido por episódios, que cataloga todos os principais cenários por onde o jogador transita, além dos personagens interativos, objetos a serem coletados e interações necessárias para o progresso cronológico da história. Nesta etapa, é possível observar mudanças em relação à ideação original. A

título de exemplo, o início do jogo, que ocorreria em uma viela, passou a ser ambientado em uma praia. Além disso, o protagonista não desperta mais nu, mas apenas acorda na areia após uma intensa noite de embriaguez. Adicionalmente, o objetivo do segundo episódio, que consistia em encontrar o uniforme da tripulação, foi substituído pela busca das chaves da cabine do capitão. O diagrama pode ser visualizado na Figura 9.

Figura 9 – Hierarquia de cenários, ações e personagens por episódios.

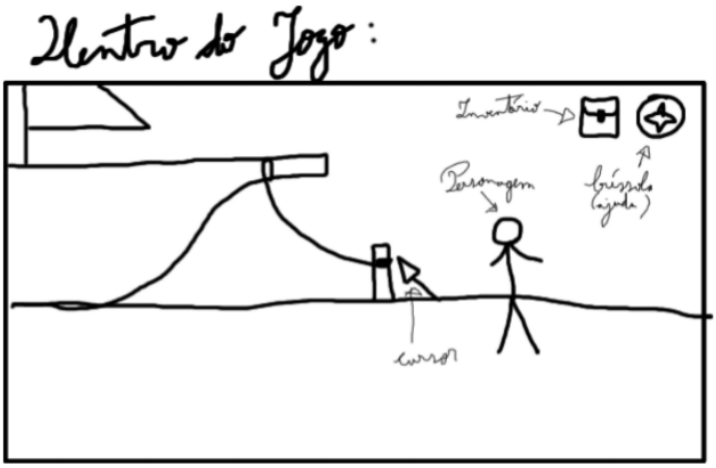


Fonte: Elaborada pelo autor (2026).

Ao final desta etapa, foram produzidos protótipos de baixa fidelidade para ilustrar visualmente o funcionamento do jogo. O protótipo, apresentado na Figura 10, demonstra como a interface do jogador foi concebida, exibindo elementos fundamentais como o inventário, a bússola e o cursor utilizado para a interação com o ambiente.

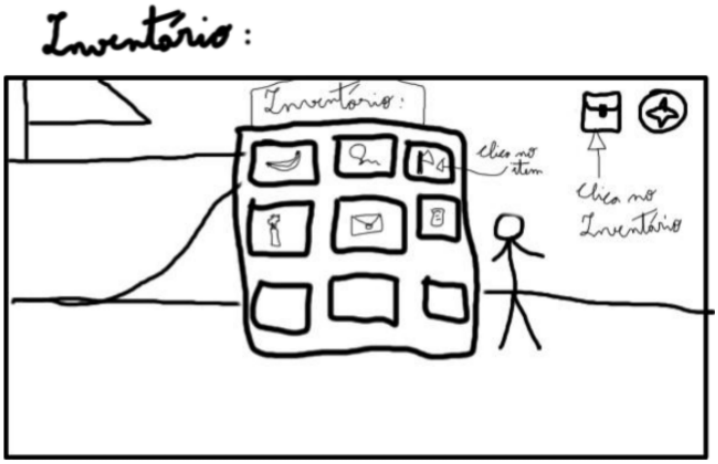
A criação destes protótipos auxiliou de forma significativa na visualização do funcionamento das mecânicas do jogo e da sua apresentação ao jogador. A Figura 11 e a Figura 12 representam a idealização do fluxo lógico de jogabilidade, demonstrando como o jogador poderia selecionar um item em seu inventário para solucionar um enigma de cenário (*puzzle*) em uma das fases projetadas.

Figura 10 – Exemplo de protótipo da visão do jogador.



Fonte: Elaborada pelo autor (2026).

Figura 11 – Exemplo de protótipo de inventário.

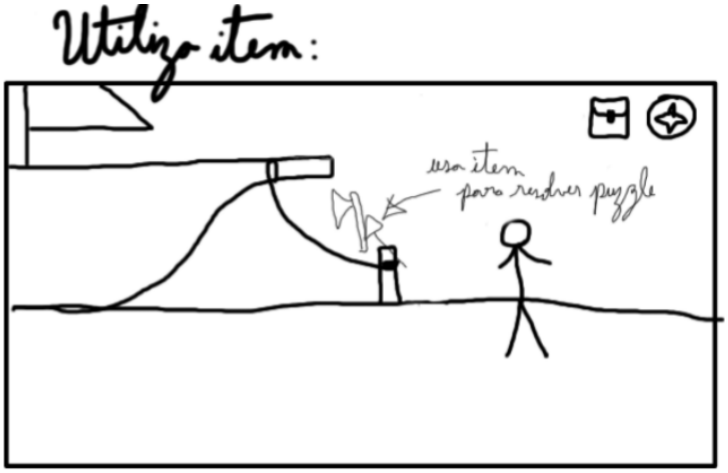


Fonte: Elaborada pelo autor (2026).

4.4.4 Desenvolvimento e gerações com IA-Gen

O desenvolvimento de jogos envolve diversas áreas. O foco desta pesquisa é o uso de ferramentas de IA-Gen como apoio ao processo de *design* do jogo. Por isso, esta seção apresenta as principais gerações de conteúdos feitas para o jogo *PinkBob J. e o Enigma do Rum*, sem aprofundamento na programação e na integração com a *game engine*; o que foi programado pelo autor e o que foi gerado por IA-Gen nos três jogos está resumido na Seção 4.7.

Figura 12 – Exemplo de uso de item para solucionar *puzzle* de cenário.



Fonte: Elaborada pelo autor (2026).

4.4.4.1 Geração de personagens

Para a criação dos personagens do jogo, a estrutura narrativa base (*story spine*) foi submetida à ferramenta Google Gemini (GOOGLE, 2026a). Solicitou-se a criação de um *prompt* focado na geração do protagonista, utilizando como principal inspiração visual o estilo do clássico *The Secret of Monkey Island* (LUCASFILM GAMES, 1990). Em seguida, o comando estruturado foi inserido na plataforma Flow Studio Nano Banana Pro (GOOGLE, 2025a) para a concepção da imagem do personagem base.

Uma vez finalizada a imagem base, o arquivo foi novamente alimentado ao Gemini (GOOGLE, 2026a), acompanhado do *story spine*. Desta vez, solicitou-se a elaboração de um *prompt* voltado para a geração da animação deste personagem em formato de *sprite sheet* (uma folha contendo múltiplos quadros sequenciais que, ao serem combinados, criam a ilusão de movimento). A seguir, é apresentado o comando gerado por esta interação, em inglês, como todos os destinados aos modelos de imagem neste trabalho:

Prompt: Geração de Sprite Sheet (PinkBob)

```
2D Game Asset Sprite Sheet: PinkBob
Overall Style: A clean, polished, high quality 2D digital painting style,
suitable for a fantasy RPG game asset. The character design is consistent
across all instances, precisely matching the pirate protagonist seen in
the reference image.
Character Details (for consistency):
Head: Medium length curly red hair and a full red beard. Wearing a black
tricorn hat with a distinct gold skull and crossbones emblem (as seen in
```

```
the reference image).
Torso: Navy blue pirate frock coat with prominent gold trim along the
collar, cuffs, and front, and gold buttons. A ruffled white shirt is
visible underneath.
Legs: Trousers with vertical blue and grey stripes.
Feet: Dark brown leather high boots.
Pose: Three quarter view facing left.
Sprite Sheet Layout and Composition:
A continuous horizontal line of six distinct full body poses. These six
poses represent a complete, fluid walking animation cycle, progressing
from left to right, exactly as depicted in the reference image. Each pose
shows a different stage of the stride (example: foot forward, mid step
raised foot, foot planted).
Background: A clean, perfectly white background.
Execution Notes: The spacing between all six sprites should be even.
The visual consistency between all instances of the character must be
maintained perfectly.
```

Utilizando este *prompt* no Flow Studio, a imagem final da *sprite sheet* do personagem foi gerada e combinada com a imagem base para a finalização do recurso a ser exportado para a *engine*. O processo envolveu o recorte individual de cada quadro para formar a animação. O resultado da *sprite sheet* final do protagonista PinkBob, contemplando todas as animações, pode ser visualizado na Figura 13.

Este mesmo procedimento foi replicado para as animações de todos os personagens da obra, abrangendo tanto o elenco secundário quanto os figurantes. A título de exemplo, a aplicação desta metodologia para a Capitã Maré Alta resultou na arte final de animação que pode ser observada na Figura 14.

Outro exemplo de aplicação da metodologia é a personagem recepcionista do hotel, integrante do primeiro episódio, e que tem uma animação simulando a leitura de um livro. A personagem encontra-se posicionada atrás do balcão de atendimento em postura de leitura constante. As suas animações foram concebidas seguindo o mesmo fluxo, iniciando-se pela criação de um visual base para, em seguida, expandi-lo em uma sequência de animação composta por três quadros. No interior da *engine*, esta sequência é executada em reprodução contínua (*loop*) com o intuito de transmitir a percepção de movimento ao usuário. A Figura 15 exemplifica o *asset* finalizado. Nas três folhas (Figuras 13 a 15), o traje e os acessórios de cada personagem se repetem de quadro a quadro, que era o objetivo de usar a imagem base como referência em cada pedido.

Figura 13 – Exemplo de *sprite sheet* com todas as animações geradas do PinkBob.



Fonte: Elaborada pelo autor com auxílio de IA-Gen (2026).

Figura 14 – Exemplo de *sprite sheet* com todas as animações da Capitã Maré Alta.



Fonte: Elaborada pelo autor com auxílio de IA-Gen (2026).

Figura 15 – Exemplo de *sprite sheet* com todas as animações da recepcionista do hotel.

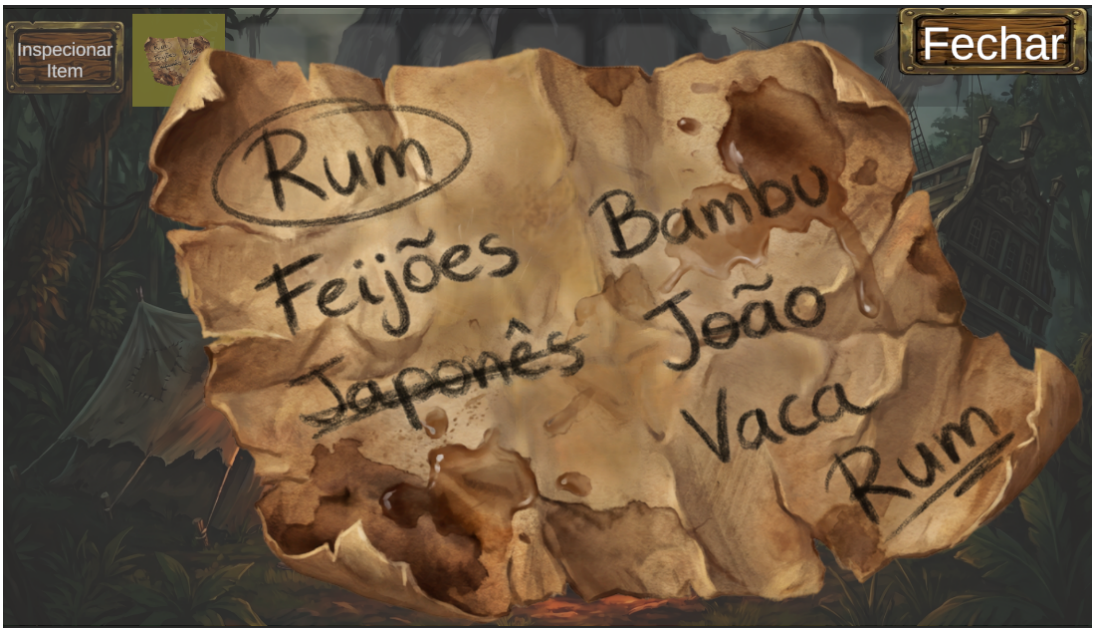


Fonte: Elaborada pelo autor com auxílio de IA-Gen (2026).

4.4.4.2 Geração de objetos

De forma análoga à criação de personagens, a geração de objetos foi realizada utilizando a imagem base do protagonista como referência de estilo artístico. Esta imagem foi submetida à ferramenta Google Gemini (GOOGLE, 2026a), em conjunto com a descrição do item, para a elaboração do *prompt*. O comando resultante foi então utilizado na plataforma Flow Studio Nano Banana Pro (GOOGLE, 2025a) para a concepção do *asset* final. Na Figura 16, é possível observar o exemplo do objeto "Anotações Misteriosas", já implementado no jogo como um item utilizável.

Figura 16 – Exemplo de objeto gerado, "Anotações Misteriosas", implementado no jogo.



Fonte: Elaborada pelo autor com auxílio de IA-Gen (2026).

4.4.4.3 Geração de cenários

Para a criação dos cenários do jogo, foram utilizados desenhos digitais manuais como base. Esses rascunhos foram submetidos à ferramenta Google Gemini (GOOGLE, 2026a), em conjunto com o *story spine*, para servir como estrutura para a geração do *prompt* da arte final. O *prompt* foi então inserido na plataforma Flow Studio Nano Banana Pro (GOOGLE, 2025a), acompanhado da imagem do protótipo inicial, resultando na concepção da arte final.

A título de exemplificação, apresenta-se o comando em inglês elaborado para a geração da taverna do jogo.

Prompt: Geração da taverna do jogo

Prompt:
A sweeping, wide-angle landscape background scene rendered in a rich, hand-painted digital art style with deep, painterly textures and soft, volumetric lighting. The art style perfectly matches the aesthetic and detailed finish of the character sprite.
The scene depicts the expansive, dimly lit interior of the "Unity Bar", emphasizing a very wide, spacious floor plan.
Left Wall (Entrance & Boarded Window): On the far left wall, serving as the main entrance, is a heavy, rough-hewn wooden door with iron hinges. Immediately next to the door, there is a window frame that has been completely sealed off using uneven, mismatched wooden planks nailed haphazardly across the opening, letting only tiny slivers of outside light through the cracks.
Back Wall (The Long Bar): Running across the entire length of the distant back wall is a long, sturdy wooden bar counter. The area behind it is shadowed, featuring shelves cluttered with various bottles and tankards, emphasizing the width of the room.
Right Side (Open Space): The right side of the room is now a wide, open expanse of worn wooden wall and floor, clear of any curtains or pillars, contributing to the feeling of a large, somewhat empty hall. Add tables and chairs to the bar. On the tables, use various items such as plates, glasses, rum bottles, and food.
Atmosphere & Lighting: The room is illuminated by warm, amber ambient light from unseen candles or lanterns, creating a cozy but slightly dingy atmosphere. This light casts soft, long shadows across the expansive wooden floorboards and highlights the rough textures of the door, the haphazard window planks, and the long bar counter.

Unity Bar, no comando, é o nome que a taverna recebeu no projeto. Na Figura 17, é possível observar o comparativo direto entre o protótipo rascunhado manualmente pelo autor e o *asset* finalizado pela IA-Gen, o qual foi efetivamente implementado na *engine* do jogo.

Já na Figura 18, é possível perceber como a IA-Gen foi capaz de preservar detalhes específicos da imagem original, como os rótulos das garrafas de “Rum” e as placas apontando para outros locais da ilha. Além disso, a imagem final traz detalhes que o rascunho não tinha, como a maior quantidade de garrafas espalhadas pela areia e as pegadas no chão. Estas características foram instruções expressamente incluídas no *prompt*, mas que não faziam parte da composição do rascunho original.

Figura 17 – Exemplo de protótipo do interior da taverna desenhado pelo autor, à esquerda, e *asset* final gerado pela IA-Gen, à direita.



Fonte: Elaborada pelo autor com auxílio de IA-Gen (2026).

Figura 18 – Exemplo de protótipo do cenário de praia desenhado pelo autor, à esquerda, e *asset* final gerado pela IA-Gen, à direita.



Fonte: Elaborada pelo autor com auxílio de IA-Gen (2026).

4.4.4.4 Geração de plugins

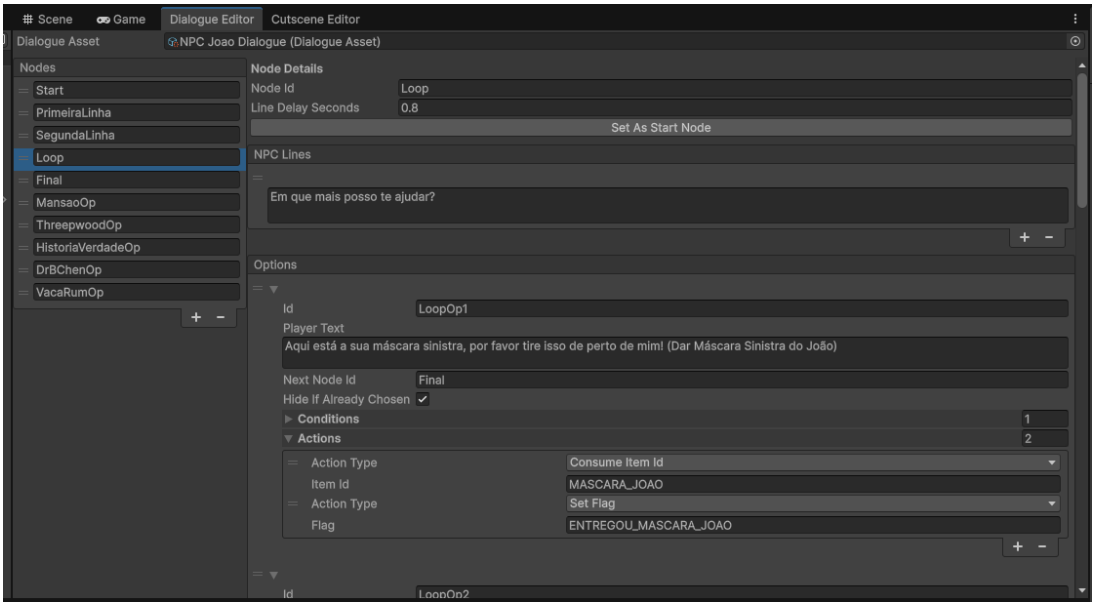
Um procedimento que não estava mapeado originalmente na metodologia, e que foi utilizado de forma pontual, foi a criação de *plugins* para o motor de jogo (*engine*) utilizando o ChatGPT Codex (OPENAI, 2026) conectado ao ambiente de desenvolvimento.

Os *plugins* são ferramentas personalizadas desenvolvidas pela IA-Gen para auxiliar no fluxo de criação da obra. Os dois principais *plugins* gerados para o desenvolvimento de *PinkBob J. e o Enigma do Rum* foram o editor de diálogos e o editor de cenas cinemáticas (*cutscenes*).

A Figura 19 apresenta um exemplo da interface contendo o *plugin* de edição de diálogos. Nela, é possível demonstrar as principais funcionalidades da ferramenta: a criação de nós de conversação (estabelecendo um fluxo lógico), a definição das falas do NPC e a estruturação de opções de escolhas para o jogador. Estas escolhas podem desencadear eventos, como a configuração de marcadores de estado (*flags*) para ativar comportamentos

específicos no jogo, como por exemplo a liberação de uma passagem, a entrega e consumo de itens em relação ao NPC, a inicialização de uma *cutscene* ou a transição para o próximo nó que dá continuidade à conversa.

Figura 19 – Exemplo de tela capturada da *engine* contendo o *plugin* de edição de diálogos.



Fonte: Elaborada pelo autor (2026).

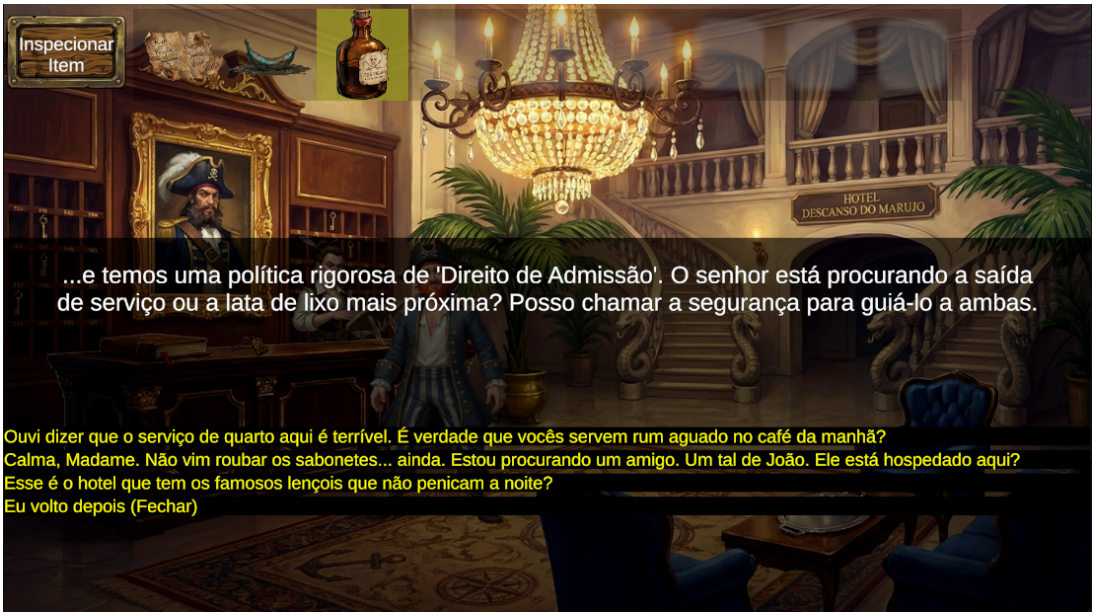
Para complementar a visualização do sistema de diálogos, a Figura 20 ilustra a interface de diálogos aplicada ao jogo, apresentando as falas do NPC e as respectivas opções de escolha disponibilizadas ao jogador.

Por sua vez, a Figura 21 ilustra um exemplo de captura de tela exibindo o *plugin* de edição de *cutscenes*. Diferentemente do sistema de diálogos, que possui nós ramificáveis em múltiplas opções, a cena cinemática ocorre de forma sequencial. Neste modo, o jogador não realiza escolhas diretas, atuando apenas como espectador de um trecho predefinido. O *plugin* possui a particularidade de integrar a movimentação dos personagens, ativando e desativando animações cronologicamente para construir uma experiência de observação cinematográfica semelhante a um filme.

4.4.4.5 Geração de áudios

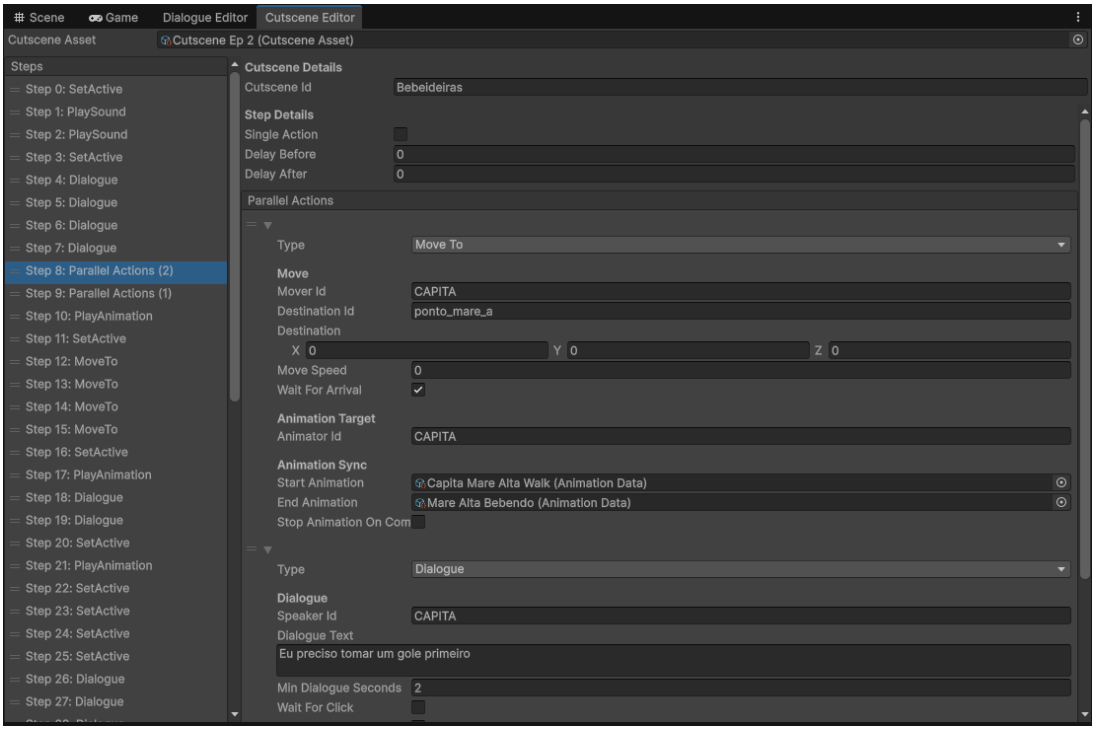
Para a geração das trilhas musicais e dos efeitos sonoros, a estrutura *story spine* foi submetida à ferramenta Google Gemini (GOOGLE, 2026a) com o objetivo de definir o tom adequado para as composições e para a ambientação, a partir do que o modelo gerou os *prompts*. Posteriormente, estes comandos foram aplicados na plataforma Suno (SUNO, 2025) para a criação das músicas e na ferramenta Kling AI (KLING AI, 2025) para a

Figura 20 – Exemplo de tela capturada do jogo contendo opções de diálogo.



Fonte: Elaborada pelo autor com auxílio de IA-Gen (2026).

Figura 21 – Exemplo de tela capturada da *engine* com o *plugin* de *cutscenes*.



Fonte: Elaborada pelo autor (2026).

síntese dos efeitos sonoros do jogo. Os áudios resultantes desta etapa estão disponíveis para consulta no repositório indicado no Apêndice A.

4.4.4.6 Considerações sobre o estudo de caso 1

A utilização das ferramentas de IA-Gen para o auxílio na criação do primeiro estudo de caso serviu como base fundamental para a aplicação nos projetos subsequentes. Este processo prático ajudou a refinar o uso destas tecnologias e a definir o melhor fluxo de trabalho para a geração dos *assets*, englobando a integração de novos recursos discutidos nas seções seguintes.

Para ilustrar a combinação de todos os *assets* produzidos ao longo destas etapas, a Figura 22 exemplifica uma cena completa e renderizada na *game engine*, integrando a arte do cenário, o NPC e o protagonista.

Figura 22 – Exemplo de tela capturada do jogo finalizado com elementos gerados por IA-Gen.



Fonte: Elaborada pelo autor com auxílio de IA-Gen (2026).

Ao todo, foram produzidos cerca de 130 *assets*. O projeto engloba 16 personagens, 20 *sprite sheets*, 15 móveis e itens, 45 cenários, nove músicas e nove efeitos sonoros. Todos os componentes gerados estão disponíveis no repositório indicado no Apêndice A.

O jogo *PinkBob J. e o Enigma do Rum* foi desenvolvido em sua totalidade. É possível encontrar um vídeo de demonstração, o *link* para *download* e as instruções de uso no repositório indicado no Apêndice A. As Figuras 23, 24, 25 e 26 retratam algumas das áreas finalizadas da obra, evidenciando a aplicação prática de todos os componentes gerados com o suporte da IA-Gen. O jogo foi apresentado ao público na feira de jogos da disciplina Design e Desenvolvimento de Jogos, em 23 de fevereiro de 2026. O retorno oral colhido na ocasião foi predominantemente favorável quanto à jogabilidade e à coesão narrativa, sem registro sistemático: nenhum dado dos visitantes foi registrado.

Pelos termos registrados nas Tabelas 9 e 10, a publicação comercial do jogo depende do áudio: o Suno só autoriza o uso comercial das músicas geradas durante uma assinatura paga, e o Kling AI, de termos de propriedade intelectual vagos, exige citação direta

indicando o uso de IA-Gen. As músicas do jogo foram geradas no plano gratuito do Suno e, para uma versão comercial, teriam de ser geradas novamente com assinatura ativa; os efeitos sonoros exigem a citação do Kling AI.

Figura 23 – Exemplo de tela capturada do jogo finalizado ilustrando o cenário do bar no primeiro episódio.



Fonte: Elaborada pelo autor com auxílio de IA-Gen (2026).

Figura 24 – Exemplo de tela capturada do jogo finalizado ilustrando o cenário do quarto de hotel no primeiro episódio.



Fonte: Elaborada pelo autor com auxílio de IA-Gen (2026).

4.5 Etapa 5: segundo estudo de caso — *Hacking Survivors*

Esta seção apresenta os resultados obtidos durante o desenvolvimento do segundo estudo de caso, denominado *Hacking Survivors*. O texto detalha o resultado da aplicação

Figura 25 – Exemplo de tela capturada do jogo finalizado ilustrando o cenário do porto no segundo episódio.



Fonte: Elaborada pelo autor com auxílio de IA-Gen (2026).

Figura 26 – Exemplo de tela capturada do jogo finalizado ilustrando o cenário do naufrágio no terceiro episódio.



Fonte: Elaborada pelo autor com auxílio de IA-Gen (2026).

das ferramentas de IA-Gen nas etapas de concepção do *design*, elaboração da documentação técnica, construção do protótipo e finalização do jogo completo.

4.5.1 Definição do estudo de caso e expansão da história

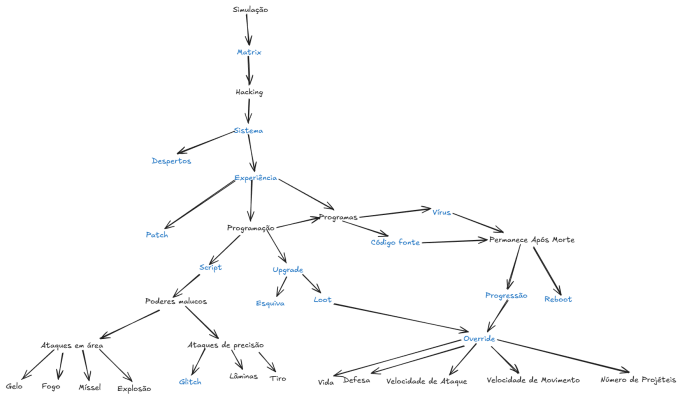
Com o intuito de demonstrar a aplicação da metodologia em diferentes gêneros de jogos, selecionou-se, a partir da lista original de cem conceitos, a primeira ideia catalogada, que não fazia parte das cinco propostas refinadas na Etapa 3:

> Um *hacker*, preso em uma simulação, precisa utilizar as suas habilidades para sobrecarregar o sistema a fim de causar a sua destruição. O jogador enfrentará milhares de programas inimigos até que o tempo termine.

A partir desta premissa, todas as etapas que anteriormente haviam sido executadas de forma manual foram realizadas de maneira direta com o auxílio dos modelos de IA-Gen selecionados. A ferramenta Google Gemini (GOOGLE, 2026a) foi utilizada para expandir o conceito sobre os três pilares fundamentais (*gameplay*, *storytelling* e *user experience*), recebendo como *prompt* a ideia original associada a algumas referências de obras semelhantes, como o jogo *Vampire Survivors* (PONCLE, 2022) e os filmes *Matrix* (WACHOWSKI; WACHOWSKI, 1999) e *Laranja Mecânica* (KUBRICK, 1971). O resultado desta expansão pode ser conferido no repositório indicado no Apêndice A.

Posteriormente, foi solicitado à IA-Gen que utilizasse este conceito expandido para realizar a listagem de palavras remetentes ao tema do jogo. Estes termos foram então combinados com uma lista do autor para a estruturação da árvore de ideias. Este processo de ideação, que no primeiro estudo de caso demandou a participação de voluntários, foi conduzido de forma inteiramente integrada com a IA-Gen, resultando nos diagramas apresentados nas Figuras 27, 28 e 29.

Figura 27 – Árvore de ideias do segundo estudo de caso, parte 1.

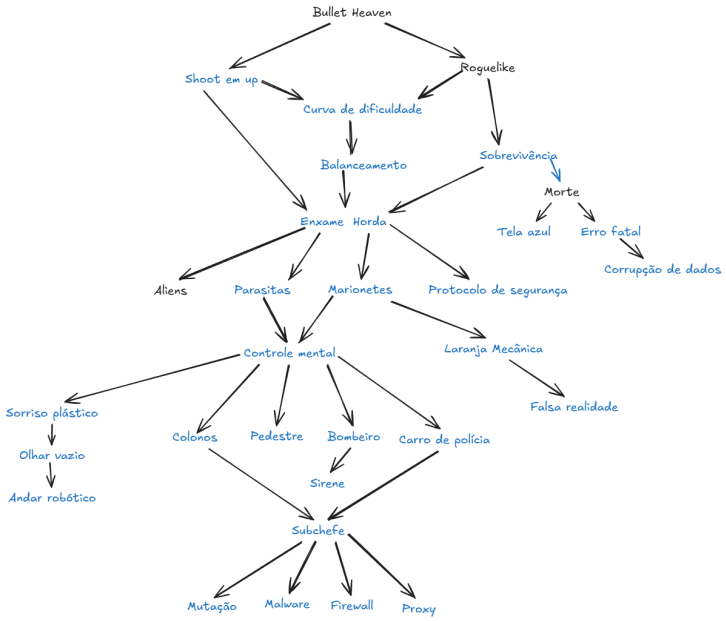


Fonte: Elaborada pelo autor com auxílio de IA-Gen (2026).

A árvore de ideias resultante foi enviada novamente ao Google Gemini (GOOGLE, 2026a), o qual foi instruído a preencher o documento de *design* do jogo utilizando os conceitos previamente estruturados. De forma resumida, os principais pontos estabelecidos foram:

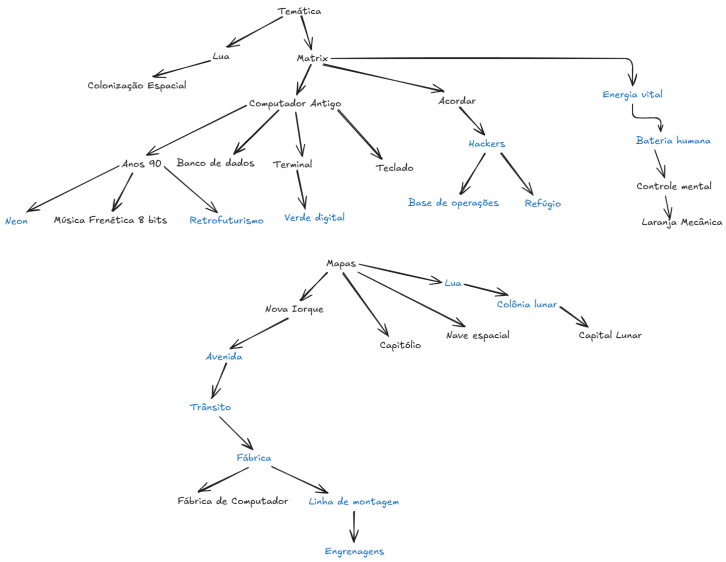
- **Tema**: Distopia, ciberespaço (*hacker*) e simulação.
- **Gênero**: *Shoot 'em up* com elementos de *roguelike*.

Figura 28 – Árvore de ideias do segundo estudo de caso, parte 2.



Fonte: Elaborada pelo autor com auxílio de IA-Gen (2026).

Figura 29 – Árvore de ideias do segundo estudo de caso, parte 3.



Fonte: Elaborada pelo autor com auxílio de IA-Gen (2026).

- **Pilares principais**: Escalada de poder (o jogador deve sentir-se como uma máquina imparável), caos controlado (gerenciamento dinâmico de espaço e posicionamento) e construção de combinações estratégicas (*builds*).
- **Estilo visual**: *Pixel art* em perspectiva superior bidimensional (*top-down* 2D). O

visual mistura o retrofuturismo com a estética de terminais de computadores antigos, com destaque para a cor verde digital.

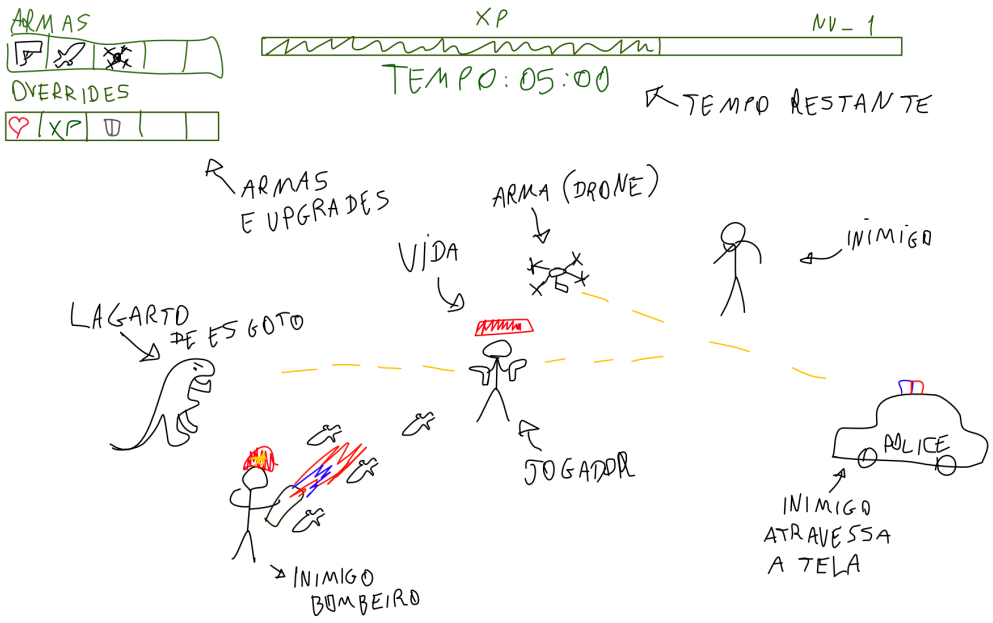
- **História**: No futuro, a colonização espacial na Lua terminou em tragédia. Uma raça de parasitas alienígenas subjugou a humanidade, aprisionando a mente dos colonos em uma simulação opressiva semelhante à Terra para drenar sua energia vital como baterias. O protagonista desperta dessa ilusão e descobre a habilidade de manipular o código da falsa realidade. Juntando forças com outros hackers em um refúgio escondido, eles iniciam invasões sistemáticas para corromper a Simulação alienígena e libertar os humanos adormecidos.
- **Audiência**: Fãs de jogos de sobrevivência de horda, otimização de *builds* e ficção científica.
- **Núcleo de *gameplay***: Entrar em um nível, eliminar inimigos, coletar pontos de experiência (XP), selecionar armamentos e melhorias (*upgrades*), sobrevivendo pelo maior tempo possível.
- **Principal Objetivo**: Sobreviver ao limite de tempo de cada fase e derrotar o chefe final.
- **Condições de vitória**: Sobreviver até o contador de tempo chegar a zero e eliminar o subchefe para progredir à próxima área.
- **Condições de derrota**: Ter a barra de vida do personagem reduzida a zero, resultando na reinicialização do ciclo da fase.

De maneira análoga ao procedimento adotado no primeiro estudo de caso, utilizou-se a árvore de ideias, o documento de *design* (GDD) e um rascunho de protótipo da tela do jogo (ilustrado na Figura 30) na ferramenta Claude 4.5 Sonnet (ANTHROPIC, 2025) para a elaboração do *story spine*. Uma versão resumida do enredo é apresentada a seguir.

No ano de 2157, a humanidade, após conquistar a Lua, foi aprisionada em uma simulação perfeita da Terra por seres invisíveis chamados Arquitetos, que drenam a energia vital dos colonos. Dentro desta falsa Nova York, o programador Rex começa a notar falhas na realidade e descobre ser capaz de manipular o código do mundo, tornando-se imediatamente alvo dos protocolos de contenção do sistema. Resgatado por Vera e levado à base secreta Nó Zero, Rex inicia uma invasão cibernética implacável pelas camadas do sistema para descobrir o paradeiro do *hardware* físico e derrubar a simulação. Ele derrota o FIREWALL_NYPD na camada de aplicação, desce à infraestrutura oculta da Fundação para destruir o GATEWAY e rastreia o fluxo de dados até o UPLINK, descobrindo por meio dos registros de transmissão (*logs*) que os servidores estão localizados na própria superfície lunar. Ao se materializar no mundo real na Lua, Rex enfrenta a

IA-Gen adaptativa ROOT e utiliza quatro fragmentos de código coletados em sua jornada para executar a sobreposição final (*Override*). Esta cartada desliga a matriz computacional, libertando dois milhões de humanos de seus casulos para reconstruírem a civilização. O evento consagra Rex como o Arquiteto Livre, ainda que fragmentos isolados do antigo sistema continuem adormecidos à espreita.

Figura 30 – Rascunho do protótipo desenvolvido para o segundo estudo de caso.



Fonte: Elaborada pelo autor (2026).

4.5.2 Desenvolvimento e gerações com IA-Gen

A principal diferença nesta etapa residiu no modelo utilizado para a geração dos comandos de entrada refinados. Com a adoção da ferramenta ChatGPT Codex (OPENAI, 2026), o processo de criação dos *prompts* empregados na geração de imagens e áudios foi totalmente integrado a este ambiente de desenvolvimento. Por meio da criação de um repositório local para o projeto, todos os artefatos gerados anteriormente (como o GDD, a imagem do protótipo de referência e o *story spine*) foram agrupados em um único diretório. Desta forma, a ferramenta tornou-se capaz de consultar automaticamente, e a qualquer momento, o material de referência necessário para o avanço contínuo do jogo.

Com o auxílio da ferramenta, a história foi dividida em quatro fases principais: Nova York, Esgotos, *Uplink* (uma base militar que abriga uma antena de transmissão de dados) e Colônia Lunar. Estas fases, os seus respectivos inimigos e as peculiaridades de cada uma foram descritos em um documento (GDD) extenso gerado pela IA-Gen. O texto detalhou minuciosamente cada aspecto do escopo, abrangendo também a sugestão para o acréscimo de novas armas, personagens jogáveis e mecânicas específicas por nível. Cabe ressaltar que

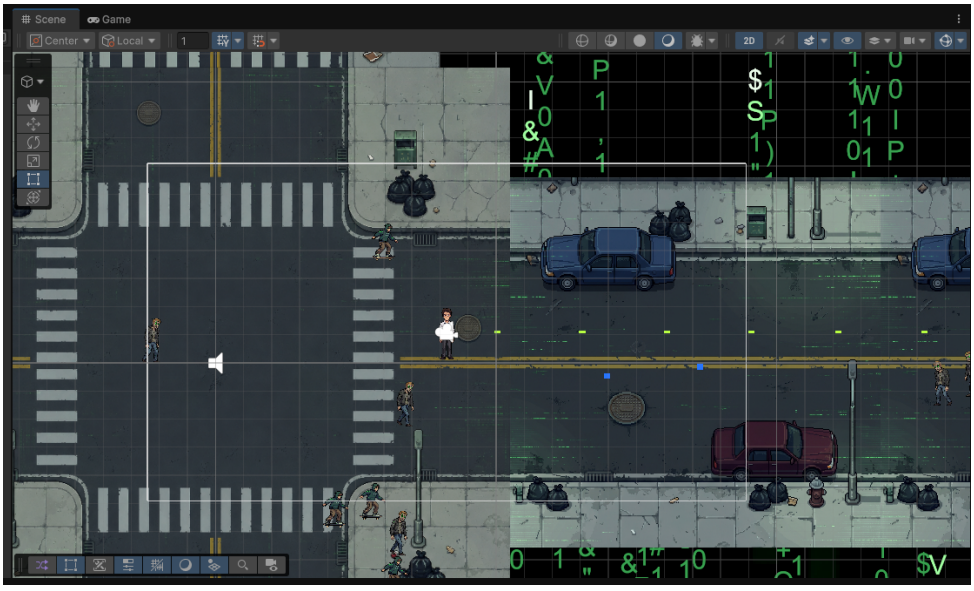
alguns destes elementos adicionais não foram incluídos na versão final da obra devido a delimitações de tempo e escopo. As fases foram implementadas de maneira sequencial, garantindo que os *assets* necessários para a construção dos cenários, inimigos, jogador e dos armamentos fossem gerados sob demanda ao longo do processo de desenvolvimento.

4.5.2.1 Geração de cenários

A primeira fase do jogo foi elaborada de forma similar ao primeiro estudo de caso. No entanto, o modelo gerador de imagens da plataforma Flow Studio demonstrou dificuldade em gerar cenários em perspectiva superior (*top-down*) completos com todos os elementos integrados. Diferentemente do projeto anterior, no qual cada cena era carregada de maneira isolada, o segundo estudo de caso utiliza uma abordagem baseada em blocos pré-montados (*tiles*). Estes blocos precisam conectar-se de forma contínua uns aos outros com o objetivo de gerar os níveis do jogo de forma procedural.

Consequentemente, o primeiro nível apresentou uma desconexão visual nas emendas entre os blocos. Esta anomalia foi posteriormente justificada como um elemento narrativo, sendo interpretada pelo enredo como uma falha aparente na matriz da simulação em que o protagonista se encontra. A Figura 31 ilustra o ponto de união entre dois *tiles* que não se alinharam de forma fluida, evidenciando a limitação computacional do modelo em gerar imagens distintas contendo bordas contínuas.

Figura 31 – Exemplo de *tiles* que não se conectam.



Fonte: Elaborada pelo autor com auxílio de IA-Gen (2026).

Para solucionar este obstáculo nas fases posteriores, adotou-se o método de geração individualizada de cada objeto componente dos cenários. Desta forma, tornou-se possível

construir os *tiles* manualmente na *engine*, utilizando os *assets* isolados fornecidos pela IA-Gen. Esta abordagem permitiu o ajuste preciso para que o encaixe das extremidades pudesse ser alcançado.

Este fluxo de trabalho fragmentado foi replicado para a montagem de todos os *tiles* correspondentes aos níveis dois, três e quatro. A Figura 32 apresenta exemplos de objetos isolados criados pela ferramenta, enquanto a Figura 33 demonstra a combinação destes elementos na composição de um *tile* capaz de conectar-se com as peças adjacentes.

Figura 32 – Exemplo de objetos gerados para construção de *tiles* da fase 3.



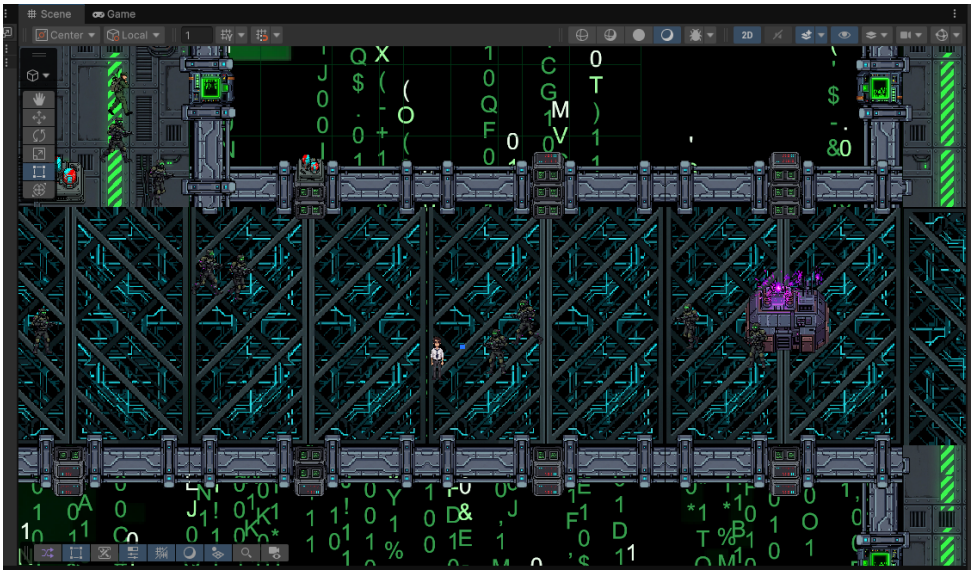
Fonte: Elaborada pelo autor com auxílio de IA-Gen (2026).

4.5.2.2 Geração de personagens e armamentos

O processo de geração dos personagens ocorreu de maneira análoga ao do primeiro estudo de caso. A única distinção foi no modelo gerador dos *prompts*, sendo utilizada a ferramenta ChatGPT Codex (OPENAI, 2026) para este fim. Diferentemente do projeto anterior, este jogo possui armamentos que podem ser obtidos pelo jogador. Os itens foram concebidos seguindo o mesmo fluxo empregado na criação da arte dos personagens, conforme detalhado nas etapas anteriores. Como exemplo de resultado desta etapa, a Figura 34 ilustra a animação da arma denominada *"Armadilha Zero-Day"*, uma espécie de mina terrestre configurada para detonar mediante o contato com os inimigos.

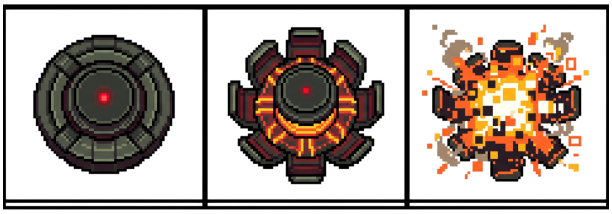
Para ilustrar a concepção dos personagens, a Figura 35 exemplifica a geração do inimigo "Executivo Infectado", pertencente à primeira fase do jogo. Trata-se de um transeunte do cenário de Nova York que foi corrompido pelo sistema da simulação para combater o protagonista. Todas as animações desenvolvidas, abrangendo tanto os oponentes

Figura 33 – Exemplo de *tiles* da fase 3 criados usando os objetos gerados por IA-Gen.



Fonte: Elaborada pelo autor com auxílio de IA-Gen (2026).

Figura 34 – Exemplo de *sprite sheet* para a arma “Armadilha Zero-Day”.



Fonte: Elaborada pelo autor com auxílio de IA-Gen (2026).

quanto o arsenal, seguiram uma padronização de três quadros (*frames*) na composição da *sprite sheet* para a ilusão de movimento.

4.5.2.3 Geração de quadrinhos

Outro aspecto que apresentou divergência em relação ao primeiro estudo de caso correspondeu à forma de apresentação da narrativa. Em vez de utilizar cenas cinemáticas (*cutscenes*) para exibir a movimentação e os diálogos dos personagens, o segundo projeto fez alusão direta ao formato das histórias em quadrinhos. Antes do início de cada nível, o jogo apresenta uma sequência de páginas estáticas que narram os acontecimentos. Esta abordagem foi adotada com o intuito de demonstrar a diversidade de possibilidades que as

Figura 35 – Exemplo de *sprite sheet* para o inimigo “Executivo Infectado” da primeira fase.



Fonte: Elaborada pelo autor com auxílio de IA-Gen (2026).

ferramentas de IA-Gen oferecem ao processo de *design* de jogos.

A Figura 36 ilustra esta aplicação, exibindo um trecho narrativo introduzido no começo da quarta fase do jogo. Neste modelo de exibição, caixas de texto conduzem a narrativa enquanto a imagem gerada pela IA-Gen atua como suporte visual para ilustrar a situação, emulando a estética tradicional de uma revista em quadrinhos.

Figura 36 – Exemplo de um trecho da história em quadrinhos apresentada entre fases do jogo.



Fonte: Elaborada pelo autor com auxílio de IA-Gen (2026).

4.5.2.4 Considerações sobre o estudo de caso 2

O restante dos processos de geração ocorreu de forma análoga ao estudo de caso inicial, com a distinção de que a ferramenta ChatGPT Codex (OPENAI, 2026) atuou

como o modelo central para o refinamento dos *prompts*. Os *assets* resultantes das gerações podem ser consultados no repositório indicado no Apêndice A.

Um ponto relevante a ser mencionado ocorreu durante a estruturação da quarta fase do jogo. O emprego da ferramenta ChatGPT Codex (OPENAI, 2026) atingiu a sua cota mensal de utilização restrita ao plano gratuito, culminando na migração compulsória do projeto para o sistema Google Antigravity (GOOGLE, 2026b), em sua versão 2.0, um aplicativo agêntico independente do editor de código, lançado publicamente em maio de 2026, durante o mesmo período de desenvolvimento. Até aquele momento, o Antigravity estava disponível apenas como ambiente de programação, com os agentes integrados ao editor.

O formato agêntico tornou possível uma manipulação mais profunda dos arquivos locais do jogo. Este evento demonstrou-se altamente benéfico para o andamento da pesquisa, visto que a nova arquitetura integrou os recursos do modelo Google Gemini (GOOGLE, 2026a) à capacidade de gerações de imagens do Nano Banana Pro, disponível no próprio Antigravity. Esta convergência ecossistêmica permitiu a geração de *assets* finalizados por meio de uma única interface, unificando modelos de alta competência em um fluxo de desenvolvimento contínuo diretamente no repositório do projeto. A solução mostrou-se útil na estruturação do projeto e serviu de base para o próximo estudo de caso.

Ao todo, foram produzidos mais de 140 *assets*. O projeto engloba 55 inimigos diferentes acompanhados de suas respectivas *sprite sheets*, um personagem jogável, cerca de 30 objetos utilizados para a construção dos cenários, três músicas, 11 efeitos sonoros, 20 artes utilizadas na elaboração da história em quadrinhos e três armas com animações únicas.

O jogo *Hacking Survivors* foi totalmente finalizado. É possível encontrar um vídeo de demonstração, o *link* para *download* e as instruções de uso no repositório indicado no Apêndice A. As Figuras 37, 38, 39 e 40 demonstram algumas das áreas finalizadas do jogo, evidenciando a aplicação prática de todos os recursos concebidos com o suporte da IA-Gen.

A publicação comercial deste jogo segue os mesmos termos do primeiro caso, já que as músicas e os efeitos sonoros vieram do Suno e do Kling AI. As músicas foram geradas no plano gratuito do Suno e, para uma versão comercial, teriam de ser geradas novamente com assinatura ativa; os efeitos sonoros exigem a citação do Kling AI.

4.6 Etapa 6: terceiro estudo de caso — *Pafúncio vs. Desenhos*

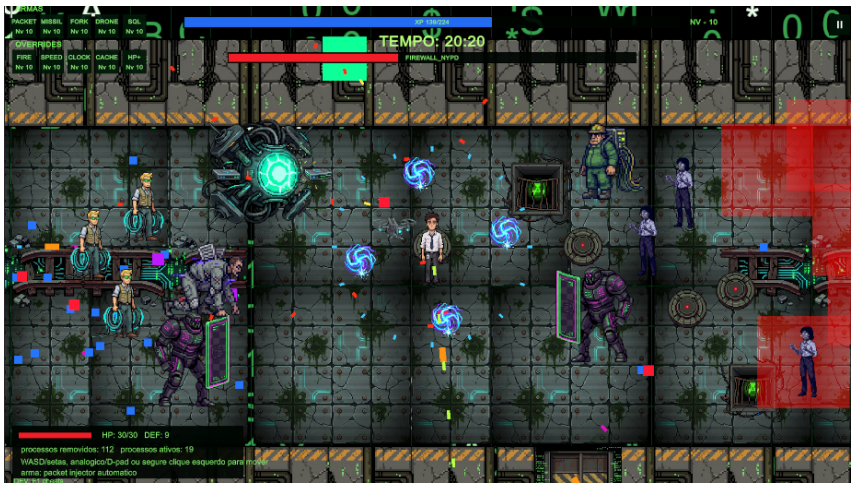
Esta seção apresenta os resultados obtidos no desenvolvimento do terceiro estudo de caso, denominado *Pafúncio vs. Desenhos*. O texto detalha o resultado da aplicação da

Figura 37 – Exemplo de tela capturada do jogo finalizado ilustrando a fase 1.



Fonte: Elaborada pelo autor com auxílio de IA-Gen (2026).

Figura 38 – Exemplo de tela capturada do jogo finalizado ilustrando a fase 2.



Fonte: Elaborada pelo autor com auxílio de IA-Gen (2026).

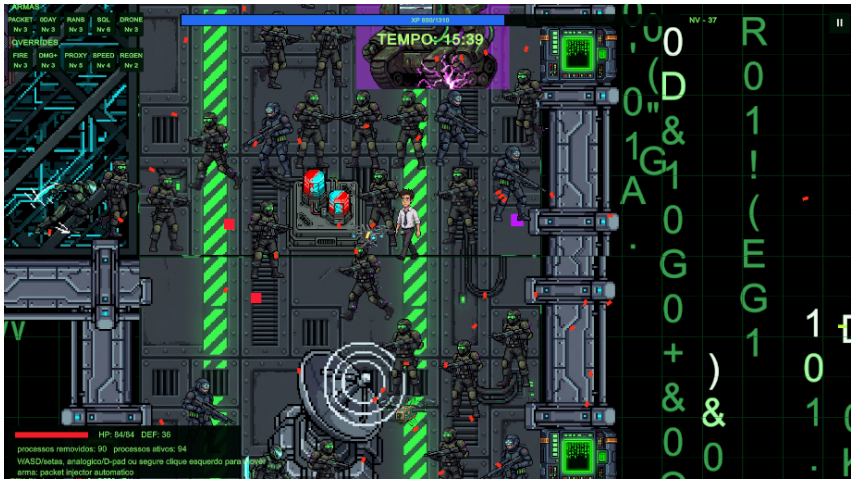
metodologia de uso de ferramentas de IA-Gen para a concepção, elaboração do *design*, estruturação de documentos e construção do jogo.

4.6.1 Definição do estudo de caso e expansão da história

Com a intenção de demonstrar a versatilidade da metodologia em diferentes gêneros de jogos, selecionou-se, a partir da lista original de cem conceitos, a seguinte ideia voltada ao gênero de plataforma, também fora das cinco propostas refinadas na Etapa 3:

Desenhos dominaram o mundo, e agora eles estão saindo do papel, a única

Figura 39 – Exemplo de tela capturada do jogo finalizado ilustrando a fase 3.



Fonte: Elaborada pelo autor com auxílio de IA-Gen (2026).

Figura 40 – Exemplo de tela capturada do jogo finalizado ilustrando a fase 4.



Fonte: Elaborada pelo autor com auxílio de IA-Gen (2026).

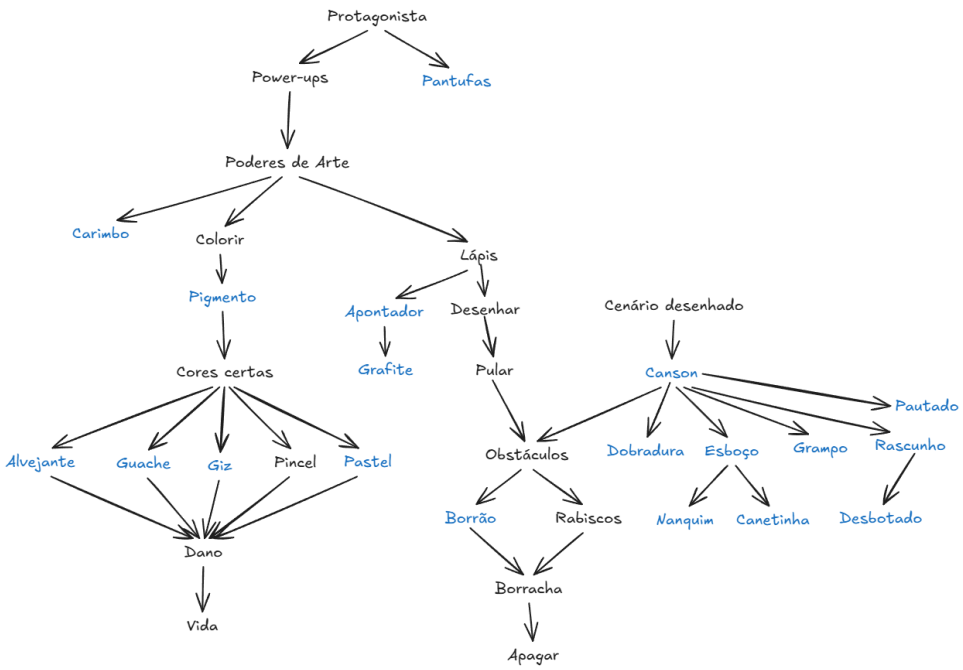
forma de controlá-los é pintando todos. O jogador enfrenta diversos desenhos e precisa pintar para eliminá-los.

Com a premissa escolhida, as mesmas etapas de refinamento foram executadas de maneira direta com o auxílio das IA-Gen selecionadas anteriormente. A ferramenta Google Gemini (GOOGLE, 2026a) expandiu o conceito da obra sobre os três pilares fundamentais (*gameplay*, *storytelling* e *user experience*), recebendo como base a ideia original juntamente com algumas referências, como por exemplo os jogos *Super Mario Bros.* (NINTENDO, 1985) e *Scribblenauts* (5TH CELL, 2009). A intenção era gerar um jogo de plataforma que combinasse elementos de um estojo de desenhos em suas mecânicas principais, permitindo

ações como atirar bolas de tinta, desenhar plataformas sólidas e apagar rabiscos.

O conceito expandido foi utilizado em conjunto com o Google Gemini (GOOGLE, 2026a) para estruturar a árvore de ideias, resultando no diagrama apresentado na Figura 41.

Figura 41 – Árvore de ideias do terceiro estudo de caso.



Fonte: Elaborada pelo autor com auxílio de IA-Gen (2026).

Em seguida, o GDD foi elaborado, estabelecendo os seguintes pontos principais:

- **Tema**: Criatividade, transformação e pintura.
- **Gênero**: Plataforma 2D de rolagem lateral (*side-scroller*) e quebra-cabeças de ação (*puzzle-action*).
- **Pilares principais**: Física da arte (manipulação de materiais de papelaria), mecânicas de pintar, riscar e apagar. Os inimigos derrotados não morrem, mas sim adquirem cor e são pacificados.
- **Estilo visual**: *Papercraft* animado 2D. Os cenários simulam texturas de papéis reais, folhas de caderno pautado e rabiscos. Os elementos hostis e cenográficos são compostos por desenhos que necessitam ser coloridos.
- **História**: Pafúncio é um ex-entusiasta da pintura que trocou os seus livros de colorir pelas obrigações da vida adulta. Após desejar voltar a ter tempo para a arte, ele acorda calçando pantufas no interior de seu antigo livro de desenhos. Neste universo,

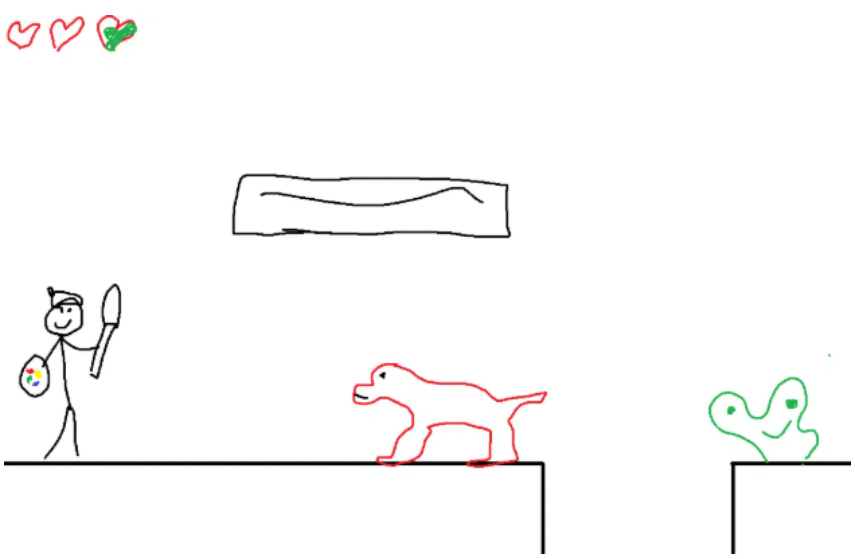
ele descobre que as obras deixadas pela metade ganharam vida e estão revoltadas com o abandono. Guiado por uma borracha falante, Pafúncio precisa usar o seu antigo estojo para colorir os rascunhos, redimir as suas criações e encontrar a saída antes que seja completamente apagado.

- **Audiência**: Famílias, jogadores casuais e fãs de plataformas expressivas.
- **Núcleo de *gameplay***: Identificar a cor do traço dos inimigos e pintá-los com as cores correspondentes, desenhar plataformas para superar obstáculos e percorrer o cenário até o final da fase.
- **Principal Objetivo**: Atravessar as seções do livro de desenhos, colorindo os oponentes e preenchendo as partes em branco do cenário.
- **Condições de vitória**: Alcançar a margem direita da página (linha de chegada da fase).
- **Condições de derrota**: Acumular três danos recebidos de inimigos, perdendo progressivamente as suas próprias cores.

Utilizando estes artefatos estruturais, somados a um rascunho de protótipo desenhado (ilustrado na Figura 42), a ferramenta Claude 4.5 Sonnet (ANTHROPIC, 2025) foi empregada para gerar a versão final do *story spine* do jogo. Uma versão resumida do enredo é apresentada a seguir:

A história acompanha Pafúncio, um homem de meia-idade que substituiu sua antiga paixão pela arte por uma rotina monótona e cinzenta. Ao reencontrar seu antigo caderno de desenhos, ele é magicamente sugado para dentro de suas páginas, acordando em um mundo de papel onde os rascunhos inacabados que ele abandonou ganharam vida e, movidos pela mágoa, tentam apagar sua cor até transformá-lo em um borrão sem forma. Guiado por Apagadinha, uma pequena borracha viva que foi sua única criação finalizada, Pafúncio embarca em uma jornada pelas diferentes seções do caderno, atravessando as áreas de frágeis rascunhos a lápis, obstáculos encharcados de tinta e inimigos de cores incompletas, usando seu velho estojo de artes para desenhar plataformas, apagar perigos e colorir as criaturas para pacificá-las. No clímax da aventura, ele enfrenta O Grande Borrão, uma monstruosidade disforme criada pela soma de todos os seus abandonos, e descobre que a verdadeira forma de vencê-lo não é apagando-o, mas sim finalizando o desenho com intenção e cor. Ao concluir sua grande obra e devolver a vida e a vibração ao caderno inteiro, Pafúncio retorna ao mundo real profundamente transformado, pronto para abrir novas páginas em branco e resgatar definitivamente o seu amor pela arte.

Figura 42 – Protótipo desenhado do jogo *Pafúncio vs. Desenhos*.



Fonte: Elaborada pelo autor (2026).

4.6.2 Desenvolvimento e gerações com IA-Gen

Com a consolidação da ferramenta Google Antigravity (GOOGLE, 2026b), o processo que outrora envolvia criar um *prompt* em uma plataforma, gerar a imagem em outra e inseri-la manualmente no projeto foi significativamente simplificado. Agora, todos os artefatos estruturados nas etapas anteriores foram organizados em um diretório unificado dentro do repositório de desenvolvimento, permitindo que o agente consultasse os materiais do jogo a qualquer momento. Desta forma, com um comando direto na interface, os *prompts* foram processados, as imagens geradas e imediatamente importadas para dentro da *game engine*, utilizando um fluxo unificado.

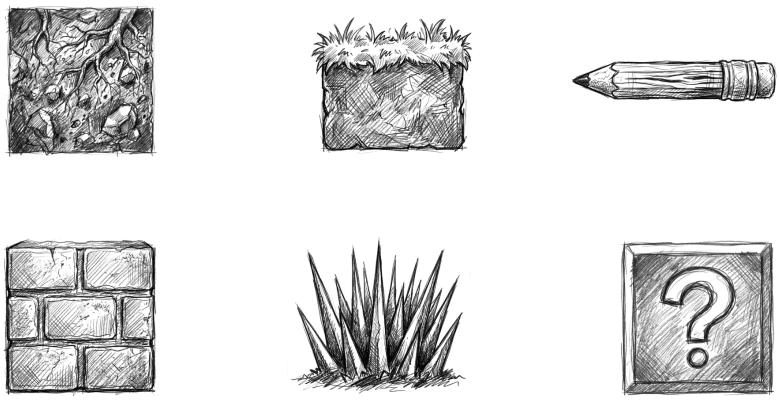
A única etapa que permaneceu dependente de plataformas externas foi a geração das músicas e dos efeitos sonoros. Neste caso, o Google Antigravity foi utilizado para estruturar os *prompts*, enquanto as ferramentas especializadas ficaram encarregadas de gerar os áudios. Por este motivo, esta seção se concentra na geração dos cenários, a qual representou o principal diferencial em relação aos estudos de caso anteriores. Todas as gerações realizadas e os seus respectivos *prompts* podem ser encontrados no repositório indicado no Apêndice A.

4.6.2.1 Geração de cenários

Para demonstrar a versatilidade do uso das ferramentas de IA-Gen no *design* do jogo, a concepção das fases adotou uma abordagem focada na criação de uma paleta de blocos modulares, os quais poderiam ser utilizados para a construção dos níveis.

Com o modelo agêntico, solicitou-se que a IA-Gen utilizasse os materiais de referência para construir blocos compatíveis com a ferramenta de *tile palette* da *engine* de jogo. Como resultado, foram elaborados os blocos fundamentais correspondentes a tijolo, espinhos, bloco de interrogação, terra, terra com grama e plataforma de lápis, os quais podem ser observados na Figura 43.

Figura 43 – Blocos gerados por IA-Gen utilizados na construção das fases.



Fonte: Elaborada pelo autor com auxílio de IA-Gen (2026).

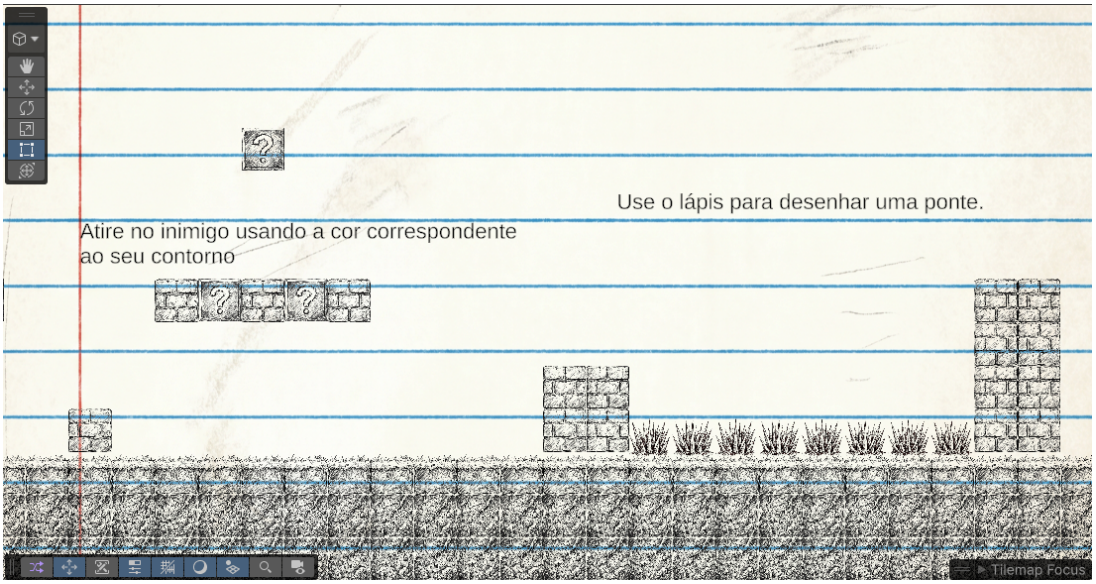
Posteriormente, estes blocos foram combinados a uma imagem de fundo, também gerada pelo Google Antigravity, que simula a textura de uma folha pautada. Esta composição visual teve o objetivo de criar a ilusão de que a fase do jogo havia sido desenhada em um caderno. O resultado prático dessa técnica pode ser observado na Figura 44.

4.6.2.2 Geração de personagens

Seguindo o mesmo princípio adotado para a geração dos blocos de construção dos cenários, solicitou-se que a IA-Gen utilizasse o GDD e o *story spine* como base para conceber a arte conceitual do personagem principal, o Pafúncio. Em seguida, por meio da retroalimentação desta primeira imagem, o modelo foi instruído a produzir uma sequência contendo três quadros para representar o movimento de caminhada do personagem, além de um quadro adicional ilustrando a ação de pular. Todas estas imagens, cinco ao todo, foram combinadas para compor o arquivo final representado na Figura 45, o qual foi importado para a *engine* a fim de viabilizar a animação do personagem nos estados ocioso, caminhando e pulando; o quadro do estado ocioso é a própria arte conceitual.

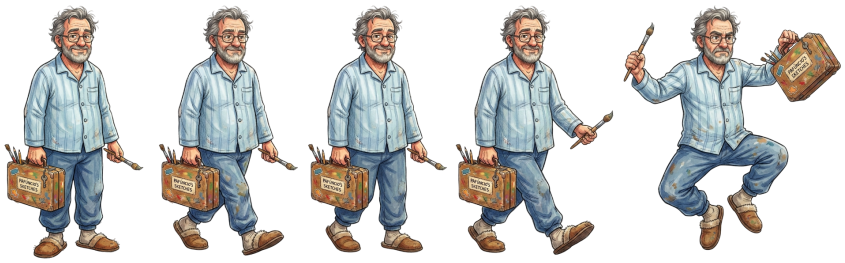
Um processo semelhante foi conduzido para a criação dos inimigos do jogo. A diferença fundamental reside na justificativa narrativa da obra: os antagonistas são rascunhos inacabados do protagonista e precisam ser coloridos para serem pacificados. Por este

Figura 44 – Exemplo de cena construída utilizando os blocos gerados por IA-Gen.



Fonte: Elaborada pelo autor com auxílio de IA-Gen (2026).

Figura 45 – Exemplo da *sprite sheet* finalizada do personagem Pafúncio.



Fonte: Elaborada pelo autor com auxílio de IA-Gen (2026).

motivo, a imagem que contém os quadros de animação apresenta os inimigos inicialmente desenhados a traço em uma só cor (vermelho, no caso do cachorro), com um estilo visual de rascunho, possuindo também uma arte final colorida que é ativada somente após o jogador aplicar sobre o inimigo a tinta da mesma cor do traço. A Figura 46 exemplifica esta mecânica aplicada ao inimigo cachorro raivoso.

Figura 46 – Exemplo da *sprite sheet* finalizada do inimigo cachorro raivoso.



Fonte: Elaborada pelo autor com auxílio de IA-Gen (2026).

4.6.2.3 Geração de elementos de interface

Os elementos de interface do jogo também fazem alusão às mecânicas centrais de pintura e descoloração. Alinhada a esta ideia, a barra de vida do personagem foi concebida na forma de três camisas coloridas. Sempre que o jogador sofre algum dano, uma destas vestimentas perde a sua coloração. Caso as três peças fiquem totalmente descoloridas, o estado de derrota é acionado e o jogador deve reiniciar a fase.

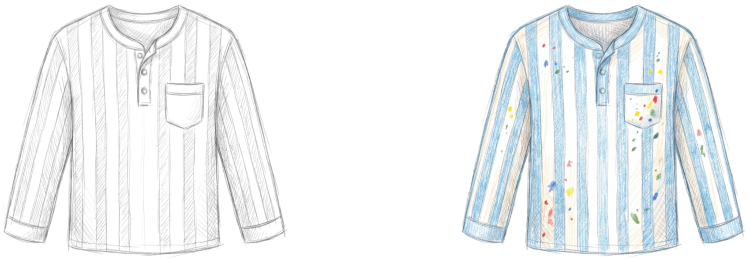
Com o intuito de traduzir esta mecânica para um formato visual prático, solicitou-se à IA-Gen a geração da imagem de uma camisa desenhada em preto e branco e a versão colorida da mesma peça. Estas artes foram integradas à interface de usuário para indicar de forma intuitiva a quantidade de vidas restantes durante a partida. A Figura 47 ilustra o resultado das gerações correspondentes a este componente.

4.6.2.4 Considerações sobre o estudo de caso 3

O restante dos processos de geração ocorreu de forma semelhante aos demais estudos de caso, destacando-se apenas que, para as imagens, a ferramenta Google Antigravity executou todo o fluxo, desde a estruturação do *prompt* até a geração da arte final. Os *assets* resultantes destas gerações podem ser consultados no repositório indicado no Apêndice A.

Cabe ressaltar que, devido ao prazo estipulado para a conclusão deste trabalho, o terceiro estudo de caso não foi integralmente finalizado. Ele foi concebido como uma prova de conceito com o objetivo de verificar a aplicação da metodologia na geração dos *assets* de um jogo de plataforma e a sua adaptabilidade a um estilo gráfico diferente, marcado pela estética de desenhos rascunhados a lápis em um caderno. Por este motivo, a implementação final do jogo limitou-se a três fases, as quais são suficientes para demonstrar a aplicação

Figura 47 – Exemplo das camisas geradas para representar a vida do jogador.



Fonte: Elaborada pelo autor com auxílio de IA-Gen (2026).

prática dos *assets* gerados operando de forma integrada com as principais mecânicas concebidas. É possível encontrar um vídeo de demonstração da versão prova de conceito, o *link* para *download* e as instruções de uso no repositório indicado no Apêndice A.

Como nos casos anteriores, as músicas e os efeitos sonoros vieram do Suno e do Kling AI, e uma versão comercial do jogo seguiria os mesmos termos, as músicas foram geradas no plano gratuito do Suno e teriam de ser geradas novamente com assinatura ativa, e os efeitos sonoros exigem a citação do Kling AI.

Para ilustrar o resultado visual desta implementação, as Figuras 48, 49, 50 e 51 apresentam, respectivamente, a tela inicial do menu, a interface de seleção de fases e dois trechos práticos de jogabilidade evidenciando os componentes gerados operando de forma conjunta.

Figura 48 – Exemplo de tela capturada do jogo ilustrando o menu principal, em que o título aparece na grafia estilizada da arte (“Pafuncio Vs Desenhos”).



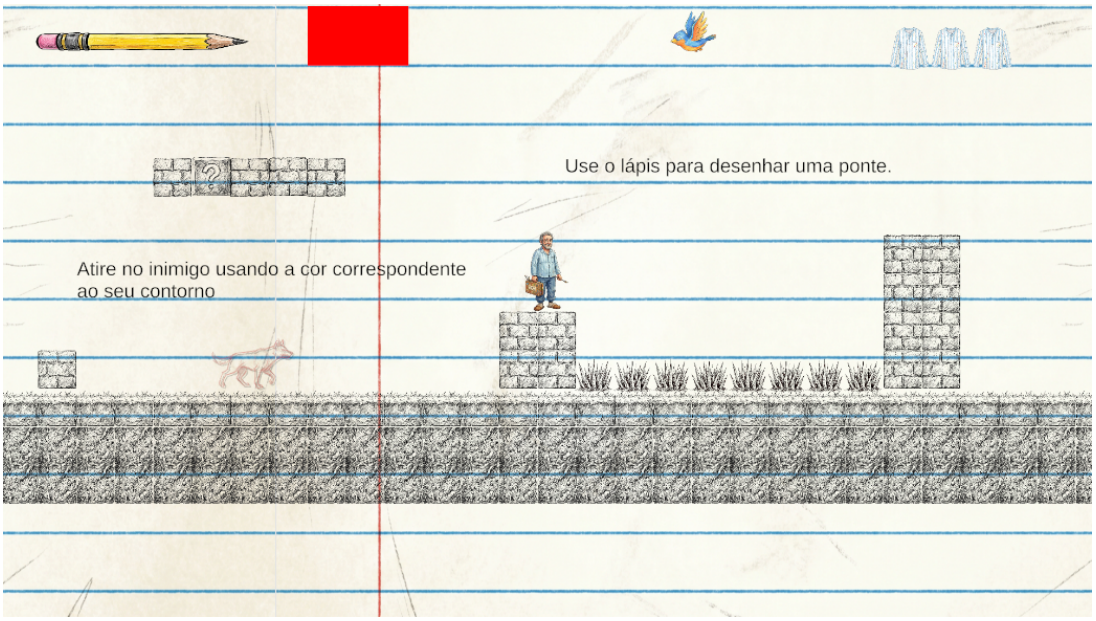
Fonte: Elaborada pelo autor com auxílio de IA-Gen (2026).

Figura 49 – Exemplo de tela capturada do jogo ilustrando a interface de seleção de fases.



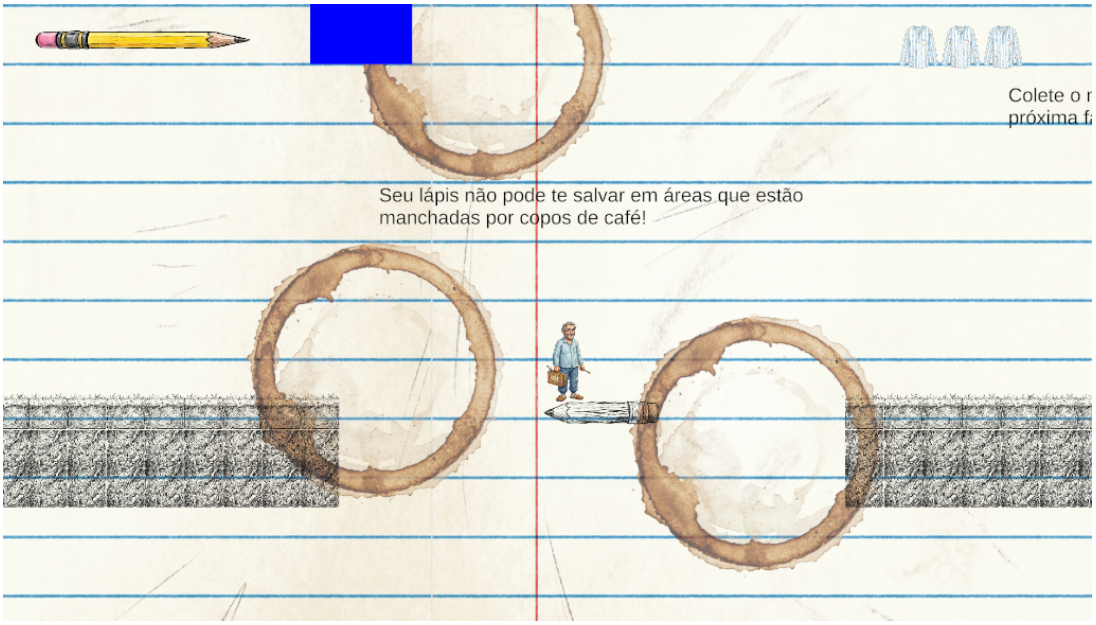
Fonte: Elaborada pelo autor com auxílio de IA-Gen (2026).

Figura 50 – Exemplo de tela capturada do jogo ilustrando as plataformas geradas por IA-Gen na construção da fase 1.



Fonte: Elaborada pelo autor com auxílio de IA-Gen (2026).

Figura 51 – Exemplo de tela capturada do jogo ilustrando as manchas de café e outros elementos gerados por IA-Gen na fase 1.



Fonte: Elaborada pelo autor com auxílio de IA-Gen (2026).

4.7 Síntese dos estudos de caso

Os três estudos de caso aplicaram o método a gêneros e estilos gráficos diferentes (uma aventura *point and click*, um *shoot 'em up* com elementos de *roguelike* e um jogo de plataforma 2D) e cada um acrescentou algo ao fluxo de produção. O primeiro, *PinkBob J. e o Enigma do Rum*, fixou a sequência de trabalho que os outros dois seguiram, da ideação e do documento de *design* aos protótipos e à geração dos ativos, com o modelo de linguagem redigindo os *prompts* para as ferramentas de imagem e de áudio; nele surgiram também os dois *plugins* de editor gerados sob demanda, e foi o único jogo apresentado ao público. O segundo, *Hacking Survivors*, passou à IA-Gen a *story spine*, escrita à mão no primeiro caso, reuniu o material de referência em um repositório local, consultado pelo ChatGPT Codex na redação dos *prompts*, e expôs um limite do modelo de imagem: os cenários em perspectiva superior não se encaixavam em blocos contínuos, e a saída foi gerar os objetos isoladamente e montar os *tiles* na *engine*. O terceiro, *Pafúncio vs. Desenhos*, concentrou no Google Antigravity todo o fluxo das imagens, do *prompt* à importação no projeto, e testou o método em um estilo gráfico distinto, o de desenhos a lápis em um caderno.

Juntos, os três casos mostram o mesmo método aplicado a projetos de gêneros e estilos diferentes e revisado de um caso para o outro. O que mais mudou foi o encadeamento das ferramentas: no primeiro caso, cada ativo passava por plataformas separadas, o modelo de linguagem para o *prompt* e as ferramentas de imagem e de áudio para a geração; no terceiro, um agente dentro do repositório do projeto reuniu o *prompt* e a imagem, e só o áudio continuou em ferramentas externas. Os dois primeiros jogos foram concluídos, e o terceiro chegou a uma versão de demonstração com três fases. O único retorno de jogadores foi o oral, colhido na feira do primeiro caso, sem registro sistemático.

Quanto à programação, a montagem dos três jogos na *game engine* foi trabalho do autor, como define a Fase 5 do método (Seção 3.1). No *PinkBob J. e o Enigma do Rum*, o autor programou as mecânicas de deslocamento, interação com itens, inventário e interface do jogo, e a IA-Gen gerou em código os dois plugins de editor, de diálogos e de cutscenes, com o ChatGPT Codex. No *Hacking Survivors*, o ChatGPT Codex fez toda a programação de mecânicas de movimento, armas, inimigos e menus, a partir da quarta fase do jogo, o Google Antigravity escreveu no repositório o *script* utilizado para construção de tiles e *cutscenes*; os tiles foram montados à mão na engine. No *Pafúncio vs. Desenhos*, o Google Antigravity, além de gerar e importar as imagens, escreveu todo o código de movimentação, controle dos inimigos, e mecânica de tiro e desenho com tinta, o autor fez o posicionamento dos objetos nas fases.

Com esses resultados, o Capítulo 5 discute o alcance e as limitações do método.

5 Considerações finais

5.1 Sumarização da pesquisa e resultados

O *design* de jogos é uma área de grande abrangência, englobando a aplicação de técnicas de ideação, concepção, criação de histórias, elaboração de personagens, artes, trilhas e efeitos sonoros. Esta pluralidade exige experiência em diversas áreas para a geração de todos os componentes necessários para se construir um jogo. Esta complexidade pode ser considerada uma barreira de entrada significativa para pequenos desenvolvedores independentes, uma vez que, para atingir um alto padrão de qualidade, é necessário o domínio de múltiplas especializações ou a contratação de uma equipe multidisciplinar, o que frequentemente é inviabilizado por restrições de orçamento (ADAMS, 2014).

Diante deste cenário, este trabalho partiu da seguinte questão: de que forma e com quais ferramentas a IA-Gen pode apoiar desenvolvedores independentes na produção de ativos multimídia e na documentação de jogos? Para respondê-la, foi utilizada uma abordagem exploratória e prática, combinando diversas técnicas e modelos de IA-Gen (como os LLMs e os modelos multimodais). O objetivo principal consistiu em desenvolver e documentar uma metodologia de apoio ao *design* e à produção de jogos digitais baseada em ferramentas de IA-Gen, voltada ao desenvolvimento independente. O resultado é o método descrito na Seção 3.1, que organiza a produção em cinco fases (concepção, planejamento da produção, seleção das ferramentas, geração dos ativos por *meta-prompting* e montagem na *game engine*), com refinamento iterativo dentro de cada uma. A resposta à questão, portanto, é que a IA-Gen pode apoiar a concepção, a documentação e a geração dos ativos de texto, imagem e áudio de um jogo, com as ferramentas descritas a seguir, e que a montagem permanece com o desenvolvedor.

Na comparação das ferramentas (Seção 4.2), o resultado obtido na seleção foi: o Claude 4.5 Sonnet para a estrutura narrativa (*story spine*), o Google Gemini 3 Pro para a expansão da ideia e o documento de *design*, o Flow Studio Nano Banana Pro para imagens, o Suno para trilhas sonoras e o Kling AI para efeitos sonoros. Nas imagens, a escolha não seguiu só a nota: o Gemini Nano Banana Pro teve a maior pontuação da Tabela 7, e o Flow Studio foi preferido por ampliar as imagens sem custo adicional e pelos recursos de edição. A geração de vídeo foi testada, mas não seguiu para a produção principalmente pela instabilidade na permanência de objetos e pelo custo por geração (Seção 4.2.4).

O critério de execução adotado na Seção 3.1 (o jogo concebido, documentado e produzido com os ativos gerados pelas ferramentas de IA-Gen selecionadas, com os ajustes que o projeto exigiu registrados) foi atendido nos três estudos de caso. *PinkBob J. e o*

Enigma do Rum e *Hacking Survivors* chegaram à versão final, cada um com mais de uma centena de ativos gerados, e *Pafúncio vs. Desenhos*, a uma versão de demonstração, na qual o critério vale para as fases do jogo que o prazo permitiu produzir. A montagem e a programação na *game engine* permaneceram trabalho autoral do desenvolvedor, com uso pontual de IA-Gen, como nos *plugins* de editor do primeiro caso (Seção 4.4.4.4). Os três jogos mostram, assim, como um único desenvolvedor independente pode articular ferramentas de modalidades diferentes ao longo das etapas de concepção e de produção, com os desafios de cada combinação registrados no Capítulo 4.

5.2 Limitações da pesquisa

Apesar dos resultados práticos na estruturação do método, a pesquisa tem limitações que devem ser consideradas.

A primeira lacuna reside na ausência de uma medição quantitativa de tempo. Por não ter sido traçado um comparativo rigoroso e cronometrado entre o método de desenvolvimento tradicional e o fluxo assistido proposto, não é possível afirmar, com rigor científico, que a IA-Gen acelerou o processo de desenvolvimento, limitando-se a classificá-la como uma ferramenta de suporte funcional.

A segunda lacuna está no instrumento de avaliação das imagens e dos vídeos. As notas que ordenam as ferramentas nas Tabelas 7 e 12 foram atribuídas exclusivamente pelo autor, com conhecimento da ferramenta de origem de cada amostra e sem replicação das avaliações (Seção 3.3), e os pesos das métricas das Tabelas 2 e 3 não foram testados contra uma ponderação uniforme. Os rankings são, portanto, a leitura de um avaliador, e não uma medida de consenso.

Os resultados das comparações valem, além disso, para as versões testadas entre novembro de 2025 e janeiro de 2026. O progresso das ferramentas de IA-Gen é contínuo e rápido: durante a própria pesquisa, surgiram soluções integradas que mudaram o fluxo de trabalho, como a versão agêntica do Google Antigravity (GOOGLE, 2026b), que combinou as capacidades do Google Gemini (GOOGLE, 2026a) e do Nano Banana Pro (GOOGLE, 2025a) em um único ecossistema, com manipulação direta dos arquivos do jogo (Seção 4.5). É por essa razão que a Fase 3 do método (Seção 3.1) recomenda refazer a comparação com os lançamentos mais recentes.

Outra restrição significativa é que a avaliação cobriu os ativos, e não os jogos. As métricas das Tabelas 2 e 3 medem atributos de cada ativo isolado, e nenhum dos três jogos passou por avaliação estruturada de jogabilidade, ausência de defeitos, diversão ou qualidade percebida. Também não se mediu como os jogadores percebem a estética característica das imagens geradas por IA-Gen, que as métricas de nitidez e de alinhamento de estilo não capturam. O único projeto apresentado a jogadores foi *PinkBob J. e o Enigma*

do Rum, na feira de jogos organizada pela disciplina Design e Desenvolvimento de Jogos da universidade, em 23 de fevereiro de 2026 (Seção 4.4.4.6). O retorno colhido na ocasião foi oral e sem registro sistemático, e uma amostra isolada como essa não substitui um protocolo de *playtest* aplicado aos três estudos de caso.

Há também uma limitação de alcance. Os três casos são jogos bidimensionais de escopo curto, feitos por um único desenvolvedor, que é o cenário que o método pressupõe (Seção 3.1); o trabalho não permite dizer se ele se sustenta em projetos de outro porte, como jogos de mundo aberto ou com modo multijogador, nem com uma equipe, ainda que reduzida, nem nas mãos de outros desenvolvedores.

No âmbito gerencial, o trabalho não avalia a viabilidade orçamentária dos modelos de IA-Gen para produções de longo prazo. O custo financeiro contínuo exigido para a manutenção de assinaturas e a aquisição de pacotes de moedas virtuais (*tokens* de geração) representa uma despesa recorrente, o que pode restringir ou inviabilizar as operações de pequenos criadores. Nesta pesquisa, o uso das ferramentas custou um valor total de 0 reais (devido a ofertas para estudantes no serviço do *Gemini Pro* do *Google*) ao longo de 10 meses. Somado a isso, os erros nativos dos modelos de IA-Gen testados e as constantes alucinações visuais e sonoras não foram só defeitos dos ativos: foram custo de processo, porque cada geração falha precisou ser corrigida ou refeita, o que consome recursos e pode atrasar os cronogramas de produção em vez de otimizá-los.

Metodologicamente, a geração de códigos de programação não foi explorada com o mesmo rigor técnico-científico aplicado à mídia visual, sonora e textual, caracterizando uma limitação analítica no desenvolvimento deste trabalho.

Outra limitação que merece destaque é a decisão de não aprofundar as complexas questões de direitos autorais referentes aos conteúdos gerados por IA-Gen, apresentadas na Seção 2.3. Como este é um campo legislativo em constante e rápida transformação, qualquer análise definitiva neste trabalho correria o risco de ficar obsoleta em um curto espaço de tempo. A consequência prática, porém, pode ser dita para os três jogos: pelos termos registrados nas Tabelas 4, 6, 9 e 10, os textos e as imagens gerados ficam com o usuário, condicionados aos termos de serviço de cada plataforma, mas as trilhas do Suno só podem ser comercializadas se geradas com assinatura ativa, e os efeitos sonoros do Kling AI estão sujeitos a termos de propriedade intelectual vagos, que exigem citação direta indicando o uso de IA-Gen. A publicação comercial dos jogos depende, portanto, de como cada ativo foi gerado. Nos três jogos, as músicas foram geradas no plano gratuito do Suno e, para uma versão comercial, teriam de ser geradas novamente com assinatura ativa.

Também é importante ressaltar que, devido ao escopo de tempo delimitado para a entrega do trabalho, o último estudo de caso não foi finalizado em sua totalidade. Apenas as fases do jogo essenciais para a demonstração prática da metodologia foram desenvolvidas, e o jogo chegou a uma versão de demonstração.

5.3 Impacto para a área de jogos e IA-Gen

As contribuições deste estudo refletem-se na demonstração de um uso concreto de IA-Gen no desenvolvimento independente de jogos. Ao organizar a geração dos ativos pela técnica do *meta-prompting*, em que um modelo de linguagem escreve o *prompt* que a ferramenta geradora executa, a pesquisa demonstra uma das formas como a barreira técnica da criação artística pode ser mitigada, como mostram os três estudos de caso (Seções 4.4 a 4.6).

Para a área de Inteligência Artificial Generativa, o trabalho explora o uso dos modelos multimodais e dos LLMs para atuar em conjunto com diretrizes de *game design* (como o preenchimento coerente de um GDD e a criação da história e dos personagens). O impacto central é a demonstração de uma abordagem na qual a IA-Gen não substitui a figura do *game designer*, mas potencializa a sua capacidade de curadoria e montagem (Seções 4.4.4 e 4.5.2), possibilitando que projetos de um único desenvolvedor ganhem materialidade visual e narrativa sem uma equipe de arte e de áudio.

5.4 Implicações e trabalhos futuros

A superação das limitações detectadas neste documento abre perspectivas valiosas para a continuidade acadêmica e prática na área.

Como sugestão principal de estudos vindouros, a exploração metodológica aprofundada da geração de lógicas e códigos de programação pode estender o apoio da IA-Gen à etapa que, neste trabalho, ficou com o desenvolvedor: a integração entre os componentes de jogo, a orquestração do código-fonte e os testes do artefato. A integração de ferramentas de geração de modelos tridimensionais (3D) permitiria, por sua vez, levar o método à produção de ativos para jogos tridimensionais.

Recomenda-se também uma pesquisa quantitativa com diferentes grupos de jogadores para avaliar os jogos desenvolvidos com o auxílio da IA-Gen, visando deixar evidentes os impactos dessa abordagem no público final ao qual o jogo é destinado, bem como suas opiniões e impressões sobre o uso desta tecnologia.

O passo seguinte do próprio método é a sua aplicação por outros desenvolvedores, que não o autor, para verificar se ele se sustenta fora das mãos de quem o criou e em projetos de outro porte. Registrar o tempo gasto em cada fase nessas aplicações daria também a linha de base que faltou a este trabalho.

Sugere-se ainda uma pesquisa mais aprofundada sobre os direitos autorais das gerações feitas pelas IA-Gens, que debata criticamente o risco legal envolvido, as regras das plataformas em detalhes, a legislação vigente e como possíveis alterações podem impactar um produto já finalizado, além do risco de similaridade com obras que podem

ter servido como treinamento para os modelos (que muitas vezes não são de código aberto e a informação não está disponível publicamente).

Por fim, a geração de vídeo, testada na Seção 4.2.4 e deixada fora da produção pelas razões registradas lá, pode ser retomada à medida que os modelos amadurecerem, junto com as animações em formato GIF, que ficaram fora antes dos testes. No entanto, essa expansão exige acesso a recursos consideráveis, seja de poder computacional robusto para execução dos modelos de IA-Gens de vídeo de código aberto (*open source*), seja financeiro para custear assinaturas de modelos em nuvem.

Referências

5TH CELL. *Scribblenauts* [Jogo eletrônico]. [S.l.]: Warner Bros. Interactive Entertainment, 2009. Citado na página 84.

ACE. *ACE Step Model* [Modelo de geração de áudio]. 2025. Disponível em: <https://comfyanonymous.github.io/ComfyUI_examples/audio/>. Acesso em: 12 nov. 2025. Citado na página 48.

ADAMS, E. *Fundamentals of Game Design*. 3. ed. Berkeley, CA: New Riders, 2014. ISBN 9780321929679. Citado 5 vezes nas páginas 18, 19, 25, 29 e 95.

ADAMS, K. *How to Improvise a Full-Length Play: The Art of Spontaneous Theater*. New York: Allworth Press, 2007. Citado na página 29.

ADOBE. *Adobe Firefly* [Modelo de geração de efeitos sonoros]. 2025. Disponível em: <https://firefly.adobe.com/generate/sound-effects>. Acesso em: 13 nov. 2025. Citado na página 50.

AL-SURAYHI, M. M. et al. Generative artificial intelligence in game design: A narrative review. *Journal of Ecohumanism*, v. 4, n. 4, p. 1702–1712, 2025. Disponível em: <https://doi.org/10.62754/joe.v4i4.6912>. Acesso em: 15 jun. 2026. Citado 2 vezes nas páginas 15 e 19.

AMRESH, A. Integrating reinforcement AI into the design of educational games. In: *Proceedings of the 17th European Conference on Games Based Learning, ECGBL 2023*. Germany: Dechema e.V., 2023. p. 13–18. Disponível em: <https://papers.academic-conferences.org/index.php/ecgbl/article/view/1709>. Acesso em: 21 jul. 2026. Citado 2 vezes nas páginas 23 e 24.

ANTHROPIC. *Claude (Versão 4.5 Sonnet)* [Modelo de linguagem grande]. 2025. Disponível em: <https://claude.ai>. Acesso em: 06 jan. 2026. Citado 6 vezes nas páginas 15, 37, 41, 59, 76 e 86.

AUTOMATIC1111. *Stable Diffusion WebUI* [Interface de geração de imagens]. 2025. Disponível em: <https://github.com/automatic1111/stable-diffusion-webui>. Acesso em: 12 nov. 2025. Citado na página 43.

BANH, L.; STROBEL, G. Generative artificial intelligence. *Electronic Markets*, v. 33, n. 1, p. 63, 2023. Disponível em: <https://doi.org/10.1007/s12525-023-00680-1>. Acesso em: 20 jul. 2026. Citado 2 vezes nas páginas 19 e 20.

BOONSTRA, L. *Prompt engineering*. 2025. Disponível em: <https://www.kaggle.com/whitepaper-prompt-engineering>. Acesso em: 31 jul. 2026. Citado 2 vezes nas páginas 20 e 21.

BRASIL. Conselho Nacional de Saúde. *Resolução nº 510, de 7 de abril de 2016*. 2016. Diário Oficial da União: seção 1, Brasília, DF, n. 98, p. 44, 24 maio 2016. Disponível em: <https://www.in.gov.br/materia/-/asset_publisher/Kujrw0TZC2Mb/content/id/22917581>. Acesso em: 28 set. 2026. Citado na página 33.

CALHABEU, G.; LIMA, T. Design e desenvolvimento de um jogo independente - relato de experiência em um contexto acadêmico. In: *Anais do XXIV Simpósio Brasileiro de Jogos e Entretenimento Digital (SBGames)*. Salvador, BA, Brasil: SBC, 2025. Disponível em: <https://doi.org/10.5753/sbgames.2025.10288>. Acesso em: 30 jul. 2026. Citado 2 vezes nas páginas 18 e 19.

CARNEIRO, M. F.; CHAI, C. G. Padrões éticos e aspectos legais sobre direitos autorais e plágio: aproximações teóricas e metodológicas. *Revista do SELL*, v. 14, n. 1, p. 310–328, 2025. Disponível em: <https://doi.org/10.18554/rs.v14i1.8429>. Acesso em: 03 ago. 2026. Citado 2 vezes nas páginas 21 e 22.

CIVITAI. *CHEYENNE* [Modelo de geração de imagem]. 2025a. Disponível em: <https://civitai.com/models/198051/cheyenne>. Acesso em: 17 dez. 2025. Citado 3 vezes nas páginas 43, 46 e 48.

CIVITAI. *Stable Diffusion: Juggernaut XL* [Modelo de geração de imagem]. 2025b. Disponível em: <https://civitai.com/models/133005/juggernaut-xl>. Acesso em: 12 dez. 2025. Citado 3 vezes nas páginas 43, 46 e 48.

CNN BRASIL. *Entenda a polêmica envolvendo IA e o Studio Ghibli*. 2025. Disponível em: <https://www.cnnbrasil.com.br/tecnologia/entenda-a-polemica-envolvendo-ia-e-o-studio-ghibli/>. Acesso em: 03 ago. 2026. Citado na página 22.

COMFYUI. *ComfyUI* [Interface baseada em nós para geração de imagens]. 2025a. Disponível em: <https://comfy.org/>. Acesso em: 12 nov. 2025. Citado na página 43.

COMFYUI. *Wan2.2 5B* [Modelo de geração de vídeo]. 2025b. Disponível em: <https://docs.comfy.org/tutorials/video/wan/wan2_2>. Acesso em: 24 nov. 2025. Citado na página 51.

DUONGVE. *NetaYume Lumina (ComfyUI)* [Modelo de geração de imagem]. 2025. Disponível em: <https://huggingface.co/duongve/NetaYume-Lumina-Image-2.0>. Acesso em: 12 dez. 2025. Citado 2 vezes nas páginas 43 e 46.

ELASRI, M. et al. Image generation: A review. *Neural Processing Letters*, v. 54, p. 4609–4646, 2022. Disponível em: <https://doi.org/10.1007/s11063-022-10777-x>. Acesso em: 31 jul. 2026. Citado 2 vezes nas páginas 30 e 31.

ELEVENLABS. *ElevenLabs Music* [Modelo de geração de áudio e música]. 2025. Disponível em: <https://elevenlabs.io/app/music>. Acesso em: 12 nov. 2025. Citado 2 vezes nas páginas 48 e 50.

FIGUEIREDO, G. R. *Repositorio-Geracoes-tcc* [Repositório de dados da pesquisa]. 2026. Disponível em: <https://github.com/Uelfol/Repositorio-Geracoes-tcc>. Acesso em: 29 set. 2026. Citado na página 107.

GALLOTTA, R. et al. Large language models and games: A survey and roadmap. *arXiv preprint arXiv:2402.18659*, 2024. Disponível em: <https://arxiv.org/abs/2402.18659>. Acesso em: 12 jun. 2026. Citado 3 vezes nas páginas 15, 22 e 24.

GOOGLE. *Flow Studio Nano banana pro* [Modelo de geração de imagem]. 2025a. Disponível em: <https://labs.google/flow/about>. Acesso em: 24 nov. 2025. Citado 8 vezes nas páginas 43, 45, 46, 48, 51, 63, 66 e 96.

GOOGLE. *Flow Studio Veo 3.1 Fast* [Modelo de geração de vídeo]. 2025b. Disponível em: <https://labs.google/flow/about>. Acesso em: 24 nov. 2025. Citado na página 51.

GOOGLE. *Gemini (Versão 3.1 Pro)* [Modelo de linguagem multimodal]. 2026a. Disponível em: <https://gemini.google.com>. Acesso em: 20 fev. 2026. Citado 17 vezes nas páginas 15, 20, 37, 41, 43, 45, 46, 48, 51, 63, 66, 69, 74, 82, 84, 85 e 96.

GOOGLE. *Google Antigravity* [Ferramenta agêntica de inteligência artificial]. 2026b. Disponível em: <https://antigravity.google/>. Acesso em: 26 mai. 2026. Citado 5 vezes nas páginas 28, 35, 82, 87 e 96.

HAILUO AI. *Hailuo AI* [Ferramenta de geração de vídeo]. 2025. Disponível em: <https://hailuoai.video/create/image-to-video>. Acesso em: 24 nov. 2025. Citado na página 51.

HAO, Y. et al. Optimizing prompts for text-to-image generation. In: *Advances in Neural Information Processing Systems 36 (NeurIPS 2023)*. New Orleans, LA, USA: Curran Associates, Inc., 2023. Disponível em: <https://arxiv.org/abs/2212.09611>. Acesso em: 19 set. 2026. Citado na página 21.

HARTWIG, S. et al. *A Survey on Quality Metrics for Text-to-Image Generation*. 2025. Disponível em: <https://arxiv.org/abs/2403.11821>. Acesso em: 31 jul. 2026. Citado na página 30.

KLING AI. *Kling AI* [Modelo de geração de efeitos sonoros, áudio, imagens e vídeo]. 2025. Disponível em: <https://app.klingai.com/global/text-to-audio/new>. Acesso em: 13 nov. 2025. Citado 5 vezes nas páginas 20, 43, 50, 51 e 69.

KUBRICK, S. *A Clockwork Orange (Laranja Mecânica)* [Filme]. [S.l.]: Warner Bros., 1971. Citado na página 74.

KULTIMA, A. Trends in gaming - views of industry professionals and game scholars. In: *Proceedings of the 20th International Conference on the Foundations of Digital Games (FDG '25)*. New York, NY, USA: Association for Computing Machinery, 2025. Disponível em: <https://doi.org/10.1145/3723498.3723743>. Citado 2 vezes nas páginas 19 e 21.

LAMBE, I. *The Surprising NEW Number of GenAI Games on Steam*. 2025. Disponível em: <https://www.totallyhuman.io/blog/the-surprising-new-number-of-genai-games-on-steam>. Acesso em: 22 fev. 2026. Citado na página 21.

LIU, Y. et al. *Beyond Prompt Content: Enhancing LLM Performance via Content-Format Integrated Prompt Optimization*. 2025. Disponível em: <https://arxiv.org/abs/2502.04295>. Acesso em: 31 jul. 2026. Citado 2 vezes nas páginas 20 e 33.

LUCASARTS. *Indiana Jones and the Fate of Atlantis* [Jogo eletrônico]. [S.l.]: LucasArts, 1992. Citado na página 58.

LUCASARTS. *Full Throttle* [Jogo eletrônico]. [S.l.]: LucasArts, 1995. Citado na página 58.

LUCASFILM GAMES. *The Secret of Monkey Island* [Jogo eletrônico]. [S.l.]: Lucasfilm Games, 1990. Citado 3 vezes nas páginas 38, 58 e 63.

LUGOSSIAN. *GDD 1-Page Template Download*. 2026. Disponível em: <https://www.lugossian.com/gdd-1-page-template-download>. Acesso em: 30 jul. 2026. Citado 2 vezes nas páginas 39 e 41.

MOJANG STUDIOS. *Minecraft* [Jogo eletrônico]. 2011. Disponível em: <https://www.minecraft.net/pt-br>. Acesso em: 21 jul. 2026. Citado na página 23.

MOTTA, R. L.; TRIGUEIRO JUNIOR, J. Short game design document (SGDD): Documento de game design aplicado a jogos de pequeno porte e advergames — um estudo de caso do advergame rockergirl bikeway. In: SBC. *Proceedings of SBGames 2013*. São Paulo, 2013. Disponível em: <https://www.sbgames.org/sbgames2013/proceedings/artedesign/15-dt-paper_SGDD.pdf>. Acesso em: 30 jul. 2026. Citado 2 vezes nas páginas 39 e 41.

NINTENDO. *Super Mario Bros.* [Jogo eletrônico]. [S.l.]: Nintendo, 1985. Citado na página 84.

OPENAI. *ChatGPT (Versão GPT-5.2)* [Modelo de linguagem grande]. 2025. Disponível em: <https://openai.com/pt-BR/index/introducing-gpt-5-2/>. Acesso em: 06 jan. 2026. Citado 4 vezes nas páginas 15, 20, 37 e 41.

OPENAI. *ChatGPT Codex* [Ferramenta de inteligência artificial]. 2026. Disponível em: <https://chatgpt.com/pt-BR/codex/>. Acesso em: 15 abr. 2026. Citado 5 vezes nas páginas 68, 77, 79, 81 e 82.

OPENART. *OpenArt* [Plataforma de geração de vídeo e imagem]. 2025. Disponível em: <https://openart.ai/help>. Acesso em: 24 nov. 2025. Citado na página 51.

OSBORN, A. F. *Applied Imagination: principles and procedures of creative thinking*. New York: Charles Scribner's Sons, 1953. Citado na página 32.

PANCHANADIKAR, R.; HU, Y.; FREEMAN, G. A dual role collision: How generative AI's intertwining productivity support and social support reshape indie game developers' creative work. In: *Proceedings of the 2026 ACM Conference on Fairness, Accountability, and Transparency*. New York, NY, USA: Association for Computing Machinery, 2026. p. 1098–1119. Disponível em: <https://dl.acm.org/doi/10.1145/3805689.3812248>. Acesso em: 30 jul. 2026. Citado na página 15.

PARK, J. S. et al. *Generative Agents: Interactive Simulacra of Human Behavior*. 2023. Disponível em: <https://arxiv.org/abs/2304.03442>. Acesso em: 06 out. 2025. Citado 2 vezes nas páginas 22 e 24.

PEDERSEN, R. E. *Game Design Foundations*. Plano, TX: Wordware Pub., 2003. (Wordware Game and Graphics Library). Disponível em: <https://books.google.com.br/books?id=DvYmFr_uaWwC>. Acesso em: 20 jul. 2026. Citado na página 18.

PENG, X. et al. Player-driven emergence in LLM-driven game narrative. In: *2024 IEEE Conference on Games (CoG)*. Milan, Italy: IEEE, 2024. p. 150–153. Disponível em: <https://ieeexplore.ieee.org/document/10645607>. Acesso em: 21 jul. 2026. Citado 2 vezes nas páginas 23 e 24.

PONCLE. *Vampire Survivors* [Jogo eletrônico]. [S.l.]: poncle, 2022. Citado na página 74.

RUNWAY. *Runway* [Ferramenta de geração de vídeo]. 2025. Disponível em: <https://app.runwayml.com>. Acesso em: 24 nov. 2025. Citado na página 51.

SANTOS, D. de V.; PENDIUK, F. O dilema da originalidade: inteligência artificial e direitos autorais no cenário artístico. *ÂNIMA - Revista eletrônica do curso de Direito da UniOpet*, v. 32, n. 1, jun. 2026. Disponível em: <https://periodicos.opet.com.br/index.php/anima/article/view/159>. Acesso em: 03 ago. 2026. Citado 2 vezes nas páginas 21 e 22.

SERRA, C. B.; SERRA, G. M. B.; CLASSE, T. M. de. Digital game development using large language models (LLMs): An exploratory study. In: *Anais do XXIV Simpósio Brasileiro de Jogos e Entretenimento Digital (SBGames)*. Salvador, BA, Brasil: SBC, 2025. Disponível em: <https://sol.sbc.org.br/index.php/sbgames/article/view/37365/37147>. Acesso em: 30 jul. 2026. Citado 2 vezes nas páginas 23 e 24.

SFX ENGINE. *SFX Engine* [Ferramenta de geração de efeitos sonoros]. 2025. Disponível em: <https://sfxengine.com/pt/app/sound-effects>. Acesso em: 13 nov. 2025. Citado na página 50.

STABILITY AI. *Stable Audio Open 1.0* [Modelo de geração de áudio]. 2025. Disponível em: <https://comfyanonymous.github.io/ComfyUI_examples/audio/>. Acesso em: 12 nov. 2025. Citado 4 vezes nas páginas 48, 49, 50 e 51.

SUNO. *Suno* [Modelo de geração de áudio e música]. 2025. Disponível em: <http://suno.com/>. Acesso em: 12 nov. 2025. Citado 4 vezes nas páginas 20, 48, 49 e 69.

SUZGUN, M.; KALAI, A. T. *Meta-Prompting: Enhancing Language Models with Task-Agnostic Scaffolding*. 2024. Disponível em: <https://arxiv.org/abs/2401.12954>. Acesso em: 19 set. 2026. Citado na página 21.

SWEETSER, P. Large language models and video games: A preliminary scoping review. *arXiv preprint arXiv:2403.02613*, 2024. Disponível em: <https://doi.org/10.48550/arXiv.2403.02613>. Acesso em: 12 jul. 2026. Citado 3 vezes nas páginas 19, 22 e 24.

TERNAR, A. et al. Generative AI in game development: A qualitative research synthesis. *arXiv preprint arXiv:2509.11898*, 2025. Disponível em: <https://arxiv.org/abs/2509.11898>. Acesso em: 20 nov. 2025. Citado 2 vezes nas páginas 15 e 21.

UBISOFT LA FORGE. *Generative Base Material: An Open-Source Prototype for PBR Material Estimation Debuting at SIGGRAPH Asia 2025*. 2025. Disponível em: <https://www.ubisoft.com/en-us/studio/laforge/news/1i3YOvQX2iArLlScBPqBZs/generative-base-material-an-opensource-prototype-for-pbr-material-estimation-debuting-at-siggraph-asia-2025>. Acesso em: 14 fev. 2026. Citado na página 15.

UDIO. *Udio* [Modelo de geração de áudio e música]. 2025. Disponível em: <https://www.udio.com/>. Acesso em: 12 nov. 2025. Citado 2 vezes nas páginas 48 e 49.

UNITY TECHNOLOGIES. *Unity (Versão 6000.3.8f1)* [Motor de jogo]. 2026. Disponível em: <https://unity.com/releases/editor/whats-new/6000.3.8f1>. Acesso em: 20 nov. 2025. Citado 2 vezes nas páginas 33 e 54.

UNIVERSIDADE FEDERAL DE OURO PRETO. Conselho Superior de Pesquisa e Pós-Graduação. *Resolução CONPEP n.º 191, de 15 de junho de 2026*. Estabelece normas e diretrizes para o uso de Inteligência Artificial Generativa (IAGen) no ensino e na pesquisa na pós-graduação da UFOP e revoga a Resolução CONPEP n.º 144/2025. Ouro Preto: UFOP, 2026. Disponível em: <https://propp.ufop.br/sites/default/files/conpep191.pdf>. Acesso em: 29 set. 2026. Citado na página 109.

VALVE CORPORATION. *Steam* [Plataforma de distribuição digital de jogos]. 2026. Disponível em: <https://store.steampowered.com/>. Acesso em: 13 jul. 2026. Citado na página 21.

WACHOWSKI, L.; WACHOWSKI, L. *The Matrix* [Filme]. [S.l.]: Warner Bros., 1999. Citado na página 74.

WANG, G. et al. *Voyager: An Open-Ended Embodied Agent with Large Language Models*. 2023. Disponível em: <https://arxiv.org/abs/2305.16291>. Acesso em: 06 out. 2025. Citado 2 vezes nas páginas 23 e 24.

YING, Z. et al. Chord: Chain of rendering decomposition for PBR material estimation from generated texture images. In: *Proceedings of the SIGGRAPH Asia 2025 Conference Papers*. New York, NY, USA: Association for Computing Machinery, 2025. (SA Conference Papers '25). ISBN 9798400721373. Disponível em: <https://doi.org/10.1145/3757377.3763848>. Citado 2 vezes nas páginas 15 e 24.

ZHOU, Y. et al. Large language models are human-level prompt engineers. In: *The Eleventh International Conference on Learning Representations (ICLR)*. Kigali, Rwanda: OpenReview.net, 2023. Disponível em: <https://openreview.net/forum?id=YdqwNaCLCx>. Acesso em: 19 set. 2026. Citado na página 21.

Apêndices

APÊNDICE A – Repositório da pesquisa

Devido à expressiva quantidade de *assets* gerados para cada um dos três jogos desenvolvidos, a exibição direta destes componentes no corpo deste trabalho resultaria em um volume desproporcional de páginas. Adicionalmente, a inclusão de arquivos multimídia, como trilhas sonoras, efeitos de áudio e demonstrações em vídeo, é inviável no formato de documento impresso ou digital. Por esta razão, todas as mídias geradas, os compilados de resultados e as etapas de ideação textual (incluindo *brainstorming* e *story spine*) foram consolidados em um único repositório na plataforma GitHub, no qual é possível analisar cada resultado individualmente.

Neste mesmo repositório, encontram-se disponíveis os arquivos executáveis necessários para testar os jogos desenvolvidos ao longo desta pesquisa, bem como os vídeos de demonstração das obras finalizadas.

Os materiais foram reunidos nesse repositório para que continuem acessíveis após a conclusão do curso.

O repositório (FIGUEIREDO, 2026) pode ser acessado publicamente por meio do seguinte *link*: <https://github.com/Uelfol/Repositorio-Geracoes-tcc>.

A Tabela 13 relaciona o conteúdo do repositório às seções do trabalho que remetem a ele.

Tabela 13 – Conteúdo do repositório da pesquisa e seções que o citam.

Conteúdo	**Pasta**	**Seções**
Planilha dos testes exploratórios: comandos, modelos testados, resultados e notas	02 Resultados Etapa 2 Pesquisa Exploratoria Ferramentas	3.3, 4.2.2 e 4.2.4
Planilha do mapeamento das categorias de ativos	02 Resultados Etapa 2 Pesquisa Exploratoria Ferramentas	4.1
Testes de texto, áudio, imagem, vídeo usados nos testes e os resultados completos	02 Resultados Etapa 2 Pesquisa Exploratoria Ferramentas	4.2.1, 4.2.2, 4.2.3, 4.2.4
Ideação: as árvores de ideias e as cinco propostas refinadas	03 Resultados Etapa 3 Processo de Ideacao e Selecao	4.3.1
Primeiro estudo de caso, *PinkBob J. e o Enigma do Rum*: *One Page* GDD, versões do enredo, ativos, áudios, vídeo de demonstração, jogo para *download* e instruções de uso	04 Resultados Etapa 4 Primeiro Estudo de caso PinkBob	4.4
Segundo estudo de caso, *Hacking Survivors*: expansão da história, ativos, vídeo de demonstração, jogo para *download* e instruções de uso	05 Resultados Etapa 5 Segundo Estudo de caso Hacking Survivors	4.5
Terceiro estudo de caso, *Pafúncio vs. Desenhos*: comandos e gerações, ativos, vídeo da versão de demonstração, jogo para *download* e instruções de uso	06 Resultados Etapa 6 Terceiro Estudo de caso Pafuncio	4.6

Fonte: Elaborada pelo autor (2026).

APÊNDICE B – Declaração sobre o uso de Inteligência Artificial Generativa

Conforme a Resolução CONPEP n.º 191, de 15 de junho de 2026, do Conselho Superior de Pesquisa e Pós-Graduação da Universidade Federal de Ouro Preto (UNIVERSIDADE FEDERAL DE OURO PRETO, 2026), que regula o uso de IA-Gen na pós-graduação e é adotada aqui como referência, segue a declaração sobre o uso de IA-Gen neste trabalho, que se deu de duas formas: na pesquisa, como objeto e instrumento, e na redação, como apoio à revisão do texto.

Uso na pesquisa.

- **Ferramentas:** os modelos de linguagem, de imagem, de áudio e de vídeo comparados na Etapa 2 e os escolhidos para os três estudos de caso, descritos no Capítulo 4; a versão ou a data de acesso de cada um está na respectiva entrada das Referências.
- **Finalidade e etapas:** a IA-Gen é o objeto da pesquisa e o instrumento de produção dos ativos dos três jogos (textos, imagens, áudios e código), das Etapas 2 a 6 (Seções 3.3 a 3.7).
- **Período:** de outubro de 2025 a julho de 2026.
- **Registro:** os comandos e os resultados estão no repositório da pesquisa (Apêndice A), e as figuras produzidas com IA-Gen trazem essa indicação na fonte.
- **Limitações:** as discutidas no Capítulo 5.

Uso na redação.

- **Tipo de solicitação:** Revisão de texto (correção gramatical, de pontuação e verificação de adequação às normas da ABNT).
- **Plataforma utilizada:** Google Gemini.
- **Data aproximada da utilização:** 24 de julho de 2026.
- **Comandos (*prompts*) usados:** Foi solicitado à inteligência artificial que atuasse como revisora, corrigindo eventuais erros gramaticais e falhas de pontuação no texto fornecido, além de verificar se a estrutura e as citações estavam de acordo com as normas da ABNT.

O autor é o único responsável pelo conteúdo final: todo conteúdo gerado com IA-Gen, na pesquisa e na redação, foi criticamente revisado, validado e adaptado.