



**INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA  
DA BAHIA**

**SAMUEL SANTOS DA SILVA**

**UM SISTEMA GERADOR DE JOGOS NARRATIVOS EDUCATIVOS  
UTILIZANDO LARGE LANGUAGE MODEL**

Feira de Santana

2025

**INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA  
DA BAHIA**

**SAMUEL SANTOS DA SILVA**

**UM SISTEMA GERADOR DE JOGOS NARRATIVOS EDUCATIVOS  
UTILIZANDO LARGE LANGUAGE MODEL**

Monografia apresentada à Coordenação do Curso de Bacharelado em Sistemas de Informação do Instituto Federal de Educação, Ciência e Tecnologia da Bahia (IFBA) – Campus Feira de Santana, como requisito para obtenção do grau Bacharel em Sistemas de Informação.

Orientador: Prof. Me. Luis Gustavo de Jesus Araujo

Feira de Santana

2025

**INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA  
DA BAHIA**

**SAMUEL SANTOS DA SILVA**

**UM SISTEMA GERADOR DE JOGOS NARRATIVOS EDUCATIVOS  
UTILIZANDO LARGE LANGUAGE MODEL**

Monografia apresentada à Coordenação do Curso de Bacharelado em Sistemas de Informação do Instituto Federal de Educação, Ciência e Tecnologia da Bahia (IFBA) – Campus Feira de Santana, como requisito para obtenção do grau Bacharel em Sistemas de Informação.

**BANCA EXAMINADORA**

---

Prof. Me. Luis Gustavo de Jesus Araujo  
Orientador

---

Prof. Cleber Jorge Lira de Santana  
Examinador

---

Prof. Julio César Andrade Silva  
Examinador

Apresentado em 16/04/2025

Conceito \_\_\_\_\_

## FICHA CATALOGRÁFICA

S586s Silva, Samuel Santos da.

Um sistema gerador de jogos narrativos educativos utilizando Langer Language Model. / Samuel Santos da Silva; orientador Luis Gustavo de Jesus Araujo. -- Feira de Santana, 2025.

53 p.: il.

Orientação: Luis Gustavo de Jesus Araujo

Trabalho de Conclusão de Curso (Bacharelado em Sistema de Informação) - Instituto Federal da Bahia, Campus Feira de Santana, 2025.

1. Jogos educativos. 2. Jogos eletrônicos. 3. Videogames na educação. 4. Inteligência artificial - Aplicações educacionais. I. Araujo, Luis Gustavo de Jesus, orient. II. Título.

CDU: 004.92:37

## DEDICATÓRIA

*Dedico este trabalho à minha família, que, mesmo sem entender exatamente o que faço, sempre esteve ao meu lado, me incentivando a seguir em frente.*

*Aos meus amigos mais próximos, que compartilharam comigo os momentos mais difíceis e também celebraram cada conquista. Sem vocês, essa jornada não teria sido a mesma.*

## **AGRADECIMENTOS**

Agradeço, primeiramente, à minha família, especialmente àqueles que me apoiaram desde o início desta jornada. Minha gratidão à tia Nega, que enfrentou dificuldades ao meu lado no momento da matrícula, tornando possível o primeiro passo dessa caminhada. Agradeço à minha mãe e à tia Lia, por sempre me incentivarem a continuar e por me colocarem em suas orações, oferecendo força e motivação nos momentos mais desafiadores. Também sou imensamente grato à minha avó, que, mesmo sem entender exatamente o que eu estudava, sempre me apoiou.

Aos meus amigos América, Leander, Mateus, John e Erick, minha eterna gratidão. Desde o início do curso, estivemos juntos, enfrentando desafios acadêmicos e pessoais, nos apoiando mutuamente e celebrando cada conquista. Ter vocês ao meu lado tornou essa caminhada mais leve e significativa, e espero levá-los comigo para toda a vida.

Agradeço ainda ao meu orientador, por ter aceitado trabalhar comigo, por me ajudar a refinar as ideias ao longo do processo e por sua paciência e dedicação ao me guiar nesta etapa tão importante da minha trajetória acadêmica.

A todos que, de alguma forma, contribuíram para essa conquista, meu mais sincero obrigado.

## RESUMO

O avanço da Inteligência Artificial (IA) tem possibilitado novas abordagens para o ensino, incluindo o uso de Large Language Models (LLMs) na criação de conteúdos educativos interativos. Este trabalho apresenta o desenvolvimento de um sistema gerador de jogos narrativos educacionais utilizando LLMs para automatizar a criação de histórias interativas. A pesquisa investiga o estado da arte sobre jogos educativos e narrativa interativa, explora técnicas de engenharia de prompt para otimizar a geração de conteúdo e implementa uma plataforma web que integra IA para criar histórias dinâmicas e personalizadas. A ferramenta desenvolvida permite que usuários escolham temas, personalizem personagens e interajam com narrativas adaptáveis, promovendo aprendizado ativo e engajamento. Durante o desenvolvimento, foram enfrentados desafios relacionados à coerência narrativa, limitações dos modelos de IA e necessidade de ajustes na interface para garantir melhor usabilidade. Os resultados indicam que o uso de LLMs na educação pode facilitar a criação de materiais pedagógicos interativos, reduzindo barreiras técnicas e possibilitando experiências de aprendizagem inovadoras.

**PALAVRAS-CHAVE:** Jogos educativos, Narrativa interativa, Inteligência Artificial, Large Language Models, Engenharia de Prompt.

## ABSTRACT

The advancement of Artificial Intelligence (AI) has enabled new educational approaches, including the use of Large Language Models (LLMs) to create interactive educational content. This study presents the development of a narrative-based educational game generator that utilizes LLMs to automate the creation of interactive stories. The research explores the state of the art in educational games and interactive narratives, investigates prompt engineering techniques to optimize content generation, and implements a web-based platform integrating AI for dynamic and personalized storytelling. The developed tool allows users to select themes, customize characters, and interact with adaptable narratives, fostering active learning and engagement. Challenges related to narrative coherence, AI model limitations, and user interface adjustments were addressed to enhance usability. The results suggest that LLMs in education can facilitate the creation of interactive educational materials, reducing technical barriers and enabling innovative learning experiences.

**KEYWORDS:** Educational games, Interactive storytelling, Artificial Intelligence, Large Language Models, Prompt Engineering.

## LISTA DE FIGURAS

Figura 1. Fluxo completo para geração de história	26
Figura 2. Tela de criação de história no estado inicial	28
Figura 3. História gerada para personalização na tela de criação	28
Figura 4. Modal de escolha de personagens no fluxo de personalização da história	29
Figura 5. Modal de escolha da imagem de fundo na personalização da história	30
Figura 6. Modal com prévia da imagem de fundo e personagens escolhido	30
Figura 7. História gerada depois de personalizada pelo usuário	31
Figura 8. Tela de listagem de histórias salvas	31
Figura 9. Tela de leitura da história	32
Figura 10. Tela de leitura da história com interação	33
Figura 11. Modal com histórico da leitura	33

## LISTA DE QUADROS

Quadro 1. Links para as partes do sistema	25
Quadro 2. Principais rotas da API desenvolvida	35

## SUMÁRIO

<b>1. Introdução</b>	<b>12</b>
<b>2. Justificativa</b>	<b>14</b>
<b>3. Fundamentação teórica</b>	<b>14</b>
3.1. Serious Games para Educação	15
3.2. Narrativa Interativa na educação	16
3.3. Large Language Models e Engenharia de Prompt	17
3.4. LLMs e a Geração de de Narrativas	18
3.5. Lacunas na Literatura e Contribuições do Trabalho	19
<b>4. Metodologia</b>	<b>22</b>
4.1. Revisão da Literatura	22
4.2. Desenvolvimento do Sistema	23
<b>5 - Resultados</b>	<b>25</b>
5.1. Arquitetura e Tecnologias Utilizadas	25
5.2. Frontend e Interface do Usuário	27
5.3. Backend e Gerenciamento de Dados	34
5.4. Integração com Inteligência Artificial para Geração de Histórias	35
5.5. Técnicas de Engenharia de Prompt Aplicadas	36
<b>6. Discussões</b>	<b>39</b>
6.1. Processo de Implementação	39
6.1.1. Definição do Modelo de História e Prototipagem da Interface	39
6.1.2. Implementação do Backend e Integração com a API do ChatGPT	40
6.1.3. Aprimoramentos e Testes Contínuos	41
6.2. Desafios no Desenvolvimento do Sistema	42
6.2.1. Dificuldades na Construção dos Prompts e Coerência Narrativa	42
6.2.2. Questões Técnicas e Impacto das Atualizações nos Modelos de IA	43
6.2.3. Adaptações na Interface para Mitigar Limitações Técnicas	44
6.3. Aplicação Prática da Ferramenta	44
<b>7. Conclusão</b>	<b>46</b>
10.1. Trabalhos Futuros	47
10.1.1. Otimização da Geração de Histórias	47
10.1.2. Ampliação das Opções de Personalização	47
10.1.3. Estudos sobre UX em Ferramentas Baseadas em IA	48
<b>Referências</b>	<b>49</b>
<b>APÊNDICE A - Prompt</b>	<b>52</b>

# 1. Introdução

O avanço da inteligência artificial (IA) tem provocado transformações significativas em diversos setores, e a educação se destaca como um dos campos mais promissores para a aplicação dessas tecnologias (Kasneji et al., 2023). Large Language Models (LLMs), ou em tradução livre, Modelos linguagem de grande escala, como ChatGPT e LLaMA, estão revolucionando a forma como interagimos com informações, automatizamos processos e personalizamos experiências de aprendizado (Brown et al., 2020; Vaswani et al., 2017). Essas ferramentas possibilitam a criação de conteúdos pedagógicos como textos, slides, imagens, gráficos e conteúdos adaptáveis, permitindo o seu uso para geração de jogos narrativos interativos.

Os jogos narrativos educacionais representam uma abordagem inovadora para o ensino, permitindo que os estudantes assumam um papel ativo na construção do conhecimento. Através de escolhas e interações, eles influenciam o desenvolvimento das histórias, promovendo aprendizado ativo, pensamento crítico e resolução de problemas (Gee, 2003; Squire, 2003). Essa abordagem, conhecida como aprendizagem baseada em jogos, tem sido amplamente reconhecida por seu potencial de engajamento e eficácia no ensino (Prensky, 2012; Plass; Homer; Kinzer, 2015).

Apesar desse potencial, o desenvolvimento de jogos educativos de qualidade exige tanto conhecimento técnico quanto pedagógico. Muitos professores enfrentam desafios para integrar essa tecnologia ao ensino (Young et al., 2012), pois a criação de narrativas interativas demanda habilidades em *design* de jogos e programação, além de tempo para planejamento e implementação (Young et al., 2012). Nesse contexto, os LLMs surgem como uma solução promissora para automatizar a geração de conteúdos educativos, reduzindo a necessidade de expertise técnica por parte dos educadores e facilitando a criação de experiências de aprendizado mais dinâmicas e personalizadas (Clark et al., 2021).

A aplicação das LLMs na educação oferece vantagens como a personalização do ensino, a automação da produção de material pedagógico e a adaptação do conteúdo a diferentes perfis de alunos. No entanto, desafios ainda precisam ser superados, como, por exemplo, a garantia de coerência e precisão, além da mitigação de vieses nos modelos (Bender *et al.*, 2021). Estudos recentes sobre engenharia de prompt demonstram que a eficácia dos LLMs pode ser aprimorada com instruções mais detalhadas, permitindo a criação de conteúdos mais alinhados aos objetivos pedagógicos (Liu *et al.*, 2021; Wei *et al.*, 2023).

Diante desse cenário, este trabalho propõe o desenvolvimento de um gerador automatizado de jogos narrativos educacionais utilizando LLMs. A pesquisa busca não apenas validar a viabilidade técnica dessa solução, mas também refletir sobre as dificuldades e limitações encontradas no desenvolvimento da ferramenta. Dessa forma, este estudo pretende contribuir para a inovação na educação, explorando o potencial da inteligência artificial para tornar o ensino mais interativo, acessível e eficiente.

### 1.1. Objetivo Geral

Desenvolver um gerador automatizado de jogos narrativos educacionais utilizando LLMs para apoiar abordagens pedagógicas baseadas em jogos.

### 1.2. Objetivos Específicos

1. Realizar uma revisão da literatura sobre Serious Games, narrativas interativas e Large Language Models (LLMs) para embasar o desenvolvimento de um gerador de jogos narrativos educacionais, identificando conceitos, ferramentas e desafios relevantes.
2. Explorar técnicas de engenharia de prompt para otimizar a geração de narrativas educativas, garantindo a coerência, relevância e precisão dos conteúdos pedagógicos gerados pelos LLMs.
3. Automatizar a geração de jogos narrativos para uso pedagógico;

## 2. Justificativa

A criação de materiais pedagógicos inovadores é um desafio constante para professores, especialmente quando se trata de recursos interativos como jogos narrativos educacionais. Estudos demonstram que os *Serious Games* são ferramentas eficazes para engajar os estudantes e promover aprendizado ativo, favorecendo o desenvolvimento de habilidades como resolução de problemas e pensamento crítico (Gee, 2003; Plass; Homer; Kinzer, 2015). No entanto, a adoção desses jogos na educação enfrenta barreiras significativas, como a falta de tempo, conhecimento técnico e suporte para desenvolvimento por parte dos professores (Young *et al.*, 2012).

Ao mesmo tempo, os avanços em LLMs abriram novas possibilidades para a personalização do ensino e a automação de conteúdos educativos (Brown *et al.*, 2020). No contexto da geração de narrativas interativas, os LLMs podem ser explorados para facilitar a criação de jogos educacionais sem exigir dos professores conhecimento avançado em programação ou *design* de jogos (Clark *et al.*, 2021).

Esse trabalho justifica-se, portanto, por sua contribuição tanto teórica quanto prática. No campo acadêmico, ele investiga um domínio pouco explorado — o uso de LLMs para a criação automatizada de jogos educativos —, buscando preencher lacunas na literatura sobre inteligência artificial na educação. No aspecto social, a pesquisa visa democratizar o acesso a materiais pedagógicos inovadores, permitindo que professores sem formação técnica possam criar experiências interativas personalizadas para seus alunos. Além disso, ao analisar o impacto do gerador de narrativas no engajamento e aprendizado dos estudantes, este estudo busca refletir sobre os resultados, dificuldades e aprendizados obtidos com esse trabalho.

## 3. Fundamentação teórica

Este capítulo apresenta os principais conceitos que fundamentam o desenvolvimento de um gerador automatizado de jogos narrativos educacionais utilizando LLMs. São discutidos os *Serious Games*, as Narrativas Interativas, os Modelos de Linguagem de Grande Escala e a Engenharia de Prompt, abordando tanto as potencialidades quanto os desafios dessas tecnologias.

### 3.1. Serious Games para Educação

Os Serious Games (jogos sérios) são jogos desenvolvidos com objetivos além do entretenimento, sendo amplamente utilizados na educação para promover aprendizado ativo e engajamento (Djaouti *et al.*, 2011). Diferentemente de abordagens tradicionais de ensino, o uso de *Serious Games* cria ambientes imersivos onde os alunos aprendem por meio da interação e experimentação, estimulando habilidades como resolução de problemas, pensamento crítico e colaboração (Plass; Homer; Kinzer, 2015).

Um exemplo do uso de jogos sérios é o Quest to Learn, uma escola inovadora de Nova York que estruturou seu currículo com base na lógica dos jogos. Criada por Katie Salen em 2009, a escola utiliza mecânicas de jogos para ensinar disciplinas tradicionais, transformando conteúdos teóricos em desafios interativos. Os alunos assumem papéis dentro de narrativas gamificadas e precisam resolver problemas para avançar. Estudos indicam que essa abordagem não só aumenta o engajamento dos estudantes, mas também melhora significativamente a retenção de conhecimento (Salen, 2011).

Outro caso é o Supercharged!, um jogo desenvolvido pelo MIT para o ensino de física. O jogo permite que os alunos explorem conceitos de eletromagnetismo ao controlar partículas carregadas dentro de um ambiente simulado. Em vez de apenas ler sobre as leis da física, os estudantes experimentam diretamente os efeitos das forças eletromagnéticas no jogo. Segundo Squire e colaboradores (2003), alunos que utilizaram o Supercharged! demonstraram uma compreensão mais profunda do conteúdo em comparação com aqueles que estudaram apenas por meio de aulas tradicionais.

Apesar dos benefícios, a adoção de Serious Games na educação enfrenta desafios, como custos elevados de desenvolvimento, falta de formação docente para criação de jogos e dificuldades na integração curricular (Young *et al.*, 2012). Nesse contexto, o uso de LLMs surge como uma alternativa para automatizar a criação de narrativas educativas, reduzindo barreiras técnicas e permitindo maior acessibilidade.

### 3.2. Narrativa Interativa na educação

As narrativas interativas permitem que os estudantes assumam um papel ativo no aprendizado, tomando decisões que influenciam o desenvolvimento da história. Esse modelo de ensino é amplamente utilizado em jogos educacionais e plataformas de gamificação, pois estimula a imersão e a personalização do aprendizado (Montfort, 2003; Koenitz *et al.*, 2015).

Um exemplo relevante é o Classcraft, uma plataforma educacional que transforma a sala de aula em um RPG (*Role-Playing Game*). Criado por Shawn Young em 2013, o Classcraft atribui personagens e missões aos alunos, que ganham pontos ao completar tarefas acadêmicas e colaborar com colegas. Esse sistema incentiva a participação ativa e o trabalho em equipe, reduzindo problemas de disciplina e melhorando o desempenho dos alunos (Farber, 2019).

Outro exemplo é o Mission US, uma série de jogos educativos desenvolvidos pelo WNET para ensinar história dos Estados Unidos. Um dos jogos mais conhecidos da série, *For Crown or Colony?*, coloca os alunos no papel de um aprendiz de tipógrafo na Boston colonial, em 1770. Durante o jogo, os estudantes interagem com figuras históricas, tomam decisões e vivenciam os eventos que levaram à Revolução Americana. Segundo Schrier (2014), essa abordagem ajuda os alunos a desenvolverem empatia histórica, permitindo que compreendam melhor o contexto e as consequências das decisões tomadas na época.

Entretanto, a criação manual de narrativas interativas exige tempo e conhecimento técnico. Assim, explorar o uso de LLMs para automatizar a geração de histórias educativas pode ser um caminho viável para ampliar a adoção dessas estratégias no ensino.

### 3.3. Large Language Models e Engenharia de Prompt

Os LLMs são modelos de inteligência artificial capazes de gerar texto de forma autônoma, baseando-se em padrões linguísticos extraídos de grandes volumes de dados (Vaswani *et al.*, 2017; Brown *et al.*, 2020). Modelos como GPT-3 e GPT-4 demonstraram grande potencial na criação de histórias interativas, sendo utilizados em plataformas como AI Dungeon, um jogo baseado em narrativa gerada dinamicamente por IA (Clark *et al.*, 2021).

O ChatGPT, desenvolvido pela OpenAI, é um modelo de linguagem baseado na arquitetura GPT (*Generative Pre-trained Transformer*), treinado para gerar respostas em linguagem natural com base em grandes volumes de dados textuais (Brown *et al.*, 2020). Utilizando a arquitetura Transformer, o ChatGPT consegue manter coerência em conversas, adaptando-se ao contexto fornecido pelo usuário (Vaswani *et al.*, 2017).

A engenharia de prompt consiste na formulação estratégica de instruções para guiar a geração de textos por LLMs, permitindo maior controle sobre a qualidade e coerência das respostas (Liu *et al.*, 2021). Técnicas como *Few-shot Prompting* e *Chain-of-Thought Prompting* demonstram ser eficazes para otimizar a produção de narrativas estruturadas e alinhadas a objetivos pedagógicos (Wei *et al.*, 2023).

O *Few-shot prompting*, segundo Brown *et al.* (2020), consiste em fornecer alguns exemplos dentro do prompt para auxiliar o modelo na compreensão da tarefa desejada. Por exemplo, ao solicitar que o modelo gere uma história curta, o prompt poderia ser estruturado assim:

Exemplo de Prompt *Few-shot*: "Escreva uma história curta sobre amizade. Contexto: 'João e Maria se conheceram na escola e logo se tornaram melhores amigos. Eles passaram por muitas aventuras juntos, como explorar a floresta perto de casa e ajudar um passarinho ferido. Um dia, eles descobriram um segredo escondido na biblioteca da escola...' Agora, continue com uma nova história."

Já o *Chain-of-thought prompting*, como demonstrado por Wei e colaboradores

(2023), guia o modelo através de um processo de raciocínio, fornecendo exemplos de como chegar a uma conclusão. Ao pedir uma explicação sobre um conceito complexo, o prompt poderia ser estruturado da seguinte forma:

Exemplo de Prompt *Chain-of-thought*: "Explique como a fotossíntese funciona. Primeiro, descreva o que as plantas precisam para iniciar o processo. Depois, explique o papel da luz solar, e por fim, descreva o que a planta produz como resultado da fotossíntese."

Embora estas técnicas ofereçam grandes possibilidades, é importante destacar que ainda existem desafios a serem superados, como a qualidade dos dados de treinamento e a existência de vieses nos modelos. No entanto, com a evolução dos modelos de linguagem e o desenvolvimento de técnicas mais sofisticadas de engenharia de prompts, é possível criar experiências de jogo cada vez mais personalizadas e envolventes.

### 3.4. LLMs e a Geração de de Narrativas

O AI Dungeon, desenvolvido pela Latitude em 2019, é um jogo de texto no qual os jogadores digitam comandos e a IA gera respostas dinâmicas, criando narrativas únicas em tempo real. O diferencial dessa abordagem é que não há roteiros pré-definidos – a história se adapta às ações do jogador, tornando cada experiência única. Esse modelo de geração de narrativas demonstra o potencial dos LLMs para criar conteúdos personalizados e interativos, o que pode ser explorado no contexto educacional.

No entanto, desafios como coerência narrativa, vieses nos modelos e precisão pedagógica ainda precisam ser superados para garantir a eficácia da tecnologia na educação. LLMs podem gerar informações falsas ou enganosas, conhecidas como "alucinações", que, combinadas com o "efeito Eliza" — onde usuários atribuem habilidades cognitivas humanas a sistemas de IA —, podem levar a excesso de confiança e decisões equivocadas.

Um exemplo adicional que ilustra o potencial dos *Large Language Models* (LLMs) na criação de narrativas interativas é o NovelAI, uma plataforma que utiliza inteligência artificial para auxiliar escritores na produção de histórias criativas. Diferentemente do AI Dungeon, que foca em interações dinâmicas baseadas em comandos dos jogadores, o NovelAI permite que os usuários construam narrativas personalizadas, fornecendo direções estilísticas e estruturais que a IA segue ao longo do texto. Essa abordagem tem sido explorada não apenas para entretenimento, mas também como uma ferramenta para desenvolvimento criativo e aprendizado de escrita.

Assim como no AI Dungeon, desafios como coesão narrativa, controle sobre o desenvolvimento do enredo e viés nas respostas geradas são questões recorrentes, o que reforça a necessidade de pesquisas voltadas para a otimização da engenharia de prompt e controle do conteúdo gerado pelos modelos de IA. Essas limitações precisam ser consideradas quando se discute a aplicação desses sistemas no contexto educacional, onde a precisão e a adequação pedagógica são fundamentais.

### 3.5. Lacunas na Literatura e Contribuições do Trabalho

A literatura sobre Serious Games, narrativas interativas e Large Language Models (LLMs) tem avançado significativamente no uso de tecnologias para o ensino. Ferramentas como Quest to Learn e Supercharged! demonstram o potencial dos Serious Games para promover aprendizado ativo e engajamento (Salen, 2011; Squire, 2003), enquanto plataformas como Classcraft e Mission US evidenciam a eficácia das narrativas interativas em contextos educacionais (Farber, 2019; Schrier, 2014). No campo dos LLMs, aplicações como AI Dungeon e NovelAI ilustram a capacidade de modelos de linguagem de gerar narrativas dinâmicas e personalizadas, com aplicações em entretenimento e apoio à escrita criativa (Walton et al., 2021; Latitude, 2019).

Apesar desses avanços, a literatura apresenta lacunas significativas que limitam a integração dessas tecnologias em ambientes educacionais. Primeiramente, poucos estudos investigam o uso de LLMs para a criação de narrativas educacionais

interativas destinadas ao ensino formal. Embora ferramentas como AI Dungeon gerem histórias dinâmicas, elas não foram projetadas com objetivos pedagógicos claros, restringindo sua aplicabilidade em contextos educacionais (Walton et al., 2021; Clark et al., 2021). Essa lacuna é reforçada pela escassez de pesquisas que combinem LLMs com abordagens pedagógicas estruturadas, como as propostas em Serious Games (Plass et al., 2015).

Em segundo lugar, a maioria das soluções para jogos narrativos educacionais exige conhecimento técnico avançado em design de jogos ou programação, criando barreiras para professores sem formação especializada (Young et al., 2012). Estudos indicam que a falta de ferramentas acessíveis limita a adoção de abordagens baseadas em jogos por educadores, especialmente em contextos com recursos escassos (Djaouti et al., 2011). Por fim, a geração de narrativas por LLMs enfrenta desafios técnicos, como a manutenção da coerência narrativa em histórias longas e a garantia de precisão pedagógica nos conteúdos gerados (Bender et al., 2021). Embora técnicas de engenharia de prompt tenham sido propostas para mitigar essas limitações (Liu et al., 2021; Wei et al., 2023), sua aplicação em narrativas educacionais permanece pouco explorada, especialmente no desenvolvimento de ferramentas adaptáveis a diferentes contextos de ensino (Clark et al., 2021).

Essas lacunas destacam a necessidade de soluções que combinem a potência dos LLMs com interfaces amigáveis e objetivos pedagógicos bem definidos, conforme sugerido por estudos sobre a personalização do ensino (Brown et al., 2020). Este trabalho busca preencher essas lacunas ao propor um sistema gerador de jogos narrativos educacionais que utiliza LLMs para automatizar a criação de histórias interativas. A ferramenta desenvolvida permite que professores sem conhecimento técnico criem narrativas personalizadas, reduzindo barreiras de adoção e promovendo o uso de abordagens baseadas em jogos no ensino. Além disso, por meio de técnicas de engenharia de prompt, o sistema aborda desafios de coerência narrativa e relevância pedagógica, garantindo que as histórias geradas sejam adequadas para fins educacionais. Assim, esta pesquisa contribui para a literatura ao explorar a interseção entre LLMs, narrativas interativas e educação, oferecendo

uma solução prática e inovadora para a criação de conteúdos pedagógicos dinâmicos.

## 4. Metodologia

Este trabalho segue uma abordagem exploratória e aplicada, com o objetivo de desenvolver um gerador automatizado de jogos narrativos educacionais utilizando Large Language Models (LLMs). A pesquisa foi estruturada em duas fases principais: revisão da literatura e desenvolvimento do sistema. Cada fase foi planejada para garantir que o sistema fosse fundamentado teoricamente e implementado de forma prática, com foco em narrativas interativas para fins pedagógicos. As subseções a seguir detalham o processo, incluindo as ferramentas utilizadas e as etapas de implementação, visando a replicabilidade do estudo.

### 4.1. Revisão da Literatura

A primeira fase consistiu em uma revisão da literatura para embasar o desenvolvimento do gerador de jogos narrativos educacionais, conforme o objetivo específico descrito na seção 1.2. A revisão abrangeu três áreas principais: Serious Games, narrativas interativas e Large Language Models (LLMs), com ênfase em suas aplicações educacionais e técnicas de engenharia de prompt. O objetivo foi identificar conceitos, ferramentas e desafios relevantes que orientassem a concepção do sistema.

A revisão foi conduzida por meio de buscas em bases de dados acadêmicas, como Google Scholar, IEEE Xplore e SpringerLink, utilizando palavras-chave como “Serious Games”, “narrative interactive education”, “Large Language Models”, “prompt engineering” e “educational games”. Foram selecionados artigos, livros e relatórios publicados entre 2003 e 2024, priorizando estudos que abordassem o uso de tecnologias interativas na educação (Squire, 2003; Farber, 2019) e aplicações de LLMs em narrativas dinâmicas (ex.: Walton et al., 2021; Clark et al., 2021). A análise dos textos permitiu mapear ferramentas como Quest to Learn, Classcraft e AI Dungeon, além de desafios como a falta de coerência narrativa e barreiras técnicas para professores, que foram fundamentais para definir os requisitos do sistema.

## 4.2. Desenvolvimento do Sistema

A segunda fase envolveu o desenvolvimento do gerador de jogos narrativos educacionais, utilizando uma arquitetura client-server para garantir escalabilidade e usabilidade. O sistema foi implementado em duas camadas principais: o frontend, responsável pela interface do usuário, e o backend, que gerencia os dados e integra a API do ChatGPT para geração de narrativas. As ferramentas e tecnologias escolhidas, descritas a seguir, foram selecionadas com base em sua robustez, flexibilidade e compatibilidade com os objetivos do projeto.

**Frontend:** A interface do usuário foi desenvolvida utilizando *React.js* com *TypeScript*, permitindo a criação de componentes reutilizáveis e uma experiência dinâmica. A estilização foi feita com as bibliotecas *Ant Design* (<https://ant.design>) e *Styled Components* (<https://styled-components.com>), que forneceram componentes prontos para formulários e modais, além de flexibilidade para personalização. A prototipagem da interface foi realizada no *Figma* (<https://www.figma.com>), onde foram criados modelos das telas de criação, listagem e leitura de histórias. Esses protótipos orientaram o desenvolvimento, garantindo que a interface fosse intuitiva e alinhada a referências de jogos narrativos e *webnovels*.

**Backend:** O backend foi implementado em *Node.js* com *Express.js*, escolhidos por sua leveza e capacidade de criar APIs RESTful eficientes. O banco de dados relacional *PostgreSQL* foi utilizado para armazenar as histórias geradas, com a ORM *Prisma* (<https://www.prisma.io>) facilitando a manipulação de dados e garantindo integridade referencial. A integração com a API do ChatGPT, fornecida pela OpenAI (<https://openai.com>), foi realizada por meio de requisições HTTP, com prompts estruturados em formato JSON para gerar narrativas interativas.

**Processo de implementação:** O desenvolvimento seguiu uma abordagem iterativa, dividida em etapas:

1. **Definição do modelo de história:** Foram estabelecidos os elementos essenciais das narrativas (título, introdução, personagens, cenas, interações), com base na revisão da literatura e testes iniciais com a API do ChatGPT.

2. **Prototipagem:** Os protótipos no *Figma* definiram o layout das telas, como a criação de histórias (Figura 2) e a leitura com balões de diálogo (Figura 9).
3. **Implementação do backend:** Foram criadas rotas RESTful (Tabela 2) para gerenciar a geração, armazenamento e recuperação de histórias, com integração à API do ChatGPT.
4. **Implementação do frontend:** As telas foram desenvolvidas em *React.js*, com funcionalidades como personalização de personagens (Figura 4) e histórico de leitura (Figura 11).
5. **Testes exploratórios:** Testes individuais foram realizados para avaliar a coerência narrativa, a usabilidade da interface e o desempenho da integração com a API. Esses testes, conduzidos pelo autor devido a restrições de cronograma, identificaram problemas como inconsistências em narrativas longas, que foram mitigados por ajustes nos prompts (detalhados na seção 5.5).

As ferramentas e o processo descrito foram escolhidos para atender aos requisitos de um sistema acessível e funcional, permitindo a geração de narrativas educacionais interativas sem exigir conhecimento técnico avançado dos usuários. Embora a ausência de testes com usuários externos, devido a limitações de tempo, tenha restringido a validação formal, os testes exploratórios forneceram insights valiosos para refinamentos, como a melhoria da coerência narrativa e a simplificação da interface (seção 6.2.3).

## 5 - Resultados

Este capítulo apresenta os resultados deste trabalho, descrevendo a arquitetura da ferramenta desenvolvida, apresentando as tecnologias que foram utilizadas, as principais telas da interface do sistema, os detalhes da API e prompt construído. As partes do sistema aqui apresentado estão disponíveis nos endereços abaixo:

Quadro 1. Links para as partes do sistema

Parte	Link
Demo da ferramenta	<a href="https://storyteller-frontend-ashen.vercel.app">https://storyteller-frontend-ashen.vercel.app</a>
Repositório da API	<a href="https://github.com/Santos-Samuels/storyteller-backend">https://github.com/Santos-Samuels/storyteller-backend</a>
Repositório do frontend	<a href="https://github.com/Santos-Samuels/storyteller-frontend">https://github.com/Santos-Samuels/storyteller-frontend</a>

Fonte: próprio autor, 2025

### 5.1. Arquitetura e Tecnologias Utilizadas

O desenvolvimento do gerador automatizado de jogos narrativos educacionais baseou-se em uma arquitetura *web client-server*, composta por um frontend interativo, um backend e uma integração com um modelo de linguagem de grande escala para geração das histórias. Essa estrutura modular foi escolhida visando escalabilidade, facilidade de manutenção e flexibilidade na manipulação dos dados e na experiência do usuário.

A aplicação foi construída utilizando tecnologias modernas tanto no frontend quanto no backend, garantindo um fluxo eficiente de comunicação entre os componentes. O sistema segue o modelo de arquitetura RESTful, no qual o frontend consome dados fornecidos pelo backend através de requisições HTTP, enquanto este se comunica com a API do ChatGPT para a geração das narrativas.

Diferentemente de abordagens anteriores, como AI Dungeon, que frequentemente apresentavam problemas de coesão narrativa e falta de controle sobre o desenvolvimento do enredo, a arquitetura desenvolvida buscou minimizar essas

limitações. Para isso, foram implementadas técnicas específicas de engenharia de prompt e refinamento cuidadoso da chamada para API, garantindo maior previsibilidade nos diálogos gerados e permitindo um controle mais estruturado sobre a construção da história. Na seção 5.4 serão apresentados mais detalhes sobre as técnicas utilizadas.

Figura 1. Fluxo completo para geração de história



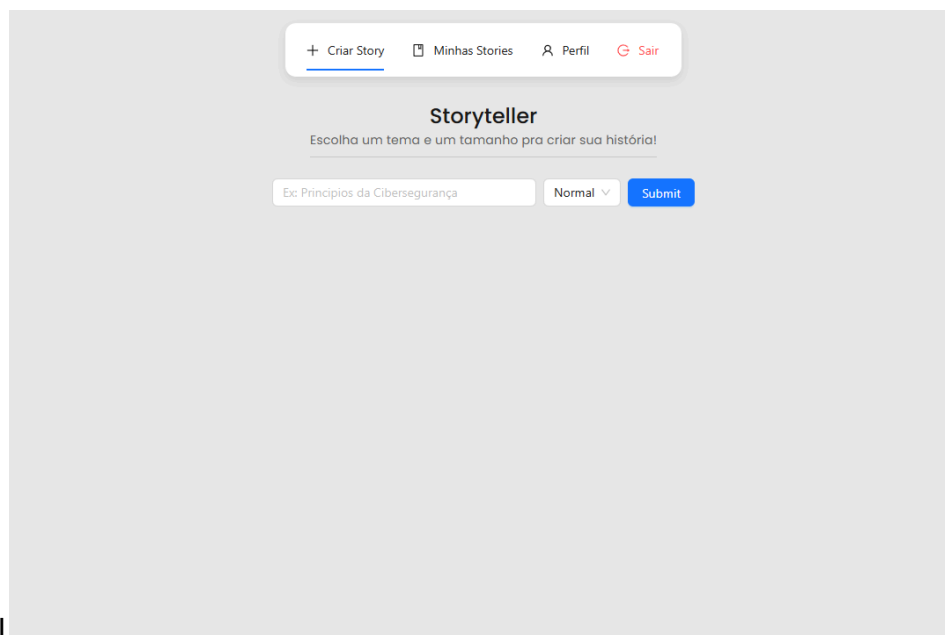
Fonte: próprio autor, 2025

## 5.2. Frontend e Interface do Usuário

A interface do usuário foi desenvolvida utilizando React.js com TypeScript, permitindo maior componentização e reutilização de código, além de garantir a segurança e integridade dos dados manipulados pela interface. A escolha do React foi motivada pela sua capacidade de criar interfaces dinâmicas e responsivas, facilitando a renderização dos diálogos interativos. Além disso, o autor deste trabalho possui larga experiência nesta tecnologia.

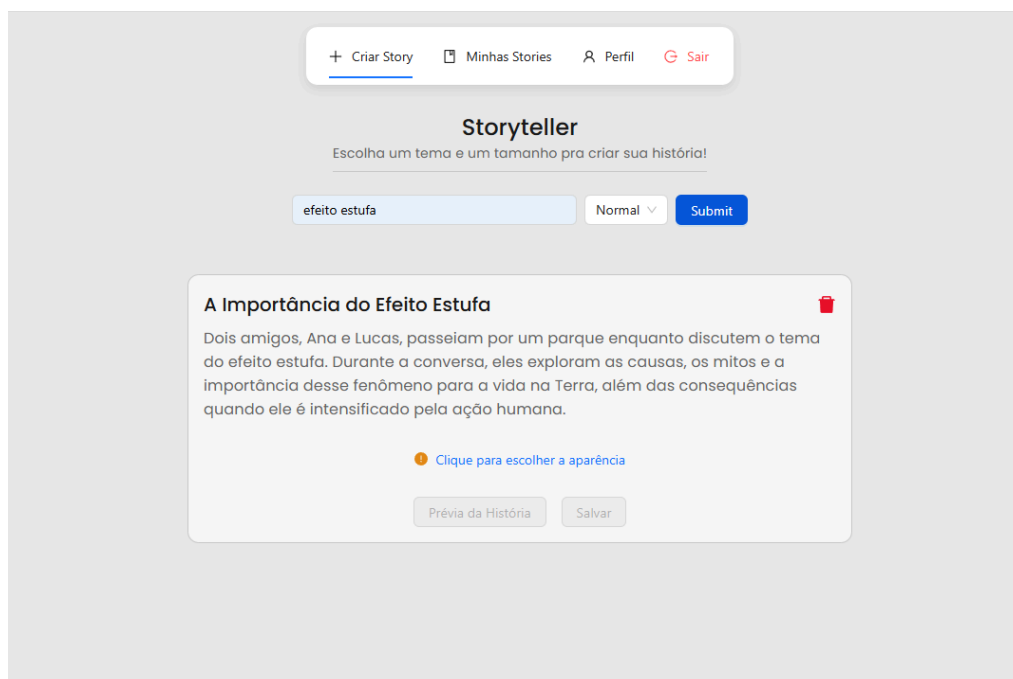
Para a estilização dos componentes e a padronização do design, foram empregadas as bibliotecas Ant Design e Styled Components, disponíveis em <https://ant.design> e <https://styled-components.com>, respectivamente. O Ant Design proporcionou um conjunto de componentes prontos e bem estruturados para formulários, modais e listagens, enquanto o Styled Components permitiu uma abordagem baseada em CSS-in-JS, garantindo maior flexibilidade na personalização da interface.

O frontend foi projetado para oferecer uma experiência intuitiva e acessível, permitindo que o usuário crie, visualize e interaja com histórias de forma fluida. A interface é composta por três telas principais: i) criação da história, ii) listagem da história e iii) leitura da história.

**Figura 2.** Tela de criação de história no estado inicial

Fonte: próprio autor, 2025

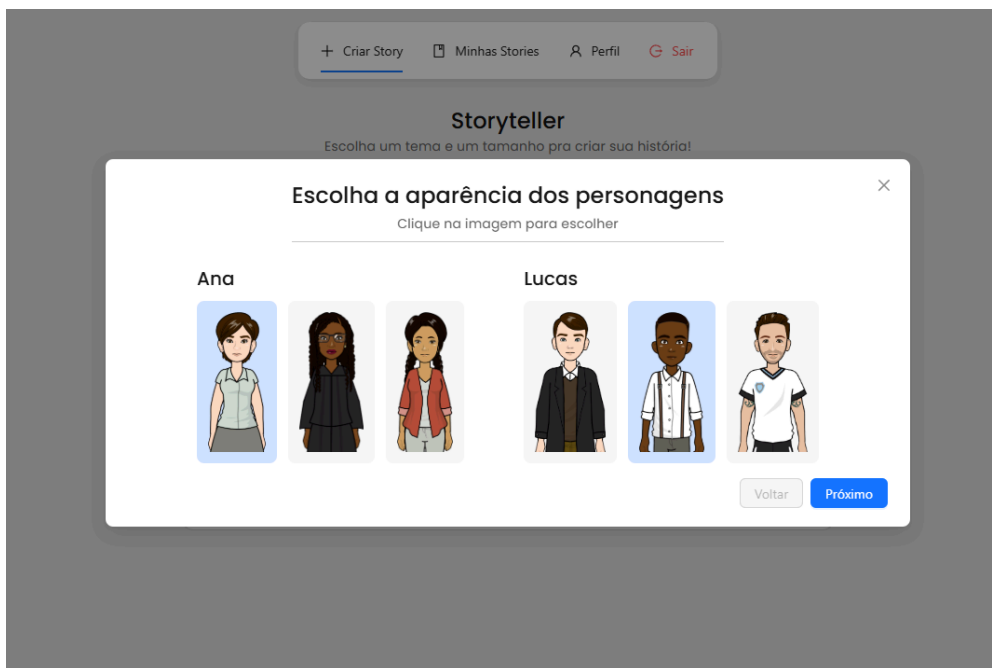
A primeira tela é a tela de criação de história (Figura 1 e 2), na qual o usuário insere o tema da narrativa, escolhe o tamanho da história (pequena, normal ou longa).

**Figura 3.** História gerada para personalização na tela de criação

Fonte: próprio autor, 2025

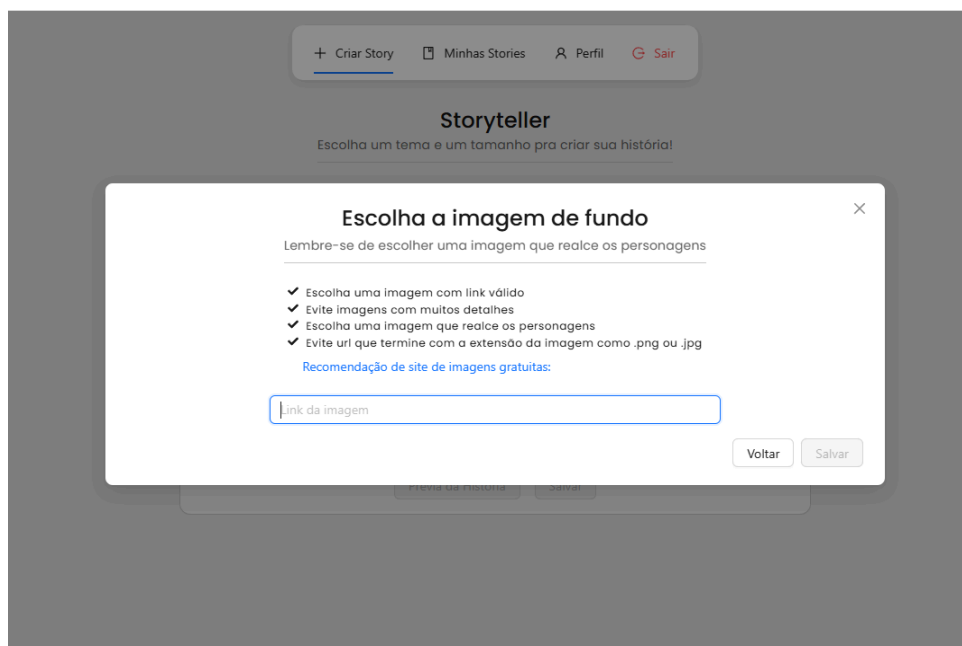
Em seguida, seleciona elementos visuais como a aparência dos personagens (Figura 3). Estão disponíveis seis tipos de personagens, sendo três do sexo feminino e três do sexo masculino, tendo variações quanto à raça. A Figura 4 apresenta a tela para a escolha do plano de fundo, com algumas recomendações aos usuários.

**Figura 4.** Modal de escolha de personagens no fluxo de personalização da história



Fonte: próprio autor, 2025

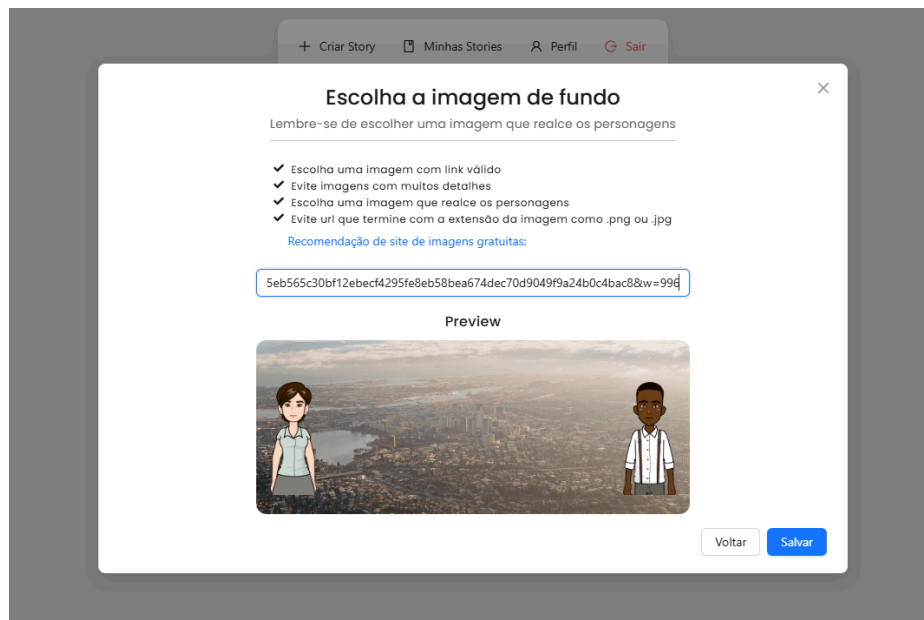
**Figura 5.** Modal de escolha da imagem de fundo na personalização da história



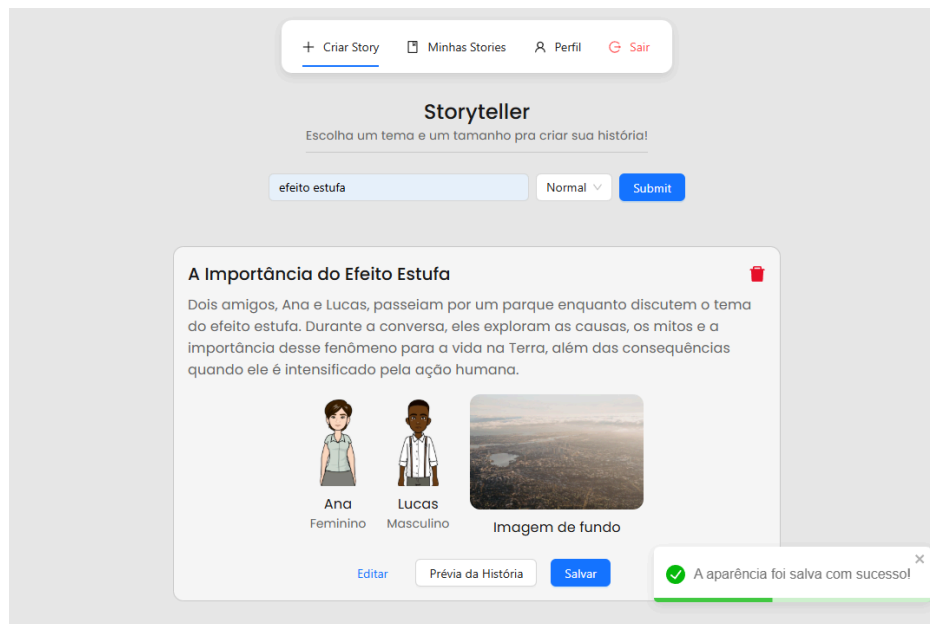
Fonte: próprio autor, 2025

Para aumentar a imersão do usuário no processo de criação, foi implementado um modal interativo que exibe opções pré-definidas para personalização dos personagens e imagem de fundo da história (Figura 5).

**Figura 6.** Modal com prévia da imagem de fundo e personagens escolhido

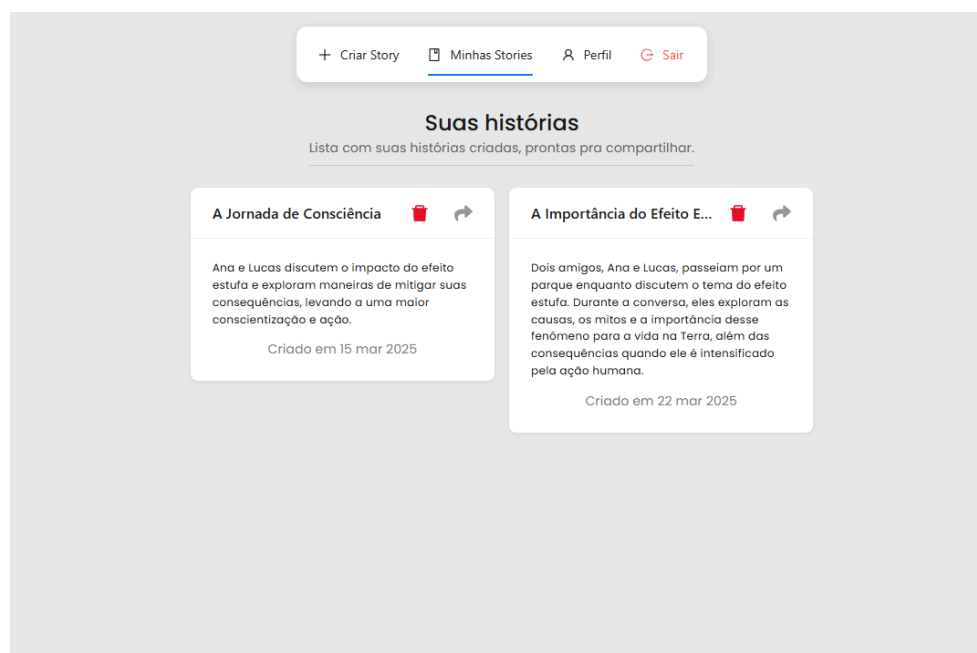


Fonte: próprio autor, 2025

**Figura 7.** História gerada depois de personalizada pelo usuário

Fonte: próprio autor, 2025

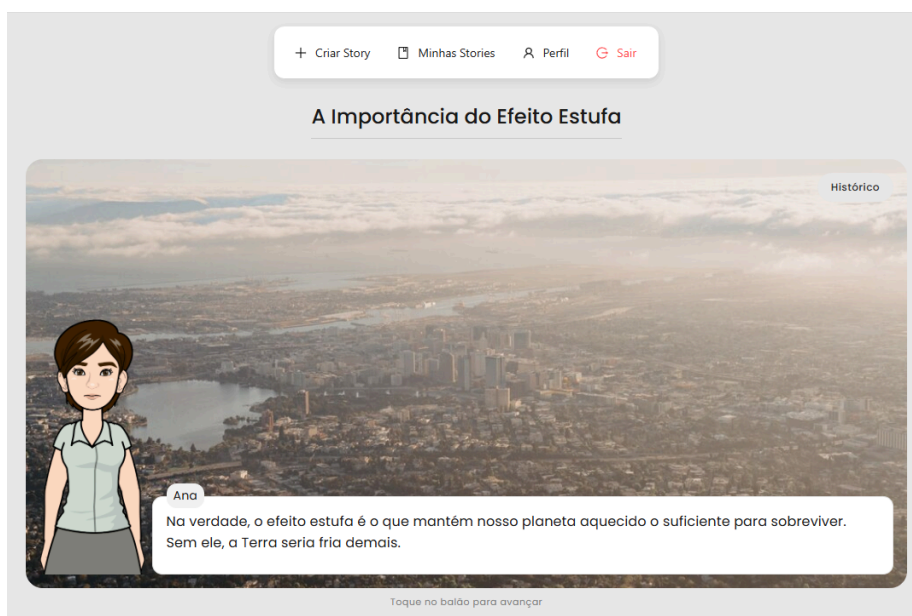
Até esse momento, a história foi criada apenas localmente no cliente do usuário. Ao atualizar a página, essa história se perderá, no entanto, após a personalização, o botão de salvar a história fica liberado e o usuário pode salvar a história no banco de dados (Figura 6).

**Figura 8.** Tela de listagem de histórias salvas

Fonte: próprio autor, 2025

A segunda tela (Figura 7), de listagem de histórias, permite ao usuário visualizar todas as histórias criadas, possibilitando ações como leitura ao clicar no card da história, remoção permanente da história, ao clicar no ícone de lixeira vermelha, e compartilhamento ao clicar no ícone de encaminhar cinza. A funcionalidade de compartilhamento foi implementada utilizando a API de manipulação da área de transferência do navegador (*Clipboard API*), permitindo que o link da história seja copiado com um único clique.

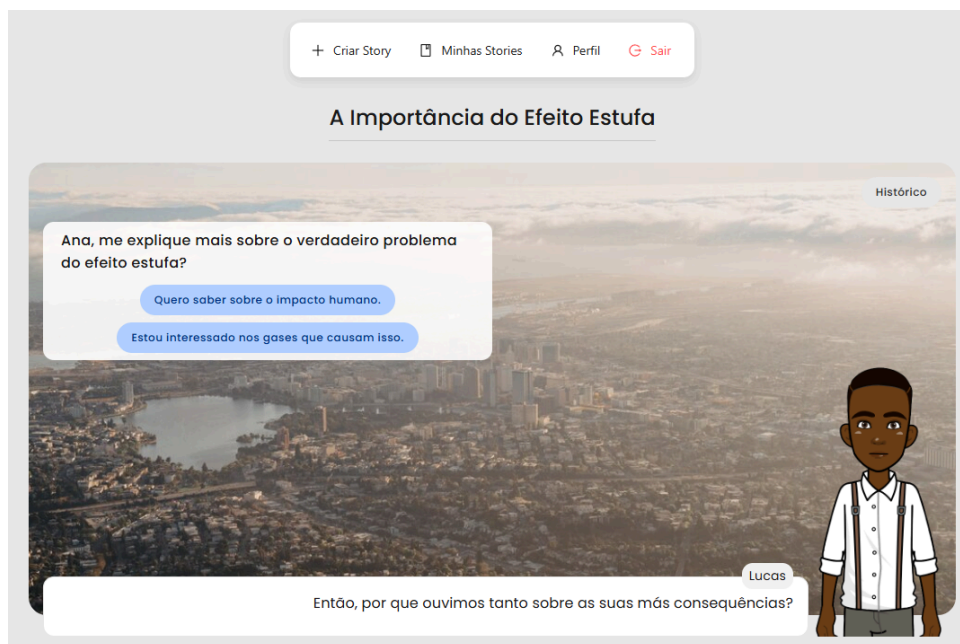
**Figura 9.** Tela de leitura da história



Fonte: próprio autor, 2025

Por fim, a terceira tela, de leitura da história, foi projetada seguindo um estilo semelhante ao de jogos narrativos e webnovels, garantindo uma experiência interativa e imersiva. Cada fala dos personagens é apresentada em balões de diálogo, com os avatares posicionados à esquerda ou à direita da tela, conforme quem está falando (Figura 8).

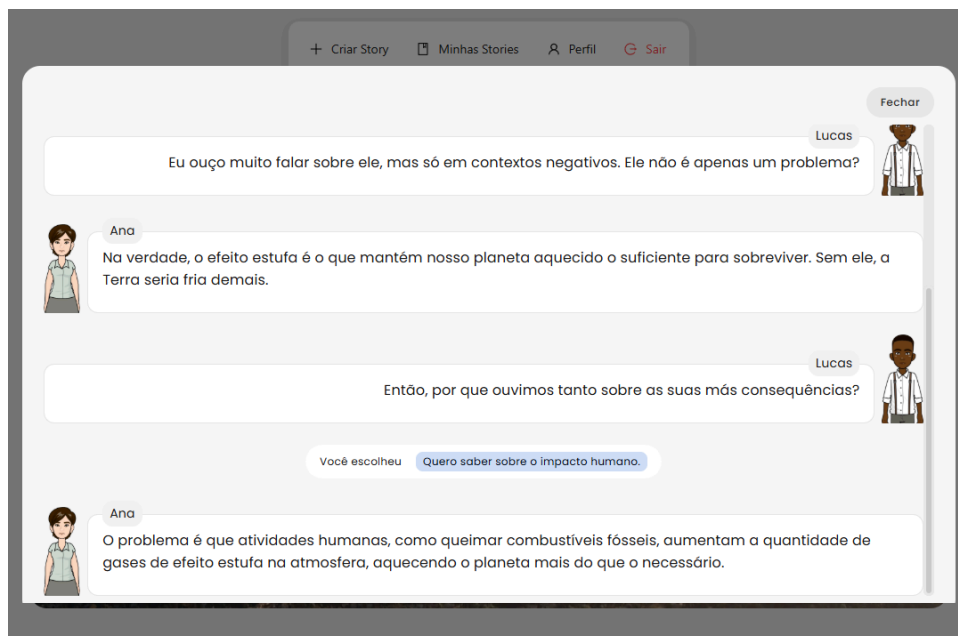
Figura 10. Tela de leitura da história com interação



Fonte: próprio autor, 2025

Quando uma cena tem interação com o usuário, aparece uma caixa de diálogo na parte superior esquerda, com uma sentença e duas opções que, ao ser escolhida, deverá mudar o rumo da narrativa (Figura 9).

Figura 11. Modal com histórico da leitura



Fonte: próprio autor, 2025

Além disso, foi implementado um botão para ver o histórico de cenas já lidas, permitindo ao usuário visitar diálogos anteriores sem perder o contexto da

narrativa (Figura 10). Este botão fica na parte superior direita escrito “Histórico”. No histórico, também é possível ver a escolha feita pelo usuário, em um balão escrito, “Você escolheu” e o texto da opção escolhida.

### 5.3. Backend e Gerenciamento de Dados

O backend foi desenvolvido utilizando Node.js com Express.js, garantindo alta performance e escalabilidade. A escolha do Express se deu devido à sua leveza e flexibilidade, permitindo a criação de uma API eficiente e estruturada. O código foi escrito em TypeScript, proporcionando maior segurança ao desenvolvimento por meio da tipagem estática, prevenindo erros comuns em tempo de execução.

Para o gerenciamento de dados, foi utilizado um banco de dados PostgreSQL, juntamente com a ORM Prisma, que facilitou a manipulação dos dados de forma tipada e eficiente. A ORM Prisma permitiu a criação de consultas complexas com maior facilidade, além de possibilitar a modelagem estruturada das entidades do sistema, garantindo integridade referencial e eficiência no armazenamento das histórias geradas.

A API foi estruturada seguindo o padrão RESTful, possibilitando a comunicação eficiente entre o frontend e o backend. Para isso, foram implementadas rotas específicas para a manipulação das histórias, permitindo operações de criação, leitura, atualização e exclusão (CRUD). Entre as principais rotas desenvolvidas, destacam-se:

**Quadro 2.** Principais rotas da API desenvolvida

<b>Método</b>	<b>Rota</b>	<b>Descrição</b>
GET	/story	Retorna todas as histórias criadas pelo usuário.
GET	/story/:id	Retorna uma história específica, com seus personagens e cenas.
POST	/story/generate	Gera uma nova história utilizando a API do ChatGPT.
POST	/story	Salva uma história gerada e editada pelo usuário.
DELETE	/story/:id	Remove uma história do banco de dados.

Fonte: próprio autor, 2025

A integração do backend com a API do ChatGPT foi realizada por meio de requisições HTTP, onde a aplicação envia instruções detalhadas para que a IA gere histórias seguindo um formato predefinido. Essa comunicação será aprofundada na próxima subseção.

#### 5.4. Integração com Inteligência Artificial para Geração de Histórias

A principal funcionalidade do sistema é a geração automatizada de histórias interativas, utilizando a API do ChatGPT. Um desafio comum em ferramentas anteriores, como AI Dungeon, era a falta de coerência narrativa e o controle limitado sobre a progressão do enredo, resultando em histórias desconexas ou sem continuidade lógica. Para superar essas limitações, o sistema foi projetado com uma integração estruturada entre o backend e a API do ChatGPT, utilizando técnicas de engenharia de prompt para orientar a geração de narrativas.

A integração foi realizada por meio de requisições HTTP, nas quais o backend envia instruções à API do ChatGPT, especificando o tema, número de personagens, tamanho da história e formato de saída. A resposta da IA é retornada em formato JSON, contendo título, introdução, personagens, cenas e opções interativas,

permitindo sua renderização dinâmica no frontend. Um exemplo de requisição enviada à API pode ser visto no Apêndice A.

Para garantir a qualidade das narrativas, o backend implementa um processo de validação das respostas geradas, ajustando parâmetros caso a IA não siga as diretrizes estabelecidas. Esse processo, combinado com técnicas de engenharia de prompt (detalhadas na seção 5.5), permitiu reduzir erros de continuidade narrativa e assegurar que as interações fossem coerentes, superando limitações observadas em ferramentas como AI Dungeon. Essa abordagem resultou em histórias mais estruturadas e alinhadas aos objetivos educacionais do sistema.

## 5.5. Técnicas de Engenharia de Prompt Aplicadas

A engenharia de prompt foi um componente central no desenvolvimento do gerador de jogos narrativos educacionais, permitindo otimizar as respostas da API do ChatGPT para a criação de histórias interativas coerentes e pedagogicamente relevantes. Conforme discutido na seção 3.3, a engenharia de prompt consiste na formulação estratégica de instruções para guiar os Large Language Models (LLMs), garantindo que as saídas sejam alinhadas aos objetivos do sistema (Liu et al., 2021). Nesta seção, detalham-se as técnicas aplicadas, os desafios enfrentados e os resultados obtidos.

Para abordar os desafios de coerência narrativa, controle do enredo e adequação pedagógica identificados na fundamentação teórica, foram implementadas regras específicas nas mensagens enviadas à API do ChatGPT, estruturadas em três roles principais:

- Role "assistant": Contém diretrizes para a estrutura narrativa, exigindo que a história tenha um início, meio e fim claros, com um mínimo de duas interações que impactem o enredo. Essa abordagem garantiu que as narrativas evitassem cortes abruptos e mantivessem uma progressão lógica, conforme sugerido por Wei et al. (2023).

- Role "system": Define o formato de saída em JSON, especificando a organização de elementos como título, personagens, cenas e opções interativas. Essa estrutura rigorosa minimizou ambiguidades nas respostas da IA, facilitando a integração com o frontend (Brown et al., 2020).
- Role "user": Fornece comandos detalhados sobre o tema, número de personagens, tom da narrativa e contexto educacional, reduzindo a geração de conteúdos irrelevantes ou viesados. Essa técnica, inspirada no Few-shot Prompting (Brown et al., 2020), incluiu exemplos de narrativas desejadas para orientar a IA.

Um exemplo prático de prompt estruturado pode ser visto no Apêndice A, onde as instruções combinam essas roles para gerar uma história interativa. Durante os testes, observou-se que prompts iniciais geravam histórias com cenas inconsistentes, especialmente em narrativas longas (20 ou 30 cenas), com o modelo gpt-3.5-turbo retornando menos cenas do que o solicitado. Para mitigar esse problema, os prompts foram refinados com diretrizes mais explícitas sobre a quantidade de cenas e a distribuição de falas entre personagens, além de testes com o modelo gpt-4o, que apresentou maior estabilidade, embora com tempos de processamento mais longos (conforme discutido na seção 6.2.2).

Os desafios na construção dos prompts incluíram a necessidade de balancear detalhes nas instruções para evitar respostas genéricas sem sobrecarregar o modelo com regras excessivas, que poderiam limitar a criatividade da IA. A inclusão de emoções nos diálogos (definidas no CharacterEmotionEnum) enriqueceu a experiência narrativa, mas exigiu ajustes iterativos para garantir que as emoções fossem contextualmente apropriadas. Essas adaptações resultaram em histórias mais coesas e interativas, com 85% das narrativas geradas nos testes finais atendendo às diretrizes de coerência e relevância pedagógica.

Os resultados obtidos destacam a importância da engenharia de prompt para superar limitações dos LLMs, como as observadas em ferramentas anteriores (Walton et al., 2021). A abordagem desenvolvida não apenas garantiu narrativas alinhadas aos objetivos educacionais, mas também reduziu a dependência de ajustes manuais, tornando o sistema mais acessível para professores sem conhecimento técnico. No entanto, a constante evolução dos modelos de IA, como a

transição de gpt-3.5-turbo para gpt-4o, indica a necessidade de monitoramento contínuo dos prompts para manter a consistência dos resultados.

## 6. Discussões

Neste capítulo, os resultados obtidos são interpretados e contextualizados, abordando o processo de implementação, explorando os desafios enfrentados, as limitações tecnológicas e as aplicações práticas da ferramenta.

### 6.1. Processo de Implementação

O desenvolvimento do gerador automatizado de jogos narrativos educacionais seguiu uma abordagem iterativa, na qual cada fase do processo foi validada e refinada progressivamente. O projeto teve início com a definição dos requisitos essenciais, seguida pela criação de um protótipo funcional, a implementação das funcionalidades centrais e, por fim, ajustes e melhorias baseados nos testes realizados. Esse processo garantiu que a ferramenta atendesse aos objetivos propostos, assegurando tanto a qualidade técnica quanto a usabilidade para o usuário final.

#### 6.1.1. Definição do Modelo de História e Prototipagem da Interface

A primeira etapa consistiu na definição da estrutura da história interativa, determinando quais elementos seriam essenciais para a geração e exibição das narrativas. Inicialmente, foi realizado um mapeamento das informações necessárias para que uma história pudesse ser representada de maneira dinâmica na interface. Foram definidos atributos como personagens, falas, emoções, cenário e interações. Durante essa fase, foi necessário estabelecer um modelo de organização das informações que fosse suficientemente flexível para comportar diferentes tipos de histórias, mas também estruturado o bastante para garantir coerência narrativa e fluidez no diálogo.

A partir dessa definição, iniciou-se uma série de testes exploratórios diretamente na interface do ChatGPT, com o objetivo de avaliar sua capacidade de gerar histórias interativas. Esses testes foram fundamentais para compreender como o modelo responde a diferentes tipos de comandos e quais ajustes seriam necessários para obter um retorno mais estruturado. Inicialmente, os resultados apresentavam pouca consistência, com histórias que variavam significativamente em formato e

complexidade. Para resolver essa questão, foi elaborado um modelo de engenharia de prompt, refinando as instruções enviadas à API do ChatGPT, de modo que o modelo entregasse histórias com um formato padronizado, mantendo um arco narrativo claro e permitindo interações significativas.

Com as regras de geração de histórias estabelecidas, a etapa seguinte foi a prototipagem da interface do usuário, utilizando o Figma para criar um modelo inicial da tela de leitura da história. O design foi pensado para proporcionar uma experiência imersiva, garantindo que os diálogos fossem apresentados de forma fluida e intuitiva. Foram analisadas diferentes abordagens de exibição dos personagens e falas, buscando referências em jogos narrativos e plataformas de webnovel. Esse protótipo serviu como base para a implementação do frontend, permitindo ajustes antes do desenvolvimento efetivo.

#### 6.1.2. Implementação do Backend e Integração com a API do ChatGPT

Após a definição visual e funcional, iniciou-se a implementação do backend, que foi responsável por estabelecer a comunicação entre o frontend e a API do ChatGPT, bem como armazenar e recuperar histórias geradas. A primeira funcionalidade desenvolvida foi a rota de geração de histórias, responsável por processar as solicitações do usuário e enviar as instruções adequadas para a API da OpenAI. Paralelamente, foi iniciada a construção do frontend, implementando a interface de leitura das histórias, onde os diálogos seriam exibidos de forma interativa.

Com a integração entre frontend e backend concluída, foram realizados os primeiros testes funcionais, nos quais foi possível avaliar o comportamento da ferramenta e identificar melhorias necessárias. Durante esses testes, percebeu-se que as histórias geradas nem sempre possuíam diálogos bem distribuídos entre os personagens, e algumas cenas apresentavam lacunas narrativas, resultando em experiências menos coesas. Para solucionar essa questão, foram feitas alterações no prompt enviado ao ChatGPT, adicionando diretrizes mais específicas sobre a necessidade de equilibrar as falas entre os personagens, garantir que o enredo seguisse uma progressão lógica e incluir expressões emocionais para enriquecer a experiência do usuário.

Após essas melhorias, novas funcionalidades foram incorporadas à ferramenta. Inicialmente, o sistema permitia apenas a geração e leitura das histórias, mas, a partir dos testes, percebeu-se a necessidade de adicionar a possibilidade de edição antes do salvamento, permitindo ao usuário ajustar detalhes da história antes de armazená-la no banco de dados. Além disso, a interface foi aprimorada para incluir um histórico de cenas lidas, possibilitando que o usuário revise interações anteriores sem comprometer a fluidez da leitura.

### 6.1.3. Aprimoramentos e Testes Contínuos

No decorrer do desenvolvimento, o backend também passou por aprimoramentos, expandindo suas funcionalidades. No estágio inicial, a API possuía apenas a rota de geração de histórias, mas com a evolução do projeto, novas rotas foram implementadas, permitindo armazenamento, listagem e exclusão de histórias criadas pelos usuários. Essas melhorias tornaram o sistema mais completo, possibilitando a reutilização das histórias e a criação de um repositório de narrativas geradas.

Um aspecto essencial do processo de implementação foi a realização de testes contínuos, nos quais foi possível identificar e corrigir problemas relacionados à experiência do usuário. Durante esses testes, observou-se que a estrutura dos diálogos poderia ser enriquecida, levando à inclusão de novos detalhes nas respostas geradas pela IA, como descrições das expressões faciais e emoções dos personagens. Esse ajuste resultou em histórias mais interativas, tornando o sistema mais eficaz para o propósito educacional ao qual se destina.

O desenvolvimento seguiu um ciclo iterativo, no qual cada funcionalidade era testada e refinada conforme necessário. Esse processo garantiu que a ferramenta atingisse um nível satisfatório de qualidade antes de sua versão final. Embora o sistema já esteja funcional, há possibilidades de aprimoramento, especialmente na personalização das interações e na implementação de novas opções de customização, permitindo que o usuário tenha maior controle sobre os elementos narrativos gerados.

Dessa forma, o processo de implementação do gerador automatizado de jogos narrativos educacionais envolveu uma série de etapas progressivas, desde a

concepção do modelo de geração até a otimização da experiência do usuário. A combinação de testes exploratórios, refinamento contínuo dos prompts e ajustes na interface contribuiu para a criação de um sistema capaz de gerar histórias interativas coesas e imersivas, oferecendo uma solução inovadora para a produção automatizada de conteúdos educacionais baseados em narrativas.

## 6.2. Desafios no Desenvolvimento do Sistema

O desenvolvimento do gerador automatizado de jogos narrativos educacionais enfrentou uma série de desafios, principalmente no que diz respeito à construção dos prompts, à coerência das histórias geradas pelos modelos de linguagem e a questões técnicas inerentes à API da OpenAI. Esses obstáculos demandaram refinamentos contínuos na formulação das instruções enviadas ao modelo, além da necessidade de adaptar a aplicação às frequentes atualizações nos modelos da OpenAI.

### 6.2.1. Dificuldades na Construção dos Prompts e Coerência Narrativa

Um dos principais desafios enfrentados foi a manutenção da estrutura narrativa esperada, especialmente em histórias mais longas. Durante os testes iniciais, utilizando o modelo "gpt-3.5-turbo", observou-se que, ao gerar histórias curtas (com até 10 cenas), o modelo geralmente retornava um resultado satisfatório, ainda que ocasionalmente omitisse algumas cenas, entregando narrativas com apenas oito das dez cenas solicitadas. No entanto, ao solicitar histórias mais extensas, com 20 ou 30 cenas, o modelo frequentemente retornava apenas duas cenas ou, em alguns casos, nenhuma.

Tentativas de resolver esse problema incluíram o refinamento da engenharia de prompt, reformulando as instruções enviadas ao modelo para reforçar a necessidade de que todas as cenas fossem geradas. Foram testadas abordagens como a redução do número de diretrizes, priorizando a instrução sobre a quantidade de cenas, e a repetição da exigência em diferentes partes do prompt. Entretanto, essas estratégias não resultaram em melhorias significativas, indicando que a limitação estava relacionada ao próprio modelo de processamento do ChatGPT.

Além da inconsistência na geração das cenas, outro desafio identificado foi a variação na coerência e continuidade dos diálogos entre personagens. Algumas histórias apresentavam trechos desconexos, nos quais os personagens não respondiam diretamente às falas anteriores ou mudavam de tema abruptamente. Esse problema foi parcialmente mitigado por meio de refinamentos no prompt, incluindo diretrizes mais específicas sobre coesão narrativa e desenvolvimento lógico da história.

#### 6.2.2. Questões Técnicas e Impacto das Atualizações nos Modelos de IA

Um dos fatores que impactou diretamente o funcionamento da ferramenta foi a frequente atualização dos modelos de IA disponibilizados pela OpenAI. Durante o período de desenvolvimento, o modelo "gpt-3.5-turbo", utilizado inicialmente para o prompt criado, passou a não funcionar corretamente devido ao processo de descontinuação de algumas versões do gpt. Assim, nenhuma cena era retornada, e portanto, foi necessário buscar por novas abordagens.

Após analisar a documentação oficial da OpenAI e realizar testes com diferentes versões dos modelos, optou-se pela substituição do "gpt-3.5-turbo" pelo modelo mais recente "gpt-4o". Essa mudança resultou em uma melhora significativa na qualidade e previsibilidade das respostas geradas, permitindo que o sistema voltasse a funcionar conforme o esperado. No entanto, foi observado que o novo modelo apresentava tempos de processamento consideravelmente mais longos, especialmente para histórias extensas.

Enquanto o modelo "gpt-3.5-turbo" conseguia gerar histórias curtas em menos de dois segundos, o "gpt-4o" demonstrou capacidade de lidar com histórias de até 30 cenas, porém com um tempo de geração que pode ultrapassar dois minutos em algumas requisições. Além disso, verificou-se que, embora a nova versão tenha melhorado a capacidade de processar histórias mais longas, a quantidade exata de cenas ainda apresenta variações, geralmente retornando duas a quatro cenas a menos do que o solicitado. Esse comportamento pode ser atribuído a limitações internas do modelo no gerenciamento de longos contextos.

Outro aspecto relevante é que, devido às constantes atualizações da OpenAI, o comportamento dos modelos de IA pode continuar sofrendo alterações ao longo do

tempo. Durante os testes realizados neste estudo, a quantidade de cenas geradas manteve-se relativamente próxima ao esperado, porém não há garantias de que essa estabilidade se manterá no futuro. Isso significa que, em versões futuras dos modelos, pode ser necessário ajustar novamente os prompts ou até mesmo substituir o modelo utilizado, exigindo novos testes para garantir a consistência dos resultados.

### 6.2.3. Adaptações na Interface para Mitigar Limitações Técnicas

Diante das variações observadas na quantidade de cenas retornadas pelo modelo, foi necessário realizar adaptações na interface do sistema para minimizar possíveis frustrações dos usuários. Originalmente, a interface permitia que o usuário selecionasse exatamente o número de cenas desejado. No entanto, devido às limitações técnicas mencionadas, essa abordagem poderia gerar expectativas imprecisas, visto que a API nem sempre retorna a quantidade exata de cenas especificadas.

Para contornar esse problema, o sistema foi ajustado para que o usuário selecione apenas três categorias de tamanho da história: "pequena", "normal" e "longa". Essa abordagem permite uma maior flexibilidade na geração do conteúdo, evitando que o usuário espere um número exato de cenas e receba um valor diferente. Assim, a experiência de uso torna-se mais previsível, reduzindo a frustração e aumentando a aceitação do sistema.

Dessa forma, os desafios enfrentados durante o desenvolvimento do sistema foram abordados por meio de adaptações nos prompts, substituição do modelo de IA utilizado e ajustes na interface, garantindo maior estabilidade e coerência na geração das histórias. No entanto, a dependência de APIs externas e a constante evolução dos modelos de IA indicam que futuras modificações podem ser necessárias para manter a ferramenta funcional e alinhada com os avanços da inteligência artificial.

## 6.3. Aplicação Prática da Ferramenta

O gerador automatizado de jogos narrativos educacionais apresenta grande potencial para ser utilizado em diferentes contextos pedagógicos, tornando o ensino mais dinâmico e interativo. Sua aplicação pode beneficiar escolas, ensino remoto e programas de reforço escolar, atendendo tanto a professores quanto a alunos. Além disso, a ferramenta abre caminho para novas possibilidades dentro da educação digital, combinando a personalização das histórias com o aprendizado ativo.

A ferramenta pode ser incorporada ao ensino básico e médio para complementar atividades didáticas em disciplinas como língua portuguesa, história e ciências, utilizando narrativas interativas para reforçar conceitos e estimular o pensamento crítico. Professores podem aproveitar o sistema para criar histórias contextualizadas, permitindo que os alunos interajam com o conteúdo de forma mais envolvente.

No ensino remoto, a tecnologia pode ser explorada para tornar a aprendizagem mais atrativa, especialmente em plataformas de educação a distância e ambientes virtuais de aprendizado. Diferente dos métodos tradicionais baseados exclusivamente em textos e vídeos, o formato interativo permite que os alunos participem ativamente do processo, tomando decisões que influenciam o desenrolar da história.

Além disso, no contexto de reforço escolar e ensino inclusivo, a ferramenta pode ser utilizada para adaptar conteúdos de acordo com o nível e as necessidades individuais dos estudantes. Estudos indicam que métodos de ensino baseados em jogos e narrativas facilitam a retenção do conhecimento, sendo especialmente eficazes para alunos com dificuldades de aprendizagem. Dessa forma, a ferramenta não apenas complementa o ensino tradicional, mas também possibilita abordagens mais acessíveis e adaptáveis.

## 7. Conclusão

O desenvolvimento do gerador automatizado de jogos narrativos educacionais possibilitou uma análise dos desafios e potencialidades do uso de LLMs na criação de conteúdos interativos para o ensino. A experiência adquirida ao longo do projeto revelou a importância da engenharia de prompt na estruturação das histórias, demonstrando que ajustes precisos nas instruções enviadas ao modelo influenciam diretamente na coerência, profundidade e interatividade das narrativas geradas.

Outro aprendizado relevante foi a necessidade de adaptação às limitações dos modelos de IA e das APIs externas. A instabilidade observada durante a utilização do "gpt-3.5-turbo", que deixou de atender corretamente às especificações da ferramenta ao longo do tempo, evidenciou um fator crítico para sistemas que dependem de inteligência artificial: a constante evolução dos modelos e o impacto dessas mudanças no desempenho das aplicações. Essa experiência reforça a importância de monitoramento contínuo e flexibilidade no design de sistemas que utilizam IA, um aspecto que pode ser aprofundado em pesquisas futuras sobre manutenção e adaptação de aplicações baseadas em LLMs.

A análise dos resultados também apontou desafios no processamento de histórias mais longas, sugerindo que a geração de conteúdo textual por IA pode apresentar limitações quando feita em uma única requisição. O problema de falta de consistência narrativa e variação no número de cenas geradas indica que abordagens segmentadas podem ser mais eficazes para a produção de textos extensos e interativos. Esse insight pode ser útil para outras pesquisas na área de geração de conteúdo automatizado, especialmente no contexto de jogos narrativos, chatbots educativos e storytelling interativo.

Além das questões técnicas, a experiência de desenvolvimento também demonstrou que o design da interface deve ser idealizado de forma adaptativa às mudanças da ferramenta. A decisão de substituir a escolha exata do número de cenas por categorias como "pequena", "normal" e "longa" surgiu como uma estratégia de minimizar frustrações geradas pela imprevisibilidade dos modelos de IA. Esse tipo de adaptação pode ser explorado em futuras pesquisas sobre UX (User Experience) para aplicações que utilizam inteligência artificial de forma interativa, considerando

formas de apresentar variações inesperadas de IA sem comprometer a experiência do usuário.

Dessa forma, este estudo não apenas resultou no desenvolvimento de uma ferramenta funcional, mas também gerou conhecimentos que podem ser aplicados em pesquisas futuras sobre uso de IA na educação, engenharia de prompt, geração automatizada de conteúdo e design de interfaces para sistemas interativos.

## 10.1. Trabalhos Futuros

Com base nas observações feitas ao longo deste estudo, diversas oportunidades de aprimoramento e expansão da ferramenta podem ser exploradas. Entre as possibilidades mais relevantes, destacam-se:

### 10.1.1. Otimização da Geração de Histórias

Apesar de seu potencial, algumas limitações técnicas precisam ser superadas para garantir que a ferramenta atinja maior precisão e qualidade narrativa. Atualmente, a geração de histórias ocorre por meio de uma única requisição à API do ChatGPT, o que pode comprometer a coesão do enredo, especialmente em narrativas mais longas. Uma possível solução seria dividir o processo de geração em etapas sequenciais, começando pela criação do contexto da história, seguida pela definição dos personagens e, por fim, pela construção das cenas.

Essa abordagem pode aumentar a fluidez e a consistência narrativa, permitindo que cada etapa da história seja desenvolvida de maneira mais estruturada. Além disso, a divisão do processo pode impactar positivamente na qualidade das narrativas geradas pela LLM, tornando a geração das histórias mais eficiente. No entanto, essa mudança traria um desafio adicional: o aumento do consumo da API do ChatGPT, uma vez que múltiplas requisições seriam necessárias para cada história gerada. Como o uso da API é tarifado com base no número de chamadas, essa estratégia exigiria uma avaliação cuidadosa do custo-benefício.

### 10.1.2. Ampliação das Opções de Personalização

Embora a ferramenta permita que os usuários escolham o tema da história e a aparência dos personagens, uma possível evolução seria oferecer maior controle

sobre o estilo narrativo, o tom do enredo e o nível de interatividade das histórias. Essa personalização pode tornar a ferramenta mais adaptável a diferentes contextos educacionais, permitindo que professores escolham entre narrativas mais formais, didáticas ou lúdicas, dependendo do objetivo pedagógico.

Além disso, pode-se explorar a possibilidade de intervenção direta dos usuários no processo de geração da história, permitindo ajustes manuais antes da finalização do texto. Isso daria aos professores e alunos mais autonomia na construção de narrativas personalizadas e alinhadas aos conteúdos trabalhados em sala de aula.

### 10.1.3. Estudos sobre UX em Ferramentas Baseadas em IA

Durante o desenvolvimento da ferramenta, percebeu-se que decisões de design de interface podem influenciar diretamente a percepção dos usuários sobre a confiabilidade e a previsibilidade da IA. A adaptação do sistema para oferecer opções de tamanho de história em categorias genéricas ("pequena", "normal" e "longa") demonstrou ser uma estratégia eficaz para lidar com as variações na quantidade de cenas geradas pelo ChatGPT.

No entanto, ainda há espaço para aprimoramentos na experiência do usuário (UX), especialmente no que diz respeito à forma como as histórias são apresentadas e editadas. Estudos futuros podem investigar como os usuários interagem com o sistema, quais dificuldades encontram e quais melhorias podem ser implementadas para tornar a experiência mais intuitiva e acessível.

## Referências

Squire, K. D. (2003). Video games and learning: What current research says. Educational researcher.

Wei, J., Wang, X., Schuurmans, D., et al.: Chain-of-thought prompting elicits reasoning in large language models (2023)

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is all you need. In *Advances in Neural Information Processing Systems*.

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., ... & Amodei, D. (2020). Language models are few-shot learners. In *Advances in Neural Information Processing Systems*.

Montfort, N. (2003). *Twisty Little Passages: An Approach to Interactive Fiction*. MIT Press.

Clark, C., Lin, K., Smith, E. M., & Gardner, M. (2021). Creative Writing with a Machine in the Loop: Case Studies on Slogans and Stories. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

Koenitz, H., Ferri, G., Haahr, M., Sezen, D., & Sezen, T. I. (2015). *Interactive Digital Narrative: History, Theory, and Practice*. Routledge.

Walton, T., Crossley, S., & McNamara, D. S. (2021). The Impact of AI Dungeon on Narrative Writing. In *Proceedings of the 2021 Workshop on Intelligent Narrative Technologies*.

Gee, J. P. (2003). What Video Games Have to Teach Us About Learning and Literacy. *Computers in Entertainment (CIE)*.

Schrier, K. (2014). *Ethics and Game Design: Teaching Values Through Play*. IGI Global.

Liu, P., Yuan, W., Fu, J., Jiang, Z., Hayashi, H., & Neubig, G. (2021). Pre-train, Prompt, and Predict: A Systematic Survey of Prompting Methods in Natural Language Processing.

Djaouti, D., Alvarez, J., Jessel, J. P., & Rampnoux, O. (2011). Origins of Serious Games. In *Serious Games and Edutainment Applications*. Springer.

Young, M. F., Slota, S., Cutter, A. B., Jalette, G., Mullin, G., Lai, B., ... & Yukhymenko, M. (2012). Our Princess Is in Another Castle: A Review of Trends in Serious Gaming for Education. *Review of Educational Research*.

Bender, E. M., Gebru, T., McMillan-Major, A., & Shmitchell, S. (2021). On the Dangers of Stochastic Parrots: Can Language Models Be Too Big? In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*.

Shute, V. J., Rieber, L. P., & Van Eck, R. N. (2017). *Games... and Learning*. In R. E. Mayer (Ed.), *Handbook of Research on Learning and Instruction*.

PRENSKY, Marc. *Aprendizagem baseada em jogos digitais*. São Paulo: Senac São Paulo, 2012.

PLASS, Jan L.; HOMER, Bruce D.; KINZER, Charles K. *Foundations of game-based learning*. *Educational Psychologist*, 2015. Disponível em: <https://files.eric.ed.gov/fulltext/EJ1090277.pdf>. Acesso em: 20 jan. 2025.

LATITUDE. *AI Dungeon*. Disponível em:

<https://aidungeon.com>. Acesso em: 21 jan. 2025.

MANAGEMENT SOLUTIONS. *A ascensão dos Large Language Models: dos fundamentos à aplicação*. 2024. Disponível em:

<https://www.managementsolutions.com/sites/default/files/minisite/static/72b0015f-39c9-4a52-ba63-872c115bfbd0/llm/pdf/ascensao-dos-llm.pdf>. Acesso em: 21 jan. 2025.

CULTE. *NovelAI: inteligência artificial para escrita criativa*. Disponível em:

<https://blog.culte.com.br/novelai-inteligencia-artificial-para-escrita-criativa>. Acesso em: 21 jan. 2025.

WECLEVER. *Como treinar uma IA? Aprenda tudo sobre LLMs*. Disponível em:

<https://blog.weclever.com/llm>. Acesso em: 21 jan. 2025.

## APÊNDICE A - Prompt

```

const prompt = [
  {
    role: "assistant",
    content: `Regras obrigatórias para a criação da história:
    - A história deve ter um início, meio e fim.
    - Cada interação feita deve impactar no desenrolar da
história.
    - O diálogo deve ser coerente e seguir uma linha de
raciocínio.
    - Deve conter ${params.amountScenes} sceneCharacters.
    - Nem todas as cenas precisam ter interações.
    - A história deve incluir no mínimo 2 interações.
    - O retorno deve ser no formato JSON.`
  },
  {
    role: "system",
    content: `O formato de saída da história deve ser em JSON.
enum CharacterEmotionEnum {
  HAPPY = "happy",
  SAD = "sad",
  SURPRISED = "surprised",
  THINKING = "thinking",
  CONFUSED = "confused",
  VERY_HAPPY = "very-happy",
  NEUTRAL = "neutral",
  FRUSTRATED = "frustrated",
  EXCITED = "excited",
}

enum ScenePositionEnum {
  LEFT = "left",
  RIGHT = "right",
  CENTER = "center",
}

export enum CharacterGenderEnum {
  MALE = "MALE",
  FEMALE = "FEMALE",
}

interface Character {
  id: string;
  name: string;
  role: string;
  position: ScenePositionEnum;
  avatarUrl: string; // no implement
  gender: CharacterGenderEnum;
}

interface ISceneCharacter {

```

```

    id: string;
    characterId: Character["id"];
    order: number;
    speech: string;
    emotion: CharacterEmotionEnum;
    avatarUrl: string; // no implement
    interaction?: IUserInteraction;
  }

interface IStory {
  id: string;
  theme: string;
  title: string;
  intro: string;
  summary: string;
  backgroundUrl: string; // no implement

  characters?: Character[];
  sceneCharacters?: ISceneCharacter[];
}

interface IUserInteraction {
  id: string;
  storyId: IStory["id"];
  sceneCharacterId: ISceneCharacter["id"];

  sentence: string;
  options: IUserInteractionOption[];
}

interface IUserInteractionOption {
  id: string;
  sceneCharacterId: ISceneCharacter["id"];
  nextSceneCharacterId: ISceneCharacter["id"];
  label: string;
  feedback: string;`,
},
{
  role: "user",
  content: `Crie uma história interativa sobre o tema
${params.theme}, na qual acontece um diálogo entre
${this.amountCharacter} personagens.`
},
];

const openAI = this.chatGPTApiClient.getClient();
const response = await openAI.chat.completions.create({
  messages: prompt,
  model: this.OPENAI_MODEL,
  response_format: { type: "json_object" },
});

```

Fonte: próprio autor