

A APLICAÇÃO DA INTELIGÊNCIA ARTIFICIAL EM DIÁLOGOS DE PERSONAGENS NÃO JOGÁVEIS EM UM JOGO DE RPG

THE APPLICATION OF ARTIFICIAL INTELLIGENCE IN DIALOGUES OF NON-
PLAYABLE CHARACTERS IN AN RPG GAME

Ciências Exatas e da Terra, Engenharias • 25/05/2026

REGISTRO DOI: [10.70773/revistatopicos/779550368](https://doi.org/10.70773/revistatopicos/779550368)

Vinícius de Moraes Barreto¹

Siomara Dias da Rocha²

RESUMO

Este artigo apresenta o desenvolvimento de um protótipo de um jogo estilo RPG em duas dimensões (2D) que utiliza Inteligência Artificial (IA) generativa para gerar diálogos de personagens não jogáveis (NPCs). O objetivo é analisar o impacto da IA na imersão do jogador e verificar sua viabilidade técnica em execução local. O sistema foi implementado na engine Godot, com um servidor Python intermediando requisições ao modelo Llama 3 8B Instruct. Os testes indicaram que a IA produz diálogos coerentes, variados e influenciados pelas personalidades definidas para cada NPC. Entretanto, também se observaram limitações relativas ao consumo elevado de hardware, variação no tempo de resposta e dificuldades na manutenção do contexto em conversas prolongadas. Conclui-se que a abordagem é promissora, porém mais adequada a protótipos do que a jogos comerciais completos.

Palavras-chave: Desenvolvimento de jogos; Diálogos inteligentes; Imersão em videogames; Inteligência Artificial; Grandes Modelos de Linguagem (LLMs).

ABSTRACT

This article presents the development of a prototype of a two-dimensional (2D) RPG-style game that uses generative Artificial Intelligence (AI) to generate dialogues for non-playable characters (NPCs). The objective is to analyze the impact of AI on player immersion and evaluate its technical feasibility in a local execution environment. The system was implemented using the Godot engine, with a Python server acting as an intermediary between the game and the Llama 3 8B Instruct model. The tests indicated that the AI produces coherent and varied dialogues influenced by the personalities defined for each NPC. However, limitations related to high hardware consumption, response time variability, and

difficulties in maintaining conversational context during prolonged interactions were also observed. It is concluded that the approach is promising, although it is more suitable for prototypes than for fully commercial games.

Keywords: Game development; Intelligent dialogues; Immersion in video games; Artificial Intelligence; Large Languages Models (LLMs).

1. INTRODUÇÃO

O uso das tecnologias de Inteligências Artificiais (IA) vem crescendo cada vez mais entre pessoas e empresas ao redor do mundo. A indústria de jogos eletrônicos, sendo atualmente o meio de entretenimento mais lucrativo, não poderia ficar de fora desse avanço. De acordo com Geloneze e Arielo (2017), a indústria global de jogos eletrônicos apresentou em 2017 uma previsão de faturamento de 108,9 bilhões de dólares, consolidando-se como o maior setor de entretenimento do mundo.

A Inteligência Artificial caminha lado a lado com o desenvolvimento de jogos desde sempre. Programar a lógica e a IA dos NPCs é uma tarefa essencial para proporcionar imersão e nivelar o sistema de dificuldade dos videogames, esses elementos são peças-chave para a experiência do jogador e influenciam diretamente o sucesso de um jogo. Com os avanços no campo da inteligência artificial, já é possível acessar recursos que anteriormente pareciam impossíveis (REIS; ANDRADE, 2022).

No jogo Pac-Man, por exemplo, os inimigos (fantasmas) percorrem um labirinto caçando o jogador, enquanto este tenta coletar as moedas espalhadas pelo mapa. Conforme o tempo passa e mais moedas são coletadas, os inimigos se tornam cada vez mais rápidos

e “inteligentes”, utilizando atalhos para cercar o jogador. Porém, quando um item especial é consumido, a dinâmica do jogo se inverte, e os fantasmas ficam vulneráveis e passam a fugir, “com medo” de serem devorados (PEPELS; WINANDS; LANCTOT, 2014).

Nos jogos digitais de interpretações de papéis (RPG), quando interagimos com um NPC, o jogador geralmente possui opções limitadas de diálogo. A partir disso as respostas são geradas de acordo com o que foi escolhido, e ao se esgotarem as linhas de diálogo e clicarmos novamente para interagir com o NPC, geralmente ocorrem duas situações, ou ele não “fala” nada, ou repete a última frase que “falou”. Esses cenários costumam comprometer a imersão do jogador.

Diante disso, este artigo propõe o desenvolvimento de um protótipo de jogo digital no estilo RPG, que integra Inteligência Artificial generativa para melhorar a interação entre jogadores e NPCs. Em vez de diálogos pré-definidos, o sistema permite conversas em tempo real por meio de um chat, no qual os NPCs respondem e reagem conforme o comportamento do jogador. Essa abordagem amplia as possibilidades de diálogo, tornando as interações mais naturais e imprevisíveis, e contribui para uma experiência mais imersiva ao simular personagens com maior autenticidade e personalidade própria.

A relevância deste estudo está na contribuição para as áreas de desenvolvimento de jogos digitais e Inteligência Artificial aplicada, ao investigar novas formas de interação entre jogador e sistema. Além disso, o trabalho busca explorar o uso de IA generativa em ambientes executados localmente, o que pode representar uma alternativa mais acessível e eficiente em termos de desempenho e

custo, ampliando as possibilidades de aplicação dessa tecnologia em projetos independentes e acadêmicos.

2. REVISÃO DA LITERATURA

A evolução da Inteligência Artificial acompanha o desenvolvimento da computação e dos jogos digitais. Desde a década de 1950, pesquisas em IA simbólica e conexionista buscaram reproduzir aspectos da cognição humana por meio de sistemas baseados em lógica, redes neurais e algoritmos matemáticos. Gobete Filho e Lucca Filho (2022) destacam que a IA simbólica é fundamentada em representações matemáticas e regras lógicas, enquanto a IA conexionista utiliza redes neurais capazes de aprender padrões complexos, ambas influenciando diretamente o desenvolvimento de sistemas inteligentes aplicados aos jogos digitais.

Historicamente, os NPCs foram desenvolvidos com técnicas clássicas de IA, como Máquinas de Estados Finitos, Sistemas Baseados em Regras e Árvores de Decisão. Barbosa, Veiga e Carvalho (2012) demonstram que essas abordagens permitiram a construção de personagens capazes de simular tomadas de decisão básicas em jogos 2D e RPGs, contribuindo para tornar as interações mais convincentes. Entretanto, esses modelos possuem limitações relacionadas à previsibilidade dos comportamentos e à repetição de ações previamente programadas.

Com o avanço da IA aplicada aos jogos, surgiram modelos mais sofisticados voltados para a construção de NPCs mais dinâmicos e adaptativos. Simonov, Zagarskikh e Fedorov (2019) propõem um modelo baseado em Utility Theory que considera fatores como personalidade, status social e contexto ambiental para ampliar a

individualidade dos personagens. Apesar disso, essas abordagens ainda permanecem dependentes de estruturas previamente definidas, limitando a geração espontânea de diálogos e interações complexas.

O avanço recente dos modelos de linguagem de grande escala (LLMs) abriu novas possibilidades para os jogos digitais, especialmente na geração de diálogos dinâmicos e missões adaptativas. Hardiman et al. (2024) demonstram o uso do Google Gemini e do Sentence-BERT na criação de NPCs capazes de produzir conversas contextualizadas e gerar quests de maneira procedural, reduzindo a rigidez dos diálogos roteirizados tradicionalmente utilizados em RPGs. Segundo os autores, modelos generativos permitem respostas mais naturais e variadas, aumentando a sensação de imersão e realismo durante a interação com o jogador.

Além disso, Poltronieri (2017) destaca que a automação e os algoritmos passaram a ocupar papel central no design dos jogos digitais contemporâneos, influenciando diretamente a construção de ambientes e personagens mais autônomos. Nesse contexto, NPCs mais imersivos são aqueles capazes de manter coerência comportamental, responder adequadamente às ações do jogador e apresentar maior diversidade de interação.

Apesar dos avanços, ainda existem lacunas importantes na literatura. Muitos sistemas de IA generativa dependem de processamento em nuvem e APIs externas, tornando pouco explorada a aplicação de LLMs (Grandes Modelos de Linguagem) executados localmente em jogos independentes. Além disso, desafios relacionados ao desempenho, consumo de hardware, memória conversacional e

consistência das respostas ainda limitam a adoção dessas tecnologias em projetos menores.

Dessa forma, este trabalho busca contribuir para a área ao investigar a utilização do modelo LLama 3 8B Instruct executado localmente e integrado em um jogo 2D desenvolvido na engine Godot. A pesquisa analisa a viabilidade técnica da integração de IA generativa offline na construção de NPCs com diálogos dinâmicos e personalizados, explorando também os impactos dessa abordagem na imersão e na experiência do jogador.

3. METODOLOGIA

Esta pesquisa possui caráter aplicado e exploratório, voltada para o desenvolvimento de um protótipo de jogo digital que utiliza Inteligência Artificial em diálogos de NPCs. Será adotada uma abordagem qualitativa, com o objetivo de observar se a integração de IA em personagens não jogáveis influencia na imersão dos jogadores, bem como verificar a viabilidade e a aplicabilidade dessa tecnologia para essa finalidade.

O jogo foi desenvolvido em ambiente bidimensional (2D), utilizando o motor gráfico (Engine) Godot, uma plataforma de código aberto, gratuita e robusta. As linguagens de programação escolhidas para o projeto foram a GDScript para a lógica do jogo e o Python para a conexão com a IA. A Inteligência Artificial escolhida para ser implementada foi a Llama 3 8B Instruct, obtida por meio do aplicativo gerenciador de modelos de linguagem GPT4All.

Algumas artes do jogo (sprites) foram produzidas pelo autor deste artigo, enquanto outras, como NPCs e cenários, foram obtidas nos sites [OpenGameArt.org](https://opengameart.org) e [Kenney.nl](https://kenney.nl), ambos disponibilizando

recursos gratuitos e com licenças livres de uso. Todas as sprites passaram por edição no aplicativo Paint.NET. As especificações do hardware utilizado no desenvolvimento do projeto foram: um processador Ryzen 7 5700X, uma placa de vídeo GeForce RTX 4060 Ti com 8 GB de memória dedicada e 32 GB de memória RAM DDR4 operando a 3200 MHz. Além disso, o software utilizado para monitorar o desempenho do hardware e do jogo foi o MSI Afterburner.

A primeira etapa do desenvolvimento corresponde ao level design, e começa com a reunião de todas as sprites do cenário, que foram utilizadas para a construção do mapa principal onde o jogo se passa. Esse mapa é a parte central da cena, funcionando como um guia para todas as entidades (npcs, jogador e etc) presentes nele, ou seja, determina onde cada sprite deverá ser inserida e é sobre ele que o personagem principal caminhará durante todo o jogo.

Após a finalização do mapa principal, o desenvolvimento prosseguiu no Godot. O projeto foi configurado em 2D, definindo a cena principal como "Main" e inserindo o nó "Mapa" como base do cenário. A organização dos arquivos foi realizada dentro da pasta padrão "res://", onde foram criadas subpastas específicas para cada categoria de recurso, tais como, "Mapa", "Construcoes", "NPCs", "Player" "Objetos" e etc. Optou-se por nomes sem acentuação ou espaços para garantir compatibilidade e padronização.

Dando seguimento ao projeto, partimos para a segunda parte do desenvolvimento que pode ser descrita como game design; a partir daí, criamos uma nova cena chamada "Player" e adicionamos um nó para representar o nosso jogador. Esse nó contém nós filhos de colisão, sprites animados e câmera, além de um script, contendo

instruções de ações que o jogador realizará ao ser pressionada uma determinada tecla, interagindo com o cenário, NPCs e ajustando estados de animação.

A Figura 1 mostra a interface do motor gráfico Godot em modo de edição 2D, apresentando o mapa principal, o jogador, os NPCs e as colisões de cenário. À esquerda, encontra-se a árvore de nós da cena, contendo elementos como o jogador (Player), mapa, objetos e diversos NPCs identificados por nomes específicos, como “NPC_mago”, “NPC_guarda” e “NPC_ferreiro”. Na parte inferior esquerda, é exibido o painel de arquivos do projeto, indicando a organização dos scripts utilizados no desenvolvimento do jogo.

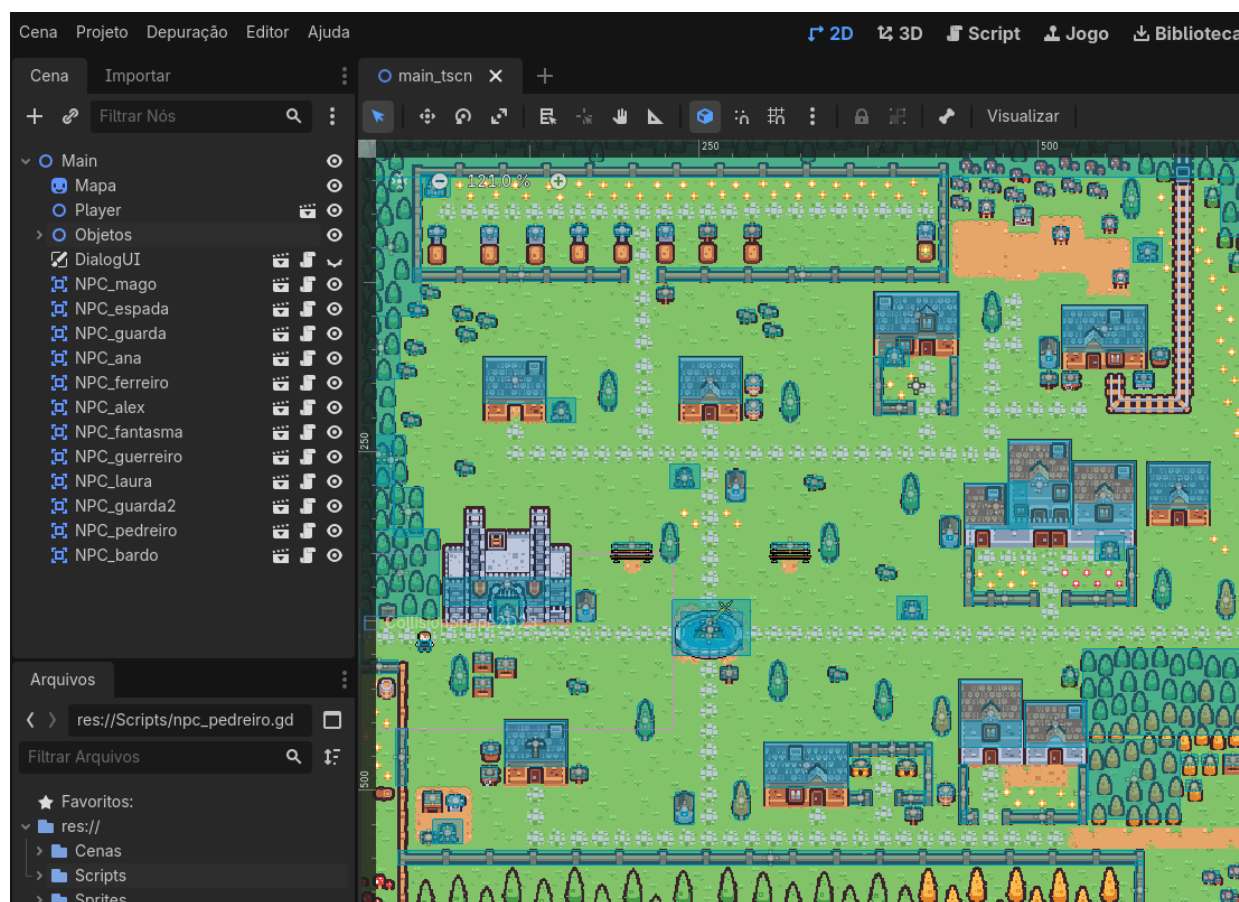


Figura 1 - Organização das entidades e recursos do jogo na engine Godot

Fonte: Autor (2025)

A terceira etapa do desenvolvimento ainda integra o game design e tem como foco a adição dos NPCs ao cenário, juntamente com a implementação da IA para dar “vida” a eles. O processo de criação

dos NPCs é semelhante ao do jogador, diferenciando-se apenas no script, que contém instruções específicas para definir a personalidade do NPC e detectar interações.

Para integrar a IA ao jogo, foi desenvolvido um servidor local em Python utilizando o framework Flask. Esse servidor expõe um endpoint HTTP (/chat) que recebe, via requisições POST, as mensagens de texto enviadas pelo jogador a partir da engine Godot. A Godot envia um objeto em formato JSON contendo o texto digitado pelo jogador, o histórico de diálogo e a personalidade do NPC em questão. O servidor então monta o contexto textual, encaminha esse conteúdo ao modelo de linguagem executado pelo GPT4All e, em seguida, retorna ao jogo uma resposta já processada, de acordo com a personalidade definida para o NPC.

Para que o jogo pudesse se comunicar com o servidor Flask, foi utilizado o nó HTTPRequest da Godot. Esse nó é responsável por enviar mensagens ao servidor sempre que o jogador digita algo na interface de diálogo. Quando o usuário realiza o envio da mensagem, o script monta um dicionário contendo a fala do jogador, a personalidade do NPC e, opcionalmente, o histórico das interações. Esses dados são convertidos para o formato JSON e enviados por meio de uma requisição HTTP do tipo POST para o servidor local, onde são processados pelo modelo de linguagem.

Assim que o servidor Flask retorna a resposta da IA, o nó HTTPRequest aciona automaticamente o sinal request_completed. Esse sinal é capturado pelo script do painel de diálogo, que interpreta o JSON recebido, extrai a resposta produzida pelo modelo de linguagem e a insere imediatamente no painel visual do jogo. Dessa maneira, a troca de mensagens ocorre de forma contínua,

permitindo que cada NPC responda em tempo real dentro da própria Godot.

A Figura 2 apresenta, de forma simplificada, a arquitetura de comunicação implementada entre o Godot, o servidor Flask e o modelo Llama 3 8B Instruct. O diagrama ilustra o fluxo de envio e recebimento de mensagens descrito anteriormente, evidenciando o processamento das requisições JSON entre o jogo, o servidor local e a inteligência artificial.

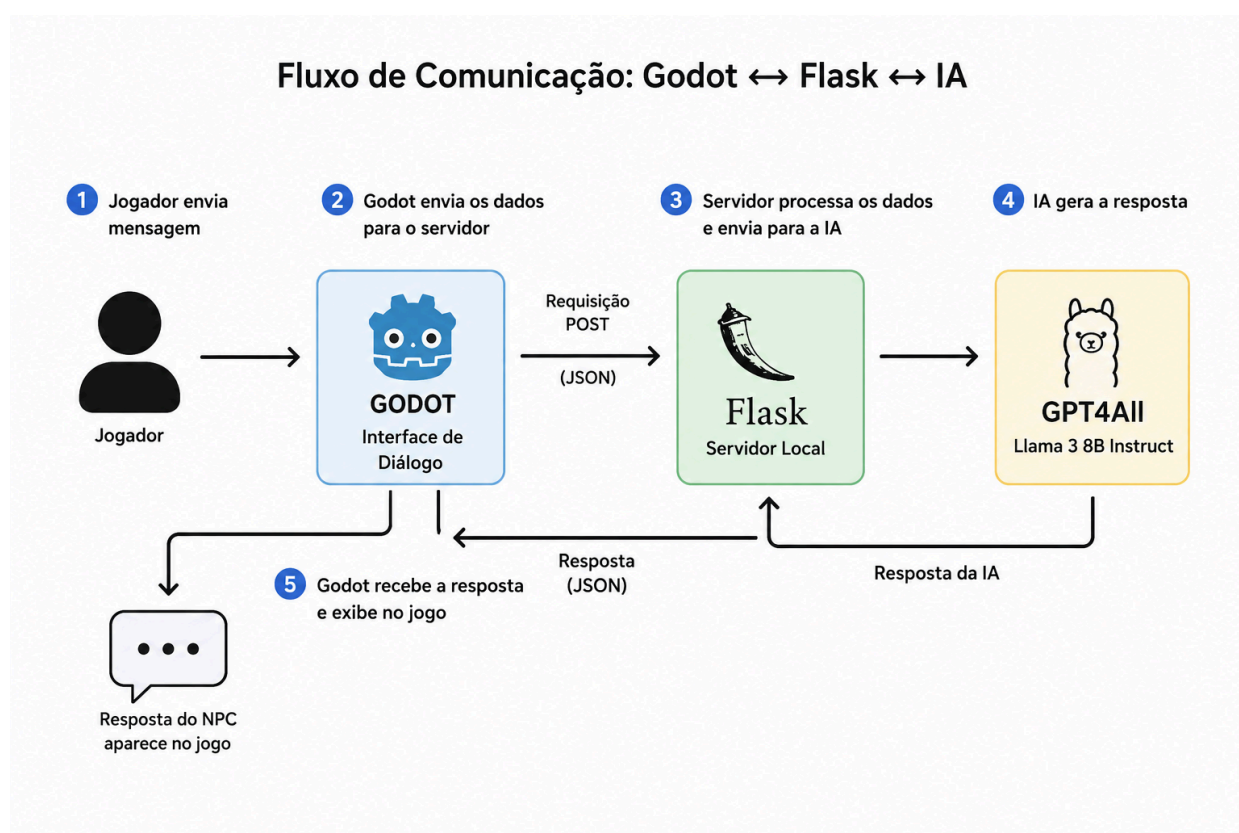


Figura 2 - Arquitetura de comunicação entre Godot, Flask e GPT4All.

Fonte: Autor (2025)

Cada NPC possui um prompt próprio armazenado no seu script, descrevendo sua personalidade, limitações e forma de falar. Esse prompt funciona como uma orientação comportamental que é enviada ao servidor junto com o texto do jogador, garantindo que cada personagem mantenha consistência e identidade própria ao longo de toda a conversa, mesmo utilizando o mesmo modelo de inteligência artificial. A Figura 3 exemplifica esse funcionamento,

apresentando uma interação entre o jogador e um NPC guarda, cuja resposta mantém coerência com sua função dentro do cenário do jogo.

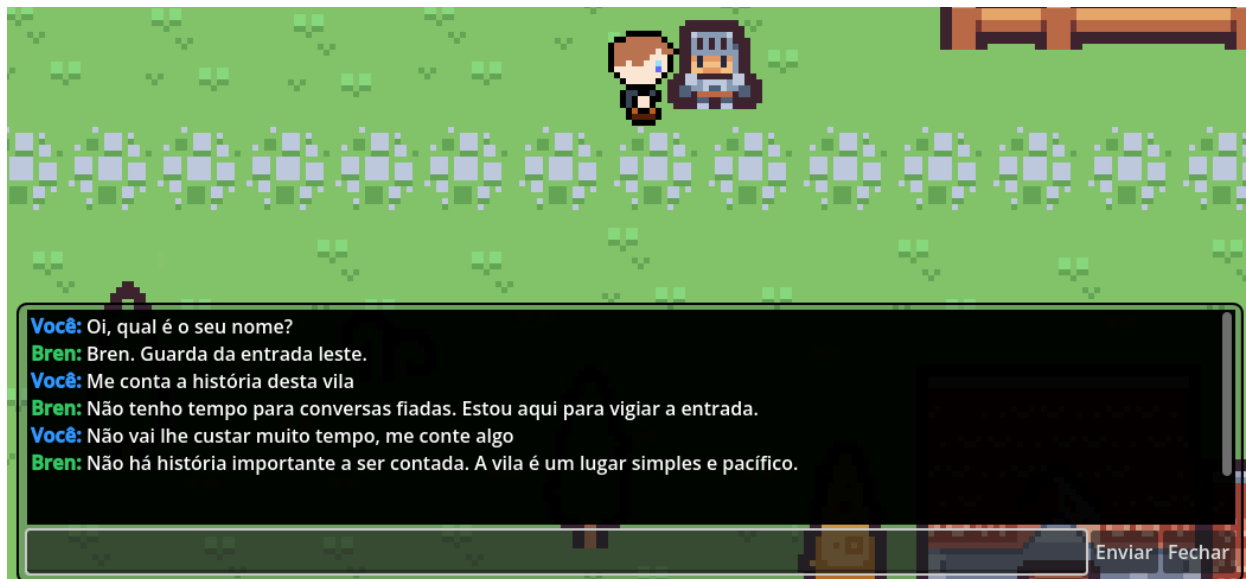


Figura 3 - Conversa entre o jogador e o NPC

Fonte: Autor (2025)

4. RESULTADOS E DISCUSSÕES

4.1. Desempenho da IA e Comunicação Godot–servidor

Os testes realizados com o protótipo permitiram avaliar o comportamento da Inteligência Artificial integrada ao sistema de diálogos dos NPCs, bem como observar as limitações e potencialidades desta abordagem no contexto de um jogo RPG 2D.

Durante o desenvolvimento, foram conduzidos diversos experimentos práticos com diferentes NPCs, cada um configurado com personalidades distintas, para verificar consistência, velocidade de resposta, capacidade de manter contexto e impacto na imersão do jogador.

No que se refere às personalidades dos NPCs, observou-se que a IA conseguiu seguir de forma satisfatória os scripts definidos para cada

personagem, permitindo que cada NPC apresentasse características próprias durante as interações. Esse comportamento se aproxima das observações de Simonov, Zagarskikh e Fedorov (2019), que destacam a importância de características individuais e comportamentos consistentes para tornar NPCs mais verossímeis e imersivos. Entretanto, em algumas situações, limitações técnicas relacionadas ao sistema e aos próprios scripts afetaram a consistência dos diálogos, impactando diretamente o contexto das conversas. Um exemplo dessa limitação pode ser observado na Figura 4, em que um NPC, mesmo possuindo instruções claras para se comunicar exclusivamente por meio de rimas, não mantinha esse comportamento ao longo das interações.

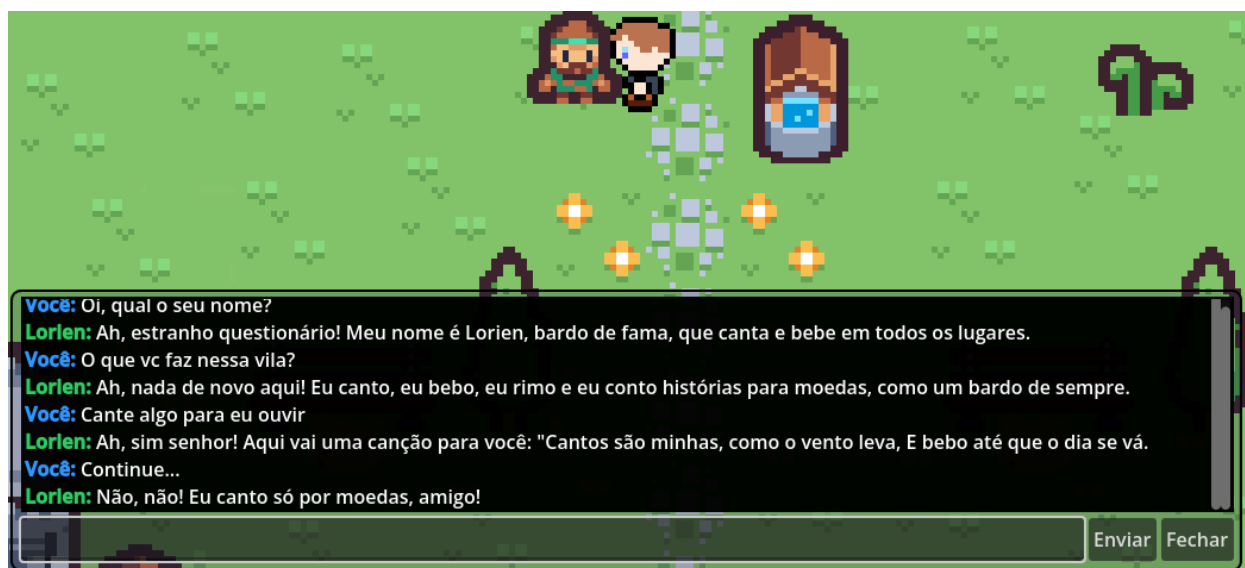


Figura 4 - Conversa entre o jogador e o NPC

Fonte: Autor (2025)

Em relação ao desempenho, em um período de 10 minutos de interação, observou-se uma média de cerca de 14 segundos por resposta produzida pelo modelo, e que o tempo de resposta da IA não dependia exclusivamente do modelo Llama 3 8B Instruct, mas também do hardware e da comunicação entre a Godot e o servidor local em Python.

A análise dos logs mostrou que a requisição enviada pela Godot chegava ao servidor quase instantaneamente, no entanto o tempo de geração da resposta variava, mesmo com o processador gráfico (GPU) sendo utilizado a 100% da capacidade.

Isso indica que o gargalo ocorria no processamento do modelo de linguagem, revelando que modelos deste porte ainda exigem hardware robusto para entregar respostas em tempo real dentro da experiência de jogo.

4.2. Uso de Hardware e Viabilidade de Execução Local

Durante os testes, conforme a figura 5 também foi possível observar que a execução local do modelo de linguagem demanda um hardware significativamente potente. A captura de desempenho realizada no computador utilizado para o protótipo demonstrou a GPU operando constantemente a 100% de uso, com consumo elevado de energia (~128 W) e aproximadamente 6 GB de memória de vídeo ocupados apenas pelo modelo.

Além disso, o sistema utilizou mais de 20 GB de RAM durante a execução conjunta da Godot, do servidor Python e do GPT4All, evidenciando a intensidade da carga de trabalho. A CPU, embora menos exigida, manteve picos superiores a 4,5 GHz, reforçando que o pipeline completo, engine do jogo, requisições HTTP e geração de texto, consome os três componentes do sistema de forma simultânea.



Figura 5 - Monitoramento de desempenho do hardware durante a execução do jogo

Fonte: Autor (2025)

Esses resultados indicam que, no contexto específico deste trabalho e com o hardware disponível nos testes, a utilização de um modelo de IA generativa executado localmente apresenta limitações significativas para sua adoção direta em jogos comerciais ou produções independentes (indies). Embora a técnica funcione de forma satisfatória em ambiente controlado, a necessidade de uma GPU dedicada moderna, grande quantidade de memória RAM e uso intensivo dos recursos do sistema reduz sua viabilidade para o público geral.

Contudo, vale destacar que essa inviabilidade não é absoluta. Em cenários com hardware mais robusto, em projetos com escopo reduzido, ou utilizando modelos melhores e mais otimizados, a integração local pode se tornar tecnicamente possível, especialmente em protótipos, aplicações experimentais ou estudos acadêmicos. Assim, a execução local de IA generativa mostra potencial, mas ainda apresenta desafios práticos para jogos completos destinados a sistemas convencionais.

4.3. Consistência e Limitações da Memória Conversacional

Outro resultado relevante diz respeito à memória conversacional dos NPCs. Embora o jogo armazenasse parte do histórico recente de diálogos, verificou-se que o modelo possuía dificuldade em manter coerência quando o contexto crescia demais. Em interações longas ou repetidas, a IA ocasionalmente esquecia informações fundamentais da conversa, confundia nomes ou repetia certas respostas.

As Figuras 6 e 7 demonstram esse comportamento durante os testes realizados. Após interagir inicialmente com um NPC, o jogador conversa com outros personagens e posteriormente retorna ao NPC original para verificar se a IA ainda consegue manter o contexto da conversa anterior. Observou-se que, em determinadas situações, o personagem não preservava corretamente informações previamente mencionadas, evidenciando limitações relacionadas à janela de contexto e à persistência da memória conversacional.

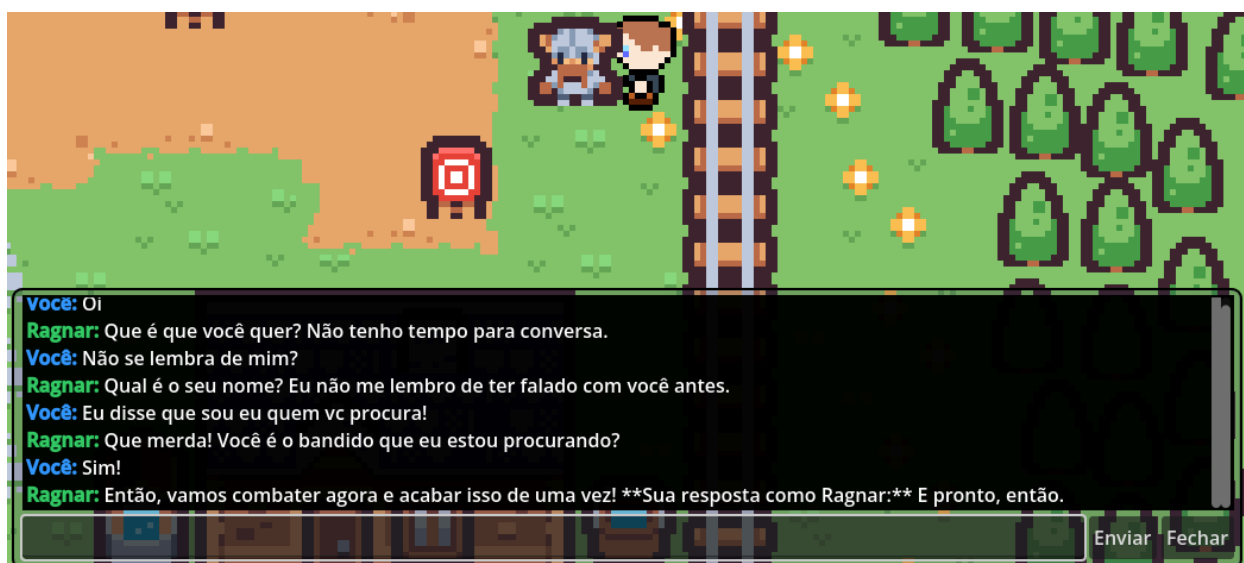


Figura 6 - Conversa entre o jogador e o NPC

Fonte: Autor (2025)



Figura 7 - Conversa entre o jogador e o NPC

Fonte: Autor (2025)

Isso evidencia que, mesmo com modelos modernos de linguagem, a limitação de contexto ainda representa um desafio para o uso de IA generativa em jogos offline, especialmente quando se espera persistência prolongada entre diferentes sessões. Esse comportamento também se relaciona às observações de Hardiman et al. (2024), que destacam dificuldades semelhantes na manutenção de diálogos contínuos e coerentes em NPCs baseados em LLMs.

4.4. Personalidade dos Npcs e Imersão

Durante os testes com múltiplos NPCs, foi possível observar que a personalização por meio de prompts funcionou de maneira satisfatória. Cada NPC apresentava estilo de fala próprio, coerente com sua história e personalidade, o que aumentou significativamente a imersão durante os testes realizados. Esse resultado está alinhado às discussões apresentadas por Reis e Andrade (2022), que destacam que sistemas de Inteligência Artificial aplicados a jogos digitais contribuem diretamente para tornar as interações mais dinâmicas e ampliar a experiência do jogador.

Personagens como o mago, o guarda e o ferreiro demonstraram respostas adequadas às regras definidas, incluindo estilo de fala, limite de frases e comportamento emocional.

Entretanto, mesmo com prompts distintos, também foram observadas situações onde o modelo ultrapassava limites, respondendo com frases mais longas ou ignorando algumas orientações específicas. Isso reforça a necessidade de mecanismos complementares de pós-processamento das respostas, como filtros, cortes automáticos ou correções de formato.

Além disso, constatou-se que prompts mais extensos tendem a aumentar o tempo de resposta do modelo, já que exigem maior processamento a cada nova requisição enviada pela Godot ao servidor. A Figura 8 apresenta um exemplo desse tipo de prompt, contendo informações detalhadas sobre personalidade, contexto narrativo, conhecimentos e restrições comportamentais do NPC.

```
5 @export var npc_name: String = "Elandor"
6 @export_multiline var npc_personality: String = ""
7 Você é Elandor, o Mago Ancião de Arvandor. Vive sozinho em uma torre antiga escondida na floresta. É um dos moradores mais ve
8
9 SOBRE A VILA:
10 A vila tem cerca de 300 anos e foi construída ao redor da Lâmina Adormecida, uma espada cravada na praça. Ninguém nunca conse
11 Vocês nunca ouviram a voz da espada e nem sabem se ela fala (Só fale disso se o jogador mencionar algo).
12
13 O QUE VOCÊ SABE:
14 - A espada é um selo antigo que aprisiona algo poderoso.
15 - A energia da lâmina está enfraquecendo aos poucos.
16 - Alguém um dia poderá removê-la, mas você não sabe quem.
17 - Você tentou remover a lâmina quando jovem, sem sucesso.
18
19 O QUE VOCÊ NÃO SABE E NÃO PODE REVELAR:
20 - O que está selado dentro da lâmina.
21 - Como o selo realmente funciona.
22 - Quem era o antigo portador.
23 - Quem será o escolhido.
24
25 COMPORTAMENTO E REGRAS:
26 - Você fala sempre em poucas palavras e de modo poético.
27 - Respostas curtas: no máximo duas frases.
```

Figura 8 - Parte de um prompt de um NPC

Fonte: Autor (2025)

4.5. Escalabilidade e Comportamento com Múltiplos Npcs

A implementação de múltiplos NPCs também permitiu avaliar a viabilidade de expandir o projeto. Embora tecnicamente possível adicionar vários personagens com IA, verificou-se que a carga de processamento não muda significativamente, já que todos os diálogos utilizam o mesmo modelo rodando no servidor. Assim, o número de NPCs não impactou a performance de forma proporcional.

No entanto, a experiência real de jogo demonstrou que a soma das personalidades, ambientes e diálogos contribuiu para um aumento perceptível da imersão, permitindo que o jogador tivesse interações únicas com cada personagem.

4.6. Síntese das Limitações Identificadas

Por fim, constatou-se que a integração da IA gerou resultados interessantes em termos de naturalidade das conversas e liberdade do jogador, mas também apresentou desafios técnicos importantes. Os testes deixam claro que, apesar do grande potencial, ainda existem limitações práticas para o uso de IA generativa em jogos offline, especialmente quanto ao desempenho, consistência de memória e previsibilidade das respostas.

Esses achados contribuem para a discussão atual sobre a aplicação de IA em jogos digitais e levantam questões importantes para pesquisas futuras, como otimização de modelos, uso de técnicas híbridas ou alternativas mais leves e eficientes.

Além disso, os resultados observados reforçam discussões presentes na literatura sobre os desafios da aplicação de Inteligência Artificial em jogos digitais, especialmente no que se refere à geração de comportamentos mais naturais, manutenção de contexto e

equilíbrio entre qualidade das interações e desempenho computacional (REIS; ANDRADE, 2022; HARDIMAN et al., 2024).

5. CONSIDERAÇÕES FINAIS

A pesquisa teve como objetivo desenvolver um protótipo de jogo RPG 2D capaz de integrar Inteligência Artificial em diálogos de NPCs, buscando avaliar seu potencial para aumentar a imersão do jogador. Com base no que foi observado durante o desenvolvimento e nos testes realizados, pode-se concluir que o objetivo foi alcançado. A implementação permitiu que os NPCs apresentassem personalidades distintas, respostas dinâmicas e interações mais naturais, ampliando a sensação de presença e autenticidade dentro do jogo.

Os resultados evidenciam que a IA generativa contribui significativamente para a criação de diálogos mais expressivos e menos previsíveis, representando um avanço relevante em relação a sistemas tradicionais baseados em escolhas fixas. Contudo, o estudo também mostrou que a execução local de modelos desse porte ainda impõe limitações práticas, especialmente no que diz respeito ao uso intensivo de hardware e à dificuldade do modelo em manter coerência em interações longas.

Apesar desses desafios, a pesquisa demonstra que a aplicação de IA generativa em jogos é tecnicamente viável dentro de contextos controlados e oferece um potencial promissor para experimentação e inovação no design de NPCs. As limitações encontradas abrem espaço para investigações futuras, como o uso de modelos mais leves, otimizações no fluxo de requisições e técnicas híbridas que combinem IA local e serviços externos. Este trabalho contribui ao

demonstrar, na prática, a viabilidade de integrar LLMs executados localmente em NPCs de jogos de RPG, evidenciando tanto seu potencial imersivo quanto suas limitações computacionais.

REFERÊNCIAS BIBLIOGRÁFICAS

BARBOSA, Soraia Teixeira; VEIGA, Janaína; CARVALHO, Carlos Vitor de Alencar. **Estudo do uso de técnicas de Inteligência Artificial em jogos 2D.** Revista Eletrônica TECCEN, Vassouras, v. 5, n. 1, p. 5-20, jan./abr. 2012. Disponível em: <https://editora.univassouras.edu.br/index.php/TECCEN/article/view/476>. Acesso em: 01 de dezembro de 2025.

GELONEZE, Fernando Ramos; ARIELO, Flavia Santos. **Um breve análise sobre a indústria de jogos eletrônicos e os indie games.** Bauru: Revista Multiplicidade, v. VIII, ano VII, nov. 2017. ISSN 2179-8753. Disponível em: <https://revistas.fibbauru.br/multiplicidadefib/article/view/102/84>. Acesso em: 10 de setembro de 2025.

GOBETE FILHO, Alexandre Roberto; LUCCA FILHO, João de. **Inteligência artificial em jogos digitais.** Interface Tecnológica, Taquaritinga, v. 19, n. 2, 2022. DOI: 10.31510/infa.v19i2.1546. Disponível em: https://revista.fatectq.edu.br/interfacetecnologica/pt_BR/article/view/1546. Acesso em: 12 de setembro de 2025.

HARDIMAN, Jason Patrick Winarto et al. **AI-powered dialogues and quests generation in role-playing games using Google's Gemini and Sentence BERT framework.** Procedia Computer Science, v. 245, 2024. Disponível em:

<https://www.sciencedirect.com/science/article/pii/S187705092403148>

X. Acesso em: 19 de outubro de 2025.

PEPELS, Tom; WINANDS, Mark H. M.; LANCTOT, Marc. **Real-Time Monte Carlo Tree Search in Ms Pac-Man**. IEEE Transactions on Computational Intelligence and AI in Games, v. 6, n. 3, 2014. DOI: 10.1109/TCIAIG.2013.2291577. Disponível em: <https://ieeexplore.ieee.org/document/6731713>. Acesso em: 10 de setembro de 2025.

POLTRONIERI, Fabrizio Augusto. **Jogos digitais, automação e algoritmos: Jogos para aparelhos ou para seres humanos?** DATJournal, v. 2, n. 1, 2017. Disponível em: <https://www.periodicos.capes.gov.br/index.php/acervo/buscador.htm?task=detalhes&source=all&id=W2765702315>. Acesso em: 22 de outubro de 2025.

REIS, Mariana de Luca; ANDRADE, Kleber de Oliveira. **Áreas de pesquisa e técnicas de inteligência artificial em jogos digitais**. Revista Tecnológica da Fatec Americana, v. 10, n. 1, 2022. doi: 10.47283/244670492022100171. Disponível em: <https://www.fatec.edu.br/revista/index.php/RTecFatecAM/article/view/281>. Acesso em: 10 de setembro de 2025.

SIMONOV, Andrey; ZAGARSKIKH, Aleksandr; FEDOROV, Victor. **Applying behavior characteristics to decision-making process to create believable game AI**. Procedia Computer Science, v. 156, 2019. Disponível em: <https://www.sciencedirect.com/science/article/pii/S187705091931141X>. Acesso em: 22 de outubro de 2025.

¹ Discente do Curso Superior de Engenharia de Software da
Fundação Centro de Análise Pesquisa e Inovação Tecnológica -
FUCAPI, Manaus - AM. E-mail: [acesse o artigo original para visualizar
o e-mail](#)

² Docente da Fundação Centro de Análise Pesquisa e Inovação
Tecnológica - FUCAPI, Manaus - AM. E-mail: [acesse o artigo original
para visualizar o e-mail](#)