



Samuel Vidal dos Santos

# **AIDungeonMaster: Projeto e Implementação de um Mestre de Jogo Artificial Baseado em LLM para Jogos de RPG Narrativos Textuais**

Recife

2026

Samuel Vidal dos Santos

# **AIDungeonMaster: Projeto e Implementação de um Mestre de Jogo Artificial Baseado em LLM para Jogos de RPG Narrativos Textuais**

Monografia apresentada ao Curso de Bacharelado em Ciências da Computação da Universidade Federal Rural de Pernambuco, como requisito parcial para obtenção do título de Bacharel em Ciências da Computação.

Universidade Federal Rural de Pernambuco – UFRPE

Departamento de Computação

Curso de Bacharelado em Ciências da Computação

Orientador: Lucas Silva Figueiredo

Recife

2026

Dados Internacionais de Catalogação na Publicação  
Sistema Integrado de Bibliotecas da UFRPE  
Bibliotecário(a): Suely Manzi – CRB-4 809

S237a Santos, Samuel Vidal dos.

AIDungeonMaster: projeto e implementação de um mestre de jogo artificial baseado em LLM para jogos de RPG narrativos textuais / Samuel Vidal dos Santos. - Recife, 2026.

74 f.; il.

Orientador(a): Lucas Silva Figueiredo.

Trabalho de Conclusão de Curso (Graduação) – Universidade Federal Rural de Pernambuco, Bacharelado em Ciência da Computação, Recife, BR-PE, 2026.

Inclui referências.

1. Inteligência artificial. 2. RPG (Linguagem de programação de computador). 3. Coesão (Linguística). 4. Sistemas de computação interativos 5. Computação - Estudo e ensino. I. Figueiredo, Lucas Silva, orient. II. Título

CDD 004

Samuel Vidal dos Santos

# **AIDungeonMaster: Projeto e Implementação de um Mestre de Jogo Artificial Baseado em LLM para Jogos de RPG Narrativos Textuais**

Monografia apresentada ao Curso de Bacharelado em Ciências da Computação da Universidade Federal Rural de Pernambuco, como requisito parcial para obtenção do título de Bacharel em Ciências da Computação.

Aprovado em: 12/02/2026

## **BANCA EXAMINADORA**

---

Prof. Dr. Lucas Silva Figueiredo (Orientador)  
Universidade Federal Rural de Pernambuco

---

Prof. Dra. Jeisa Pereira De Oliveira Domingues (Examinadora Interna)  
Universidade Federal Rural de Pernambuco

# Resumo

O uso de modelos de linguagem de grande porte (LLMs) em jogos de RPG narrativos tem se tornado cada vez mais comum, permitindo experiências interativas baseadas em linguagem natural. No entanto, essas abordagens frequentemente apresentam problemas relacionados à consistência do estado do mundo, coerência narrativa e previsibilidade das consequências das ações do jogador, especialmente em interações prolongadas. A ausência de mecanismos estruturados de validação, mediação e persistência do estado narrativo compromete a credibilidade, imersão e a continuidade dessas experiências.

Este trabalho apresenta o AI Dungeon Master, um sistema projetado como uma alternativa arquitetural para jogos de RPG narrativos assistidos por LLMs, com foco na separação explícita entre interpretação de intenções, validação mecânica, execução de ações e geração narrativa. A proposta busca mitigar inconsistências comuns ao delegar à LLM apenas responsabilidades narrativas e interpretativas, enquanto regras, estados e consequências são controlados por camadas determinísticas do sistema.

O AI Dungeon Master utiliza um fluxo baseado em intenções, agentes especializados e representação estruturada do estado do mundo, permitindo maior controle sobre ações permitidas, consequências diretas, além de facilitar a manutenção da coerência narrativa ao longo da campanha. Como resultado, o sistema demonstra uma abordagem mais robusta para integração de LLMs em RPGs narrativos, preservando a criatividade da geração de texto sem comprometer a integridade do mundo de jogo.

**Palavras-chave:** Inteligência Artificial Generativa; RPG narrativo; Coerência Narrativa; Sistemas Interativos.

# Abstract

The use of large language models (LLMs) in narrative role-playing games (RPGs) has become increasingly common, enabling interactive experiences based on natural language. However, such approaches often suffer from issues related to world state consistency, narrative coherence, and the predictability of player action consequences, especially during prolonged interactions. The lack of structured mechanisms for validation, mediation, and persistence of the narrative state undermines the credibility, immersion, and continuity of these experiences.

This work presents AI Dungeon Master, a system designed as an architectural alternative for LLM-assisted narrative RPGs, focusing on the explicit separation between intent interpretation, mechanical validation, action execution, and narrative generation. The proposal aims to mitigate common inconsistencies by assigning LLMs only narrative and interpretative responsibilities, while rules, states, and consequences are managed by deterministic system layers.

AI Dungeon Master employs an intent-based workflow, specialized agents, and a structured representation of the world state, allowing greater control over permitted actions and both direct and indirect consequences, while facilitating the maintenance of narrative coherence throughout a campaign. As a result, the system demonstrates a more robust approach to integrating LLMs into narrative RPGs, preserving the creative potential of text generation without compromising the integrity of the game world.

**Keywords:** Generative Artificial Intelligence; Narrative RPG; Narrative Coherence; Interactive Systems.

# Lista de ilustrações

Figura 1 – Fluxograma input DO . . . . .	30
Figura 2 – Fluxograma input SAY . . . . .	31
Figura 3 – Fluxograma input SEE . . . . .	32
Figura 4 – Grafo de Entidades . . . . .	34
Figura 5 – Tela Dashboard de Campanhas . . . . .	50
Figura 6 – Tela Criação de Campanha . . . . .	50
Figura 7 – Tela Campanha . . . . .	51
Figura 8 – Interação SEE . . . . .	53
Figura 9 – Interação SAY . . . . .	54
Figura 10 – Interação MOVE . . . . .	56
Figura 11 – Interação ATTACK . . . . .	57
Figura 12 – Uso da Poção de Cura . . . . .	58
Figura 13 – Interação USEITEM . . . . .	59
Figura 14 – Interação NARRATIVE . . . . .	61

# Lista de tabelas

Tabela 1 – Detalhamento das Localizações e Ambiências da Campanha. . . . .	36
Tabela 2 – Atributos e Papéis dos Personagens (Agentes). . . . .	36
Tabela 3 – Propriedades dos Itens e Equipamentos. . . . .	36
Tabela 4 – Modelos de Linguagem Utilizados no AI Dungeon Master . . . . .	63
Tabela 5 – Análise comparativa de performance e viabilidade entre os modelos. . . . .	63

# Lista de abreviaturas e siglas

LLM	Large Language Model
RPG	Role-Playing Game
IA	Inteligência Artificial
NPC	Non-Player Character
DM	Dungeon Master
NLI	Natural Language Inference
RAG	Retrieval-Augmented Generation

# Sumário

<b>Lista de ilustrações</b>	<b>4</b>
<b>1 INTRODUÇÃO</b>	<b>9</b>
1.1 <b>Problema da Pesquisa</b>	<b>11</b>
1.2 <b>Objetivos</b>	<b>12</b>
1.2.1 Objetivos Gerais	12
1.2.2 Objetivos Específicos	12
1.3 <b>Contribuições</b>	<b>13</b>
1.4 <b>Disponibilidade de Código e Dados</b>	<b>14</b>
<b>2 TRABALHOS RELACIONADOS</b>	<b>15</b>
2.1 <b>AI Dungeon</b>	<b>15</b>
2.2 <b>RPGBench</b>	<b>16</b>
2.3 <b>CharacterBox</b>	<b>17</b>
2.4 <b>Codifying Character Logic in Role-Playing</b>	<b>18</b>
<b>3 FUNDAMENTAÇÃO TEÓRICA</b>	<b>20</b>
3.1 <b>Modelos de Linguagem de Grande Porte (LLMs)</b>	<b>20</b>
3.1.1 Definição e Funcionamento Geral	20
3.1.2 Capacidades em Geração Narrativa	20
3.1.3 Limitações Intrínsecas dos LLMs	21
3.1.4 Implicações para RPGs Narrativos Assistidos por LLMs	21
3.2 <b>Geração Aumentada de Recuperação (RAG)</b>	<b>22</b>
3.3 <b>Narrativas Interativas e RPGs Textuais</b>	<b>23</b>
3.3.1 Conceito de RPG Narrativo	23
3.3.2 O Papel do Narrador (Dungeon Master)	23
3.3.3 Elementos Fundamentais de RPGs Textuais	24
3.3.4 Narrativa Linear versus Narrativa Emergente	25
3.3.5 Implicações para Sistemas Narrativos Assistidos por LLMs	25
3.4 <b>Representação e Persistência de Estado em Sistemas Interativos</b>	<b>26</b>
3.4.1 Conceito de estado em sistemas interativos	26
3.4.2 Persistência de mundo em jogos	26
3.4.3 Problemas de estado implícito em modelos de linguagem	27
3.4.4 Representações estruturadas versus memória textual	27
3.4.5 Trade-offs entre memória simbólica e narrativa	28

<b>4</b>	<b>DESENVOLVIMENTO</b>	<b>29</b>
<b>4.1</b>	<b>Visão geral do sistema</b>	<b>29</b>
<b>4.2</b>	<b>Persistência de Dados</b>	<b>32</b>
4.2.1	Campanha <i>Toy</i>	35
4.2.1.1	Estado Global e Ambiente (WorldState)	36
<b>4.3</b>	<b>Processamento da Entrada do Jogador</b>	<b>36</b>
4.3.1	Player input	36
4.3.2	Context Builder (Estado Básico do Jogo)	38
4.3.3	ActionResolverAgent	39
4.3.4	Intent Validator	41
4.3.5	SkillCheck Validator	42
<b>4.4</b>	<b>Construção de Contexto Específico por Intent</b>	<b>43</b>
<b>4.5</b>	<b>Geração Narrativa</b>	<b>45</b>
<b>4.6</b>	<b>Reação do Mundo e Consequências</b>	<b>46</b>
4.6.1	DirectMechanicalReactor	47
<b>5</b>	<b>RESULTADO E DISCUSSÃO</b>	<b>49</b>
<b>5.1</b>	<b>Ambiente de Execução e Fluxo Geral do Sistema</b>	<b>49</b>
<b>5.2</b>	<b>Demonstração dos Intents e suas Reações</b>	<b>51</b>
5.2.1	Intent SEE — Observação do Ambiente	52
5.2.2	Intent SAY — Interação Social	53
5.2.3	Intent MOVE — Alteração Espacial	55
5.2.4	Intent ATTACK — Conflito e Consequências Mecânicas	56
5.2.5	Intent USEITEM — Manipulação de Inventário e Estado	58
5.2.6	Intent NARRATIVE — Ação Livre e Flexível	60
<b>5.3</b>	<b>Modelos de Linguagem Utilizados e seus Papéis no Sistema</b>	<b>62</b>
<b>5.4</b>	<b>Discussão</b>	<b>63</b>
5.4.1	Aspectos que funcionaram adequadamente	63
5.4.2	Limitações e aspectos que não funcionaram plenamente	64
5.4.3	Pitfalls e Lições Aprendidas	65
5.4.4	Síntese da discussão	66
<b>6</b>	<b>TRABALHOS FUTUROS</b>	<b>67</b>
<b>7</b>	<b>CONCLUSÃO</b>	<b>69</b>
	<b>REFERÊNCIAS</b>	<b>71</b>

# 1 Introdução

O *Role-Playing Game* (RPG) caracteriza-se como um sistema de jogo em que os participantes assumem papéis de personagens dentro de um mundo fictício, cujas interações são governadas por regras estruturadas e estados de jogo em constante evolução. Nesse contexto, os *Large Language Models* (LLMs) têm ampliado significativamente as possibilidades de criação de narrativas interativas, consolidando-se como ferramentas promissoras para o desenvolvimento de RPGs baseados em texto (CHEN et al., 2024; YU et al., 2025). Modelos como GPT-4, DeepSeek e Gemini demonstram elevada capacidade de geração de linguagem natural, combinando criatividade narrativa com manutenção de contexto em larga escala, o que possibilita a construção dinâmica de mundos virtuais, personagens complexos e enredos não lineares (PARK et al., 2023; ACHIAM et al., 2023; TEAM et al., 2023). Essas características representam uma alternativa relevante aos sistemas tradicionais baseados em árvores de decisão e scripts pré-programados, oferecendo experiências mais flexíveis, adaptativas e responsivas às ações do jogador (LEANDRO et al., 2024).

Apesar desse potencial, a utilização direta de LLMs como motores centrais de RPGs narrativos apresenta limitações importantes que impactam a qualidade, a coerência e a confiabilidade da experiência interativa. Um dos principais desafios é a manutenção da consistência narrativa e do estado do mundo ao longo de interações prolongadas (YU et al., 2025). Em cenários contínuos, os modelos tendem a apresentar o fenômeno conhecido como *narrative drift*, no qual informações previamente estabelecidas são esquecidas, contraditas ou reinterpretadas de forma inconsistente, comprometendo a continuidade lógica do mundo de jogo. Esse problema se torna ainda mais evidente em campanhas longas, onde decisões passadas deveriam influenciar eventos futuros de forma previsível e coerente.

Além disso, LLMs demonstram dificuldades em aderir rigorosamente a sistemas de regras formais, fundamentais para mecânicas de RPG como movimentação, combate, uso de itens e progressão de personagens (YU et al., 2025; MU et al., 2023). Quando tais regras são tratadas implicitamente pelo modelo, surgem situações em que ações fisicamente impossíveis são permitidas, consequências são ignoradas ou resultados narrativos contradizem o estado atual do mundo. Problemas adicionais incluem a representação consistente de personagens com traços psicológicos estáveis em mundos dinâmicos (WANG et al., 2025; PENG; SHANG, 2025), bem como a imprevisibilidade das consequências das ações do jogador quando toda a mediação do sistema é delegada exclusivamente à geração textual do modelo.

Diante desse cenário, este trabalho apresenta o AI Dungeon Master, um sistema desenvolvido como uma alternativa arquitetural para jogos de RPG narrativos assistidos por LLMs. A proposta central do sistema é a separação explícita de responsabilidades entre as camadas que interpretam a linguagem natural do jogador, validam mecanicamente as ações, executam alterações no estado do mundo e geram a narrativa apresentada ao usuário. Essa separação busca mitigar limitações observadas em abordagens monolíticas, nas quais o modelo de linguagem atua simultaneamente como intérprete, árbitro de regras e narrador.

Nesse modelo, as LLMs são utilizadas principalmente para tarefas narrativas, interpretativas e geração de conteúdo dinâmico, como a compreensão da intenção do jogador e a geração de descrições textuais e outros conteúdos narrativos, enquanto decisões críticas relacionadas a regras, estados e consequências são controladas por componentes determinísticos do sistema. Essa organização permite maior previsibilidade no comportamento do mundo de jogo, sem restringir a expressividade e a criatividade da narrativa produzida.

O AI Dungeon Master adota um fluxo baseado em *intents*, agentes especializados e uma representação estruturada do estado do mundo, permitindo identificar com clareza as ações pretendidas pelo jogador, verificar sua validade, aplicar consequências diretas e indiretas de forma controlada. Esse fluxo facilita não apenas a validação de ações, mas também a manutenção da memória do mundo, das interações passadas e das condições atuais dos personagens e ambientes, elementos essenciais para campanhas narrativas consistentes.

A proposta dialoga com trabalhos recentes da literatura, como o RPGBench (YU et al., 2025), ao reconhecer a necessidade de estruturas explícitas para avaliação e controle da coerência narrativa, e com abordagens como o CharacterBox (WANG et al., 2025), ao considerar a importância da consistência comportamental de personagens. No entanto, o AI Dungeon Master avança ao integrar esses princípios em um fluxo arquitetural unificado, voltado não apenas à avaliação, mas à execução prática de campanhas narrativas interativas. Ao estruturar explicitamente a mediação entre linguagem natural, regras e estado do mundo, o sistema busca reduzir a ocorrência de inconsistências narrativas e lógicas observadas em soluções baseadas exclusivamente em LLMs.

Assim, este trabalho contribui ao apresentar uma abordagem arquitetural alternativa para RPGs narrativos assistidos por LLMs, demonstrando que a combinação entre modelos generativos e camadas determinísticas de mediação pode preservar a criatividade narrativa sem comprometer a integridade, a coerência e a previsibilidade do estado do mundo de jogo ao longo de interações prolongadas.

## 1.1 Problema da Pesquisa

Apesar do potencial demonstrado pelos *Large Language Models* (LLMs) na geração de narrativas interativas para *text-based RPGs*, sua utilização direta como mediadores centrais da experiência de jogo apresenta limitações estruturais que comprometem a coerência, a previsibilidade e a continuidade das campanhas narrativas. Em interações prolongadas, observa-se com frequência a dificuldade dos modelos em manter a consistência do estado do mundo e da narrativa, manifestada pelo fenômeno conhecido como *narrative drift* (YU et al., 2025), o que impacta negativamente a imersão do jogador.

Além disso, LLMs apresentam limitações significativas quanto à aderência rigorosa a sistemas de regras formais, essenciais para a implementação de mecânicas típicas de RPG, como movimentação, combate, uso de itens e progressão de personagens (YU et al., 2025; MU et al., 2023). Quando tais regras são tratadas implicitamente pelo modelo, surgem inconsistências lógicas, permissões indevidas de ações e consequências narrativas incompatíveis com o estado atual do mundo. Soma-se a isso a dificuldade em manter a representação consistente de personagens não jogáveis (NPCs) em ambientes dinâmicos, especialmente no que diz respeito a traços comportamentais, memórias e reações previsíveis ao longo do tempo (WANG et al., 2025; PENG; SHANG, 2025).

Essas limitações evidenciam a ausência de mecanismos estruturados de validação, mediação e persistência do estado narrativo nas abordagens atuais, nas quais a LLM é frequentemente responsável, de forma monolítica, por interpretar intenções, aplicar regras e gerar a narrativa. Tal sobrecarga compromete a confiabilidade do sistema e dificulta a construção de experiências narrativas coerentes e mecanicamente consistentes em longo prazo.

Diante desse contexto, o problema de pesquisa deste trabalho pode ser formulado da seguinte maneira: Como projetar uma arquitetura para jogos de RPG narrativos assistidos por LLMs que permita preservar a criatividade da geração de linguagem natural, garantindo simultaneamente a consistência do estado do mundo, a previsibilidade das consequências das ações do jogador e a aderência a regras de jogo bem definidas?

A solução investigada neste trabalho parte da hipótese de que a separação explícita de responsabilidades entre interpretação de intenções, validação mecânica, execução de ações e geração narrativa pode mitigar os problemas observados nas abordagens atuais. Nesse sentido, o AI Dungeon Master é proposto como uma alternativa arquitetural que integra princípios discutidos em trabalhos como o RPGBench (YU et al., 2025) e o CharacterBox (WANG et al., 2025), mas avança ao aplicá-los de

forma unificada em um fluxo operacional voltado à execução prática de campanhas narrativas, e não apenas à avaliação de modelos. Assim, o sistema busca demonstrar que LLMs podem atuar de maneira eficaz como componentes narrativos e interpretativos dentro de um ecossistema controlado, sem assumir o papel de árbitros exclusivos do estado e das regras do mundo de jogo.

## 1.2 Objetivos

### 1.2.1 Objetivos Gerais

Desenvolver e avaliar o AI Dungeon Master, um sistema arquitetural para jogos de RPG narrativos assistidos por *Large Language Models* (LLMs), projetado para preservar a criatividade narrativa da geração de linguagem natural enquanto garante consistência do estado do mundo, aderência a regras de jogo e previsibilidade das consequências das ações do jogador por meio de camadas determinísticas de mediação.

### 1.2.2 Objetivos Específicos

- a) Projetar uma arquitetura modular baseada na separação explícita de responsabilidades, distinguindo interpretação de linguagem natural, validação mecânica de ações, execução de mudanças no estado do mundo e geração narrativa.
- b) Implementar um fluxo de interação baseado em tipos de intenções (*intents*) e agentes especializados, capaz de identificar e classificar as ações do jogador de forma estruturada e previsível.
- c) Utilizar LLMs de forma controlada e contextualizada, restringindo sua atuação às tarefas narrativas e interpretativas, sem delegar a elas o controle direto sobre regras, estados ou consequências do mundo de jogo.
- d) Desenvolver uma representação estruturada e persistente do estado do mundo, incluindo personagens, locais, itens e histórico de interações, de modo a reduzir inconsistências narrativas em campanhas prolongadas.
- e) Integrar mecanismos de validação mecânica, como verificação de permissões de ações e resolução de consequências imediatas, assegurando a coerência lógica das interações típicas de RPG (movimento, diálogo, combate e uso de itens).
- f) Incorporar princípios discutidos em trabalhos relacionados, como o RPGBench (YU et al., 2025) e o CharacterBox (WANG et al., 2025), adaptando-os para um fluxo operacional voltado à execução prática de campanhas narrativas, e não apenas à avaliação de modelos.

- g) Avaliar qualitativamente a abordagem proposta, discutindo como a arquitetura do AI Dungeon Master contribui para mitigar problemas de *narrative drift*, inconsistência mecânica e imprevisibilidade observados em sistemas baseados exclusivamente em LLMs.

### 1.3 Contribuições

Este trabalho contribui para a área de jogos narrativos assistidos por inteligência artificial ao propor, implementar e analisar uma abordagem arquitetural alternativa para a integração de *Large Language Models* (LLMs) em sistemas de RPG narrativos. Diferentemente de abordagens monolíticas, nas quais o modelo de linguagem atua como mediador exclusivo da interpretação, validação e geração narrativa, o AI Dungeon Master introduz uma separação explícita de responsabilidades, promovendo maior controle, previsibilidade e coerência ao longo de campanhas narrativas prolongadas.

Como principal contribuição, o trabalho apresenta uma arquitetura modular baseada em *intents* e agentes especializados, na qual a LLM é empregada de forma controlada, restrita a tarefas narrativas e interpretativas, enquanto regras de jogo, validações mecânicas e persistência do estado do mundo são gerenciadas por camadas determinísticas. Essa organização demonstra, na prática, como é possível preservar a criatividade e a fluidez da geração de linguagem natural sem comprometer a integridade lógica e temporal do mundo de jogo.

Adicionalmente, o trabalho contribui ao propor uma representação estruturada e persistente do estado do mundo para RPGs narrativos assistidos por LLMs, contemplando personagens, locais, itens e histórico de interações. Essa abordagem evidencia as limitações do uso exclusivo de memória textual implícita em modelos generativos e reforça a importância de mecanismos explícitos de estado para a manutenção da coerência narrativa e mecânica em interações prolongadas.

No contexto da literatura existente, este trabalho avança além de propostas voltadas à avaliação de modelos, como o RPGBench (YU et al., 2025), e de frameworks focados na consistência comportamental de personagens, como o CharacterBox (WANG et al., 2025), ao integrar esses princípios em um fluxo operacional completo, orientado à execução contínua de campanhas narrativas. Assim, o AI Dungeon Master contribui não apenas como um estudo conceitual, mas como uma demonstração prática de uma arquitetura viável para RPGs narrativos mediados por LLMs.

## 1.4 Disponibilidade de Código e Dados

Com o objetivo de promover a reprodutibilidade desta pesquisa e contribuir com a comunidade de desenvolvedores de jogos baseados em Inteligência Artificial, todo o material técnico produzido foi disponibilizado em um repositório público. O código-fonte do *AI Dungeon Master*, os *prompts* utilizados pelos agentes, bem como os *scripts* de definição da campanha *toy* (`campaign_001`), podem ser acessados no seguinte endereço:

[<https://github.com/Samuelvidal99/ai-dungeon-master>](https://github.com/Samuelvidal99/ai-dungeon-master)

O repositório inclui instruções detalhadas para a configuração do ambiente, incluindo as dependências de *frontend*, *backend* e os modelos locais utilizados via *Ollama*.

## 2 Trabalhos relacionados

### 2.1 AI Dungeon

O *AI Dungeon* (Latitude Inc., 2019) é uma plataforma de ficção interativa baseada em inteligência artificial, lançada inicialmente em 2019, que permite a criação de narrativas dinâmicas em formato de RPG textual tanto em modo *single-player* quanto *multiplayer*. A proposta central da plataforma é utilizar *Large Language Models* (LLMs) como mecanismo principal para geração contínua de narrativas, possibilitando experiências abertas, não lineares e potencialmente ilimitadas, nas quais o jogador pode interagir livremente por meio de linguagem natural.

O objetivo principal do *AI Dungeon* é oferecer uma experiência de narrativa interativa altamente flexível, na qual a criatividade do jogador não é limitada por roteiros pré-definidos ou estruturas narrativas rígidas. Para isso, o sistema delega à LLM a responsabilidade quase total pela interpretação das ações do jogador, pela evolução do estado narrativo e pela geração das consequências textuais dessas ações, atuando efetivamente como um *Dungeon Master* automatizado.

Do ponto de vista metodológico, o *AI Dungeon* adota uma abordagem predominantemente *end-to-end*, na qual a LLM recebe como entrada o histórico textual da narrativa, incluindo ações anteriores, descrições e eventuais anotações de contexto fornecidas manualmente pelo jogador, e produz como saída a continuação da história. Embora a plataforma ofereça mecanismos auxiliares, como campos de memória ou contexto fixo, esses elementos são tratados como extensões textuais do *prompt*, não havendo uma separação explícita entre estado do mundo, regras de jogo e geração narrativa. Dessa forma, a coerência do universo fictício e a aplicação consistente de regras dependem exclusivamente da capacidade do modelo em inferir e manter tais informações implicitamente ao longo da interação com o jogador.

Como resultado, o *AI Dungeon* demonstrou o potencial prático dos LLMs para a geração de narrativas interativas ricas e envolventes, a plataforma evidenciou que modelos de linguagem são capazes de improvisar diálogos, descrições e eventos de forma fluida, criando a sensação de um narrador responsivo e criativo.

Entretanto, diversos problemas emergem em interações prolongadas, especialmente no que se refere à manutenção do estado do mundo e à consistência narrativa. A ausência de uma representação estruturada e persistente de entidades, eventos e regras leva frequentemente a contradições internas, esquecimento de elementos centrais da história e alterações arbitrárias no comportamento de personagens não jo-

gáveis. Embora o sistema permita que o jogador insira manualmente informações de contexto para tentar mitigar esses problemas, tal abordagem transfere ao usuário a responsabilidade de manter a coerência narrativa, o que se mostra trabalhoso e pouco eficaz em campanhas mais longas ou complexas.

Essa limitação evidencia uma lacuna importante na arquitetura do *AI Dungeon*, que é a dependência quase exclusiva da LLM como mediadora central de interpretação, execução e narração. Ao não separar explicitamente responsabilidades como validação de ações, persistência do estado do mundo e aplicação de regras formais, o sistema compromete a previsibilidade e a confiabilidade das interações. É justamente nesse ponto que se insere a proposta do *AI Dungeon Master*, que busca preservar a criatividade narrativa proporcionada pelos LLMs, ao mesmo tempo em que introduz camadas determinísticas de controle, validação e persistência, reduzindo problemas de *narrative drift* e inconsistência mecânica observados em plataformas como o *AI Dungeon*.

## 2.2 RPGBench

O *RPGBench* (YU et al., 2025) é um *benchmark* proposto para avaliar grandes modelos de linguagem (LLMs) no papel de motores de jogos de RPG baseados em texto, com foco na análise de criatividade, coerência narrativa e consistência lógica em interações sequenciais. Diferentemente de sistemas voltados à execução contínua de campanhas ou à mediação completa da experiência de jogo, o *RPGBench* tem como objetivo principal oferecer um instrumento padronizado de avaliação comparativa do desempenho de LLMs nesse domínio.

O *benchmark* é estruturado em duas tarefas centrais: *Game Creation* (GC) e *Game Simulation* (GS). Na tarefa de criação, o modelo deve gerar um mundo de RPG válido a partir de uma representação estruturada de estados e eventos, respeitando regras e condições de término. Já na tarefa de simulação, o modelo conduz interações narrativas em múltiplas rodadas, atualizando estados internos e aplicando regras do jogo. Para avaliação, o *RPGBench* combina métricas objetivas, como verificação automática de transições de estado e consistência de variáveis, com métricas subjetivas, incluindo qualidade narrativa, interesse do conteúdo e capacidade de interpretação de papéis, estas últimas avaliadas por meio de um modelo de linguagem atuando como juiz.

Os resultados apresentados indicam que modelos de estado da arte são capazes de produzir narrativas envolventes e contextualmente ricas, porém frequentemente apresentam dificuldades em manter a consistência de regras, estados e variáveis ao longo de cenários mais extensos ou complexos, indicam também que muitos modelos

possuem grande dificuldade durante a etapa GC, gerando, em sua maioria, estruturas de mundo incorretas e incompletas. Isso evidencia limitações importantes das LLMs quando utilizadas como motores únicos de jogos, especialmente em contextos que exigem persistência e controle rigoroso do estado do mundo.

Apesar de sua relevância para a avaliação de modelos, o *RPGBenchmark* não se propõe a funcionar como uma *engine* de RPG propriamente dita. Os cenários gerados são episódicos, geralmente baseados em personagens ou universos já existentes na mídia, e a interação do jogador é limitada a um conjunto fixo de opções pré-definidas por turno. Além disso, toda a persistência do mundo e do contexto narrativo é centralizada em uma estrutura estática, sem atualização contínua ao longo da “campanha”, o que restringe a evolução dinâmica da narrativa e do estado do jogo.

Dessa forma, observa-se uma lacuna entre a avaliação da capacidade das LLMs em ambientes controlados e a construção de sistemas que atuem efetivamente como mestres de jogo, capazes de gerenciar mundos persistentes, manter coerência narrativa de longo prazo e integrar mecanismos explícitos de controle de estado. Essa lacuna motiva propostas que buscam combinar LLMs com arquiteturas externas de memória, representação de mundo e regras, como é o caso do sistema desenvolvido neste trabalho.

## 2.3 CharacterBox

O *CharacterBox* (WANG et al., 2025) é um *framework* proposto com o objetivo de avaliar a capacidade de modelos de linguagem de grande porte em interpretar e manter comportamentos coerentes de personagens inseridos em ambientes narrativos dinâmicos. Diferentemente de abordagens focadas exclusivamente na geração textual, o trabalho enfatiza a consistência comportamental e psicológica dos personagens ao longo de interações sucessivas.

O método adotado pelo *CharacterBox* baseia-se em uma arquitetura composta por dois agentes principais: o *Character Agent* e o *Narrator Agent*. O *Character Agent* é fundamentado em conceitos oriundos da ciência comportamental e da psicologia, sendo responsável por tomar decisões e agir de acordo com traços de personalidade e estados internos previamente definidos. Já o *Narrator Agent* atua como mediador narrativo, contextualizando as ações do personagem e descrevendo suas consequências no ambiente. Além disso, o *framework* registra trajetórias comportamentais detalhadas, que podem ser reutilizadas tanto para avaliação quanto para o aprimoramento de modelos de linguagem, incluindo modelos *open source* e de menor porte.

Como resultado, o *CharacterBox* demonstra que a decomposição do processo narrativo em agentes especializados contribui para uma avaliação mais precisa da

capacidade dos LLMs em interpretar personagens de forma consistente. O trabalho também propõe os modelos *CharacterNR* e *CharacterRM*, desenvolvidos para substituir chamadas a APIs proprietárias, como o GPT-4, reduzindo custos e ampliando a acessibilidade do *framework* para a comunidade científica.

Apesar de seus avanços, o *CharacterBox* apresenta limitações relevantes no contexto de motores de RPG textuais completos. O *framework* tem como foco principal a avaliação e o refinamento da interpretação de personagens, não contemplando mecanismos robustos de persistência de estado do mundo, evolução contínua da narrativa ou gerenciamento explícito de regras e eventos ao longo de campanhas extensas. Nesse sentido, observa-se uma lacuna na integração entre consistência comportamental de personagens e a manutenção estruturada de estados narrativos e lógicos do mundo do jogo. Essa lacuna motiva a proposta do AI Dungeon Master, que busca unificar controle de estado, coerência narrativa e persistência de contexto em uma engine de RPG textual, indo além do escopo de comportamento de personagens apresentado pelo *CharacterBox*.

## 2.4 Codifying Character Logic in Role-Playing

O trabalho *Codifying Character Logic in Role-Playing* (PENG; SHANG, 2025) propõe uma abordagem alternativa para a interpretação de personagens em sistemas de role-playing baseados em LLMs, introduzindo o conceito de *Codified Profiles*. Diferentemente das abordagens tradicionais baseadas exclusivamente em *prompts* textuais, os autores representam a lógica comportamental dos personagens por meio de funções estruturadas e executáveis, compostas por regras explícitas de controle, como estruturas condicionais (if-then-else) e verificações semânticas sobre o estado da cena.

O objetivo principal do trabalho é aumentar a consistência, persistência e controlabilidade do comportamento de personagens interpretados por modelos de linguagem. Para isso, cada perfil codificado define uma função `parse_by_scene(scene)`, responsável por avaliar a cena atual e produzir um conjunto de afirmações comportamentais fundamentadas em lógica explícita. Essas decisões são guiadas por perguntas semanticamente relevantes, por exemplo, “o personagem está em perigo?”, que são avaliadas pelo próprio LLM como verdadeiras, falsas ou desconhecidas, servindo como base para a ativação das regras comportamentais.

Como método de validação, os autores propõem um *benchmark* composto por 83 personagens e 5.141 cenas extraídas da plataforma Fandom, utilizando métricas baseadas em *Natural Language Inference* (NLI) para comparar as ações geradas pelos modelos com ações consideradas corretas. Os resultados experimentais indicam que

os *Codified Profiles* superam abordagens puramente baseadas em *prompt* em três aspectos principais: (i) persistência comportamental, ao garantir que a lógica do personagem seja sempre executada; (ii) facilidade de atualização, permitindo inspeção e modificação explícita das regras; e (iii) controle da aleatoriedade, viabilizando variações estocásticas de comportamento de forma mais previsível. Um resultado relevante é que, ao deslocar parte significativa do raciocínio para uma etapa de pré-processamento lógico, até mesmo modelos com cerca de 1 bilhão de parâmetros alcançam desempenho competitivo em tarefas de *role-playing*.

Apesar das contribuições, o trabalho apresenta limitações importantes no que diz respeito à expressividade e evolução dos personagens ao longo do tempo. A lógica comportamental é fortemente baseada em regras explícitas e condicionais estáticas, como “se o personagem estiver em perigo, então deve fugir”, o que reduz a complexidade das decisões possíveis e aproxima o modelo de um sistema determinístico de regras. Além disso, o artigo não especifica mecanismos claros para reavaliação, adaptação ou enriquecimento dinâmico dessas regras à medida que a narrativa evolui ou que o personagem se desenvolve, limitando a capacidade de representar mudanças psicológicas, aprendizado narrativo ou contradições internas, elementos centrais em narrativas longas e abertas. Essa lacuna evidencia a necessidade de abordagens híbridas que conciliem controle estrutural, memória persistente e flexibilidade narrativa, especialmente em cenários de *role-playing* mais amplos que vão além da reação de um personagem específico para determinada situação.

## 3 Fundamentação teórica

### 3.1 Modelos de Linguagem de Grande Porte (LLMs)

Os *Large Language Models* (LLMs) constituem uma classe de modelos de aprendizado de máquina voltados à compreensão e geração de linguagem natural, treinados a partir de grandes volumes de dados textuais (BROWN et al., 2020). Nos últimos anos, esses modelos passaram a desempenhar um papel central em aplicações interativas baseadas em linguagem, incluindo assistentes conversacionais, sistemas de recomendação e jogos narrativos baseados em texto (BOMMASANI et al., 2022).

#### 3.1.1 Definição e Funcionamento Geral

Os LLMs são majoritariamente fundamentados em arquiteturas do tipo *Transformer* (VASWANI et al., 2017), caracterizadas pelo uso de mecanismos de atenção para modelar dependências entre *tokens* ao longo de sequências textuais. O treinamento desses modelos ocorre, em geral, de forma auto-supervisionada, por meio da tarefa de predição do próximo *token* em uma sequência (BROWN et al., 2020), o que possibilita a aprendizagem de padrões sintáticos, semânticos e pragmáticos da linguagem natural a partir de grandes corpora textuais.

Durante o processo de inferência, um LLM recebe um contexto textual e gera respostas de maneira autoregressiva, isto é, cada *token* produzido depende do contexto fornecido e dos *tokens* previamente gerados. Esse mecanismo permite a produção de textos extensos com coerência local, viabilizando a simulação de diálogos, descrições narrativas e interações complexas em linguagem natural.

#### 3.1.2 Capacidades em Geração Narrativa

No contexto de jogos de RPG narrativos, os LLMs demonstram elevada capacidade de geração criativa, sendo capazes de descrever ambientes, interpretar falas de personagens, improvisar diálogos e responder a ações do jogador de forma fluida e contextualizada (FAN; LEWIS; DAUPHIN, 2018). Modelos como GPT-4, DeepSeek e Gemini destacam-se pela habilidade de manter coerência textual em janelas de contexto extensas (ACHIAM et al., 2023; TEAM et al., 2024), além de adaptar o tom e o estilo narrativo conforme o cenário apresentado.

Essas características configuram uma alternativa relevante aos sistemas tradicionais de narrativa baseados em árvores de decisão e scripts pré-programados, os

quais tendem a limitar a expressividade e a flexibilidade das interações (RIEDL; BULITKO, 2013). Em contraste, abordagens fundamentadas em LLMs possibilitam narrativas emergentes e menos determinísticas (YANNAKAKIS; TOGELIUS, 2018), favorecendo experiências mais dinâmicas e responsivas às ações do jogador.

### 3.1.3 Limitações Intrínsecas dos LLMs

Apesar de suas capacidades expressivas, os LLMs apresentam limitações estruturais que impactam diretamente sua utilização como motores centrais de sistemas interativos complexos, como RPGs narrativos persistentes.

Uma limitação significativa refere-se à ausência de um modelo explícito e persistente de estado do mundo. Os LLMs não mantêm memória estruturada entre interações, dependendo exclusivamente do contexto textual fornecido a cada requisição (PARK et al., 2023). Em interações prolongadas, essa característica pode resultar no fenômeno conhecido como *narrative drift*, no qual informações previamente estabelecidas são esquecidas, contraditas ou reinterpretadas de maneira inconsistente (ZHAO et al., 2023).

Outra limitação relevante diz respeito à dificuldade de aderência rigorosa a sistemas de regras formais. Embora os LLMs sejam capazes de reproduzir regras descritas em linguagem natural, não há garantias de que essas regras sejam aplicadas de forma consistente e previsível (MU et al., 2023), especialmente em cenários que envolvem mecânicas complexas, como combate, restrições físicas do mundo ou validação de ações do jogador. Tal comportamento compromete a confiabilidade de sistemas que delegam integralmente à LLM a responsabilidade de decidir consequências e validar mecânicas complexas (YANNAKAKIS; TOGELIUS, 2018).

Adicionalmente, esses modelos são suscetíveis à geração de respostas plausíveis, porém inconsistentes com o estado do mundo ou com informações previamente estabelecidas, fenômeno comumente descrito como *alucinação* (JI et al., 2023). Em jogos narrativos, esse comportamento pode resultar em contradições lógicas, criação indevida de entidades ou avanço não autorizado do tempo e da narrativa.

### 3.1.4 Implicações para RPGs Narrativos Assistidos por LLMs

As limitações apresentadas evidenciam que, embora os LLMs sejam ferramentas eficazes para geração narrativa e interpretação semântica de ações em linguagem natural (FAN; LEWIS; DAUPHIN, 2018), sua utilização como autoridade única sobre regras, estado do mundo e consequências das ações do jogador compromete a coerência e a integridade da experiência de jogo (JI et al., 2023; ZHAO et al., 2023).

Dessa forma, torna-se necessária a adoção de abordagens arquiteturais que delimitem e orientem o papel dos LLMs, atribuindo-lhes responsabilidades compatíveis com suas capacidades, como a geração de descrições narrativas e a interpretação da intenção do jogador. Aspectos críticos, como validação mecânica, persistência de estado e aplicação de consequências, devem ser tratados por camadas determinísticas e estruturadas do sistema (RIEDL; BULITKO, 2013).

Essa separação de responsabilidades fundamenta a proposta do AI Dungeon Master, que integra LLMs como agentes narrativos e interpretativos em um fluxo controlado por regras explícitas e por uma representação estruturada do estado do mundo (PARK et al., 2023), visando preservar a coerência narrativa sem comprometer a integridade do sistema (YANNAKAKIS; TOGELIUS, 2018).

## 3.2 Geração Aumentada de Recuperação (RAG)

A técnica de Geração Aumentada de Recuperação, conhecida pela sigla RAG (Retrieval-Augmented Generation), surge como uma solução para mitigar duas limitações críticas dos modelos de linguagem de grande porte: o conhecimento estático, limitado à data de corte do seu treinamento, e a propensão a alucinações em contextos de longa duração (LEWIS et al., 2020). O RAG permite que o modelo consulte uma base de dados externa de documentos ou fatos antes de gerar uma resposta, garantindo que o conteúdo produzido seja ancorado em informações verificáveis.

Em sistemas interativos e RPGs narrativos, o funcionamento do RAG pode ser compreendido como um processo de três etapas principais:

- a) **Indexação:** as informações do mundo (histórico de eventos, biografias de NPCs, descrições de locais) são convertidas em vetores numéricos (*embeddings*), ou em outras estruturas de dados, e armazenadas em um banco de dados vetorial, em grafos ou dos mais variados tipos.
- b) **Recuperação:** quando o jogador interage com o sistema, o sistema busca na base de dados os fragmentos, ou entidades, mais relevantes para o contexto atual.
- c) **Aumento e Geração:** as informações recuperadas são inseridas no contexto (prompt) enviado à LLM, que utiliza esses dados como "memória de trabalho" para gerar uma resposta coerente e factual (GAO et al., 2023).

A implementação do RAG transforma a LLM de um sistema puramente probabilístico em um sistema assistido por memória de longo prazo. Isso é particularmente

relevante para a persistência narrativa, pois permite que o *NarrativeAgent* recupere detalhes específicos do passado da campanha que já haviam saído da janela de contexto imediata, preservando a continuidade e a integridade do universo ficcional (BARNETT et al., 2024).

### 3.3 Narrativas Interativas e RPGs Textuais

Os jogos de *Role-Playing Game* (RPG) constituem uma forma de narrativa interativa na qual os jogadores assumem o papel de personagens inseridos em um mundo fictício regido por regras, personagens e eventos (YANNAKAKIS; TOGELIUS, 2018). Diferentemente de mídias narrativas tradicionais, como livros ou filmes, os RPGs se caracterizam pela participação ativa do jogador na construção da história, cujas ações influenciam diretamente o desenrolar da narrativa (HIGGINS, 1998).

No contexto dos RPGs textuais, essa interação ocorre predominantemente por meio da linguagem natural, seja na forma de descrições, diálogos ou comandos narrativos (MONTFORT, 2005). Esse formato estabelece um terreno fértil para a aplicação de modelos de linguagem, ao mesmo tempo em que impõe desafios específicos relacionados à coerência, consistência e mediação narrativa (RIEDL; BULITKO, 2013).

#### 3.3.1 Conceito de RPG Narrativo

Um RPG narrativo pode ser definido como um sistema interativo no qual a progressão da história emerge das decisões dos jogadores, mediadas por um conjunto de regras e por um agente narrador (YANNAKAKIS; TOGELIUS, 2018). O foco principal desse tipo de jogo não está apenas na resolução mecânica de ações, mas na construção compartilhada de uma narrativa coerente e significativa, o que caracteriza o RPG como uma forma de "fantasia compartilhada" (FINE, 2002).

Nos RPGs narrativos, a história não é rigidamente pré-definida. Em vez disso, ela se desenvolve de forma dinâmica, a partir da interação entre o jogador, o mundo fictício e o narrador (SYLVESTER, 2013). Esse caráter emergente distingue os RPGs narrativos de jogos baseados exclusivamente em narrativas lineares ou em estruturas fortemente roteirizadas, permitindo que o sistema funcione como um "motor de histórias" em vez de um roteiro fixo (TEKINBAS; ZIMMERMAN, 2003).

#### 3.3.2 O Papel do Narrador (Dungeon Master)

Tradicionalmente, o papel do *Dungeon Master* (DM) é central em RPGs de mesa e em suas adaptações digitais. O DM atua como mediador entre o mundo do jogo e os jogadores, sendo responsável por descrever ambientes, interpretar persona-

gens não jogáveis (NPCs), apresentar consequências às ações do jogador e manter a coerência narrativa do universo fictício (TYCHSEN et al., 2005).

É importante destacar que o DM não atua como o criador absoluto da história, mas como um facilitador da narrativa emergente (FINE, 2002). Suas decisões são, idealmente, guiadas por regras explícitas, pelo estado do mundo e pelas ações dos jogadores. Dessa forma, o narrador exerce um papel interpretativo e descritivo, enquanto a lógica do sistema funciona como uma "âncora" de verossimilhança (TYCHSEN et al., 2005; ELIAS; GARFIELD; GUTSCHERA, 2012).

Essa distinção entre mediação narrativa e autoridade mecânica é fundamental para compreender a proposta do AI Dungeon Master, no qual o agente narrativo assume funções semelhantes às do DM tradicional, sem concentrar em si a responsabilidade pela validação das regras do sistema (RIEDL; BULITKO, 2013).

### 3.3.3 Elementos Fundamentais de RPGs Textuais

Os RPGs textuais podem ser compreendidos como sistemas compostos por um conjunto de elementos fundamentais, cuja interação sustenta a experiência narrativa (MONTFORT, 2005):

- a) Mundo persistente: o universo do jogo deve manter um estado consistente ao longo do tempo, preservando informações sobre locais, eventos, objetos e personagens. A persistência do mundo é essencial para garantir continuidade narrativa e credibilidade das interações (BARTLE, 2004).
- b) Personagens: incluem tanto o personagem do jogador quanto os personagens não jogáveis (NPCs). Esses agentes devem apresentar comportamentos e características relativamente estáveis, de modo a reforçar a sensação de identidade e coerência dentro do mundo fictício (ISBISTER, 2022).
- c) Regras: definem as possibilidades e limitações das ações do jogador, regulando aspectos como movimentação, combate, uso de itens e interações sociais. As regras são responsáveis por estabelecer previsibilidade e justiça dentro do sistema (TEKINBAS; ZIMMERMAN, 2003).
- d) Consequências: toda ação significativa deve produzir efeitos perceptíveis no mundo do jogo. A relação entre ação e consequência é um dos pilares da experiência de RPG, pois confere peso narrativo às decisões do jogador (HIGGINS, 1998).

A ausência ou fragilidade de qualquer um desses elementos tende a comprometer a imersão e a coerência da experiência narrativa, especialmente em interações prolongadas (TYCHSEN et al., 2005).

### 3.3.4 Narrativa Linear versus Narrativa Emergente

Narrativas lineares caracterizam-se por uma progressão fixa de eventos, na qual as escolhas do jogador possuem impacto limitado ou meramente cosmético (TEKINBAS; ZIMMERMAN, 2003). Esse modelo é comum em jogos fortemente roteirizados, nos quais a história principal segue um caminho previamente definido.

Em contraste, os RPGs narrativos se fundamentam na ideia de narrativa emergente, na qual a história é construída dinamicamente a partir das interações entre jogador, sistema e narrador (SYLVESTER, 2013). Nesse modelo, as ações do jogador podem alterar significativamente o estado do mundo, influenciar personagens e modificar o rumo da narrativa, o que requer que o sistema de jogo funcione como um conjunto de regras que gera resultados imprevistos (JUUL, 2011).

Embora a narrativa emergente proporcione maior liberdade e engajamento, ela também impõe desafios adicionais em termos de coerência e controle do estado do mundo (TYCHSEN et al., 2005). A mediação adequada entre liberdade narrativa e consistência sistêmica torna-se, portanto, um aspecto crítico no projeto de RPGs narrativos assistidos por LLMs (RIEDL; BULITKO, 2013).

### 3.3.5 Implicações para Sistemas Narrativos Assistidos por LLMs

A análise dos RPGs textuais e de seus elementos fundamentais evidencia que a narrativa, embora central, não deve ser tratada de forma isolada. A coerência da experiência depende de uma articulação cuidadosa entre narrativa, regras e estado do mundo (ELIAS; GARFIELD; GUTSCHERA, 2012).

Nesse contexto, delegar integralmente ao modelo de linguagem o papel de narrador, juiz de regras e gestor do estado do mundo tende a gerar inconsistências e perda de previsibilidade. Torna-se, assim, necessário um modelo arquitetural no qual a função narrativa seja claramente delimitada, permitindo que a IA foque na expressividade linguística enquanto o motor de jogo garante a integridade lógica (SWANSON; GORDON, 2012).

Essa perspectiva fundamenta o papel do *NarrativeAgent* no AI Dungeon Master, concebido como um mediador narrativo responsável por interpretar ações e descrever consequências, mas não por decidir, de forma autônoma, a validade mecânica das ações ou as alterações estruturais do mundo de jogo (PARK et al., 2023). Esta abordagem alinha-se ao conceito de arquiteturas desacopladas, onde a geração de conteúdo é guiada por restrições sistêmicas externas (RIEDL; BULITKO, 2013).

## 3.4 Representação e Persistência de Estado em Sistemas Interativos

A representação e a persistência do estado do mundo constituem elementos centrais em sistemas interativos, especialmente em jogos de RPG narrativos (BARTLE, 2004). Diferentemente de sistemas puramente descritivos, esses ambientes dependem da manutenção contínua de informações sobre personagens, locais, objetos, eventos passados e consequências de ações anteriores (ELIAS; GARFIELD; GUTSCHERA, 2012). A ausência de mecanismos adequados para modelar e preservar esse estado compromete diretamente a coerência e a credibilidade da experiência, uma vez que o sistema perde a capacidade de "lembrar" as transformações causadas pelo jogador (TEKINBAS; ZIMMERMAN, 2003).

No contexto de sistemas assistidos por modelos de linguagem de grande porte, a gestão do estado do mundo assume papel ainda mais crítico. Embora LLMs sejam capazes de gerar descrições narrativas plausíveis e contextualizadas, elas não operam, por natureza, sobre representações formais persistentes de estado, o que impõe limitações significativas quando utilizadas como núcleo único de sistemas interativos complexos (JI et al., 2023; PARK et al., 2023).

### 3.4.1 Conceito de estado em sistemas interativos

Em sistemas interativos, o conceito de estado refere-se ao conjunto de informações que descrevem a situação atual do sistema em determinado momento (ELIAS; GARFIELD; GUTSCHERA, 2012). Esse estado pode incluir variáveis explícitas, como posições de personagens, atributos, inventário e condições ambientais, bem como relações implícitas entre entidades e eventos passados que influenciam decisões futuras. Em jogos, o estado do mundo define o conjunto de ações possíveis, suas consequências e a forma como o sistema reage às interações do jogador.

A correta modelagem do estado é essencial para garantir consistência temporal e causal (BJÖRK; HOLOPAINEN, 2005). Ações realizadas em um instante devem produzir efeitos observáveis e persistentes, evitando contradições internas e rupturas na lógica do mundo fictício. Assim, a gestão do estado transcende a narrativa, configurando-se como um problema fundamental de engenharia de sistemas (YANNAKAKIS; TOGELIUS, 2018).

### 3.4.2 Persistência de mundo em jogos

A persistência do mundo é uma característica essencial de RPGs, na qual eventos passados influenciam o presente e moldam o futuro da narrativa. Elementos como

decisões do jogador, mudanças no ambiente e evolução de personagens devem ser registrados e reutilizados ao longo da campanha. Sistemas tradicionais implementam essa persistência por meio de estruturas de dados formais e bancos de dados, garantindo que o mundo evolua de forma consistente e mantenha a integridade da simulação (BARTLE, 2004).

Em RPGs narrativos assistidos por LLMs, a persistência torna-se um desafio adicional, pois a geração textual, por si só, não assegura a retenção fiel de informações ao longo do tempo. Sem mecanismos explícitos de armazenamento, o sistema depende exclusivamente da memória contextual do modelo, que é limitada e sujeita a variações probabilísticas que podem comprometer a continuidade lógica da história (ZHAO et al., 2023).

### 3.4.3 Problemas de estado implícito em modelos de linguagem

Modelos de linguagem de grande porte operam predominantemente sobre contexto textual implícito, utilizando janelas de contexto finitas para inferir relações entre eventos, personagens e ações. Embora essa abordagem seja suficiente para tarefas de curta duração, ela se mostra inadequada para interações prolongadas e sistemas que exigem consistência rigorosa, uma vez que a atenção do modelo é diluída à medida que o volume de informações aumenta (VASWANI et al., 2017).

Entre os principais problemas associados ao estado implícito em LLMs destacam-se o esquecimento de informações relevantes e a introdução de contradições narrativas, fenômenos que contribuem para o chamado *narrative drift*, no qual o mundo ficcional perde coesão ao longo do tempo (JI et al., 2023). Além disso, a ausência de uma representação formal dificulta a validação de ações e a aplicação previsível de consequências, prejudicando a jogabilidade e a confiança do usuário no sistema, que passa a operar sob uma lógica puramente probabilística em vez de factual.

### 3.4.4 Representações estruturadas versus memória textual

Uma alternativa à dependência exclusiva da memória textual das LLMs é o uso de representações estruturadas do estado do mundo. Essas representações podem assumir a forma de grafos, tabelas ou modelos de dados que descrevem explicitamente entidades e suas relações. Ao manter o estado de forma estruturada, o sistema dispõe de uma fonte confiável e consultável de informações, permitindo o que se denomina raciocínio simbólico, independente da fluidez da geração narrativa (GARNELO; SHANAHAN, 2019).

Em contraste, a memória textual, embora flexível, carece de garantias formais de consistência e persistência. Enquanto representações estruturadas favorecem vali-

dação e controle lógico, a memória textual oferece maior liberdade criativa e naturalidade linguística. Sistemas narrativos robustos tendem a combinar essas abordagens em arquiteturas híbridas, utilizando estruturas formais para governar o estado e a lógica, e memória textual como suporte à expressividade da geração narrativa (SURÍS; MENON; VONDRICK, 2023).

#### 3.4.5 Trade-offs entre memória simbólica e narrativa

A escolha entre memória simbólica estruturada e memória narrativa textual envolve trade-offs importantes. A memória simbólica proporciona precisão, previsibilidade e facilidade de validação, mas pode limitar a expressividade narrativa se utilizada de forma isolada. Por outro lado, a memória narrativa favorece fluidez e criatividade, porém carece de mecanismos confiáveis para garantir coerência a longo prazo (RIEDL; BULITKO, 2013).

Abordagens híbridas têm se mostrado particularmente eficazes ao combinar o melhor de ambos os paradigmas. Nesses sistemas, o estado do mundo é mantido por estruturas simbólicas persistentes, enquanto a LLM é empregada para interpretar intenções e produzir narrativas coerentes com esse estado. Essa separação permite preservar a integridade lógica do mundo de jogo sem sacrificar a riqueza da experiência narrativa (HU et al., 2023).

Dessa forma, a representação e a persistência explícita do estado do mundo configuram-se como requisitos fundamentais para sistemas narrativos interativos assistidos por Inteligência Artificial. Ao tratar o estado como um componente de engenharia, e não apenas como um subproduto da narrativa, torna-se possível construir experiências mais coerentes e imersivas, mitigando limitações inerentes ao uso isolado de modelos de linguagem de grande porte (PARK et al., 2023).

## 4 Desenvolvimento

### 4.1 Visão geral do sistema

Esta seção apresenta uma visão geral do método proposto neste trabalho, descrevendo a arquitetura do sistema **AI Dungeon Master** e os principais fluxos de processamento de entrada do jogador. O objetivo é oferecer uma compreensão global do funcionamento do sistema antes do detalhamento individual de cada módulo nas subseções seguintes.

O sistema foi projetado para interpretar e responder às ações do jogador em um ambiente narrativo interativo, utilizando modelos de linguagem de grande porte (LLMs) combinados com mecanismos de validação, simulação e reação do mundo. Para isso, a arquitetura adota uma abordagem modular, na qual cada componente possui responsabilidades bem definidas, permitindo extensibilidade, controle narrativo e coerência do estado do mundo.

Essa modularização permite o emprego de LLMs com diferentes capacidades em cada etapa do processamento. Estrategicamente, tarefas mecânicas são delegadas a modelos menores e eficientes, enquanto atividades que demandam raciocínio complexo e síntese narrativa utilizam modelos mais robustos. Tal abordagem otimiza os custos computacionais, amplia a flexibilidade experimental e possibilita a evolução de componentes específicos de forma isolada, sem comprometer a integridade dos demais módulos do sistema.

As interações do jogador são classificadas em três tipos principais de entrada: **DO**, **SAY** e **SEE**. Cada tipo representa uma intenção distinta e, conseqüentemente, segue um fluxo de processamento específico, com diferentes níveis de validação, impacto narrativo e influência sobre o estado do mundo.

O fluxo mais complexo refere-se às entradas do tipo **DO** (Figura 1), que representam ações diretas do jogador, como combate ou interação com objetos. Inicialmente, o *input* é processado pelo *Context Builder*, que gera o *GameState* da cena. Na seqüência, o *ActionResolverAgent* classifica a ação em um *Intent*, submetendo-o às etapas de validação (*Intent Validator*) e, se necessário, de probabilidade (*SkillCheck Validator*).

Após as validações, o *Context Builder* estrutura os dados específicos do *intent* para o *NarrativeAgent*, que sintetiza a resposta narrativa. Simultaneamente, o *WorldReactionEngine* avalia as conseqüências da ação, garantindo a atualização das mecânicas de jogo e a reatividade do mundo.

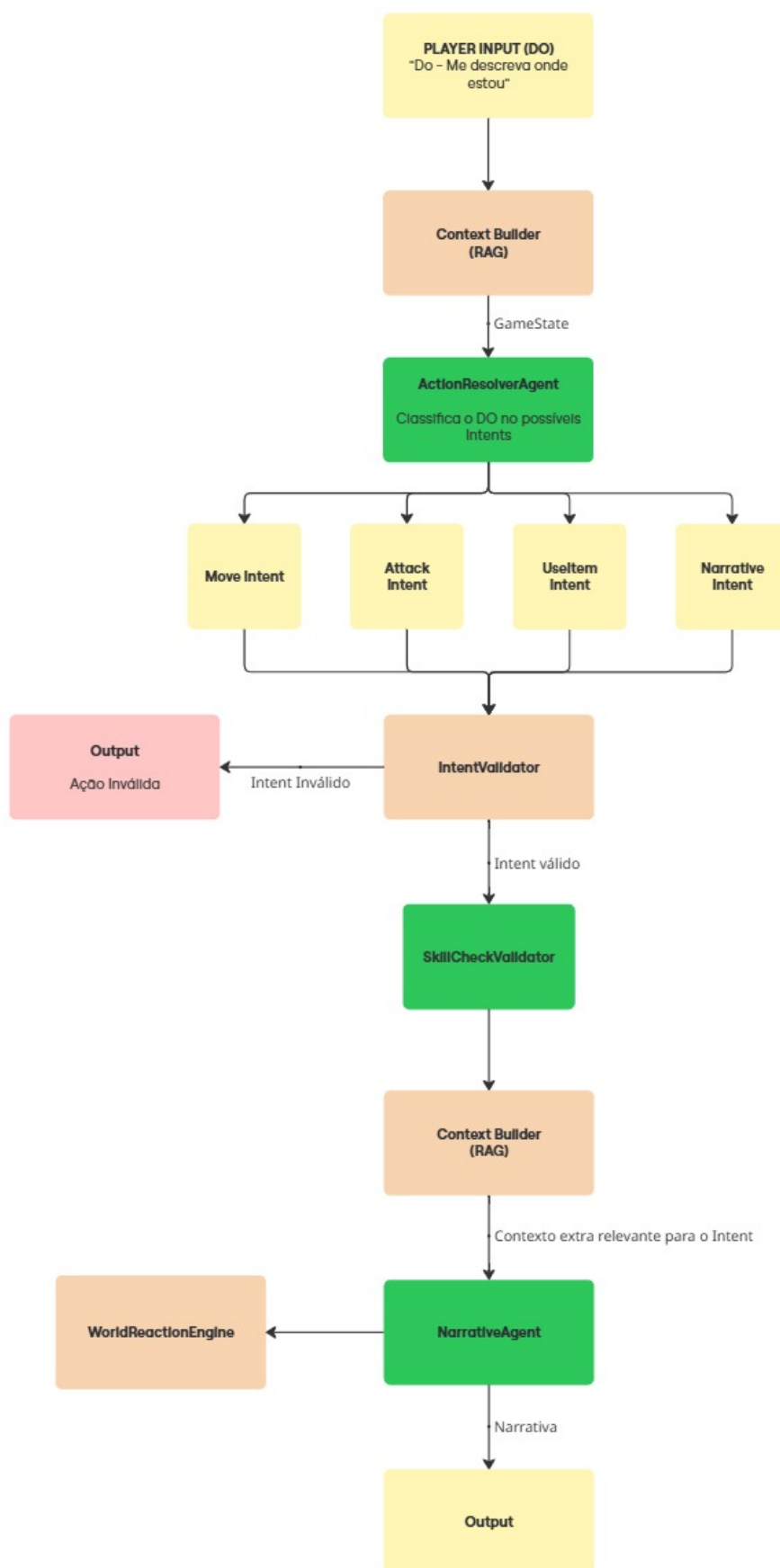


Figura 1 – Fluxograma input DO

As entradas do tipo **SAY** representam falas do jogador, ilustrado na **Figura 2**, direcionadas a personagens não jogáveis (NPCs) ou ao ambiente narrativo. Esse tipo de ação é sempre permitido, não exigindo validações de *intent* ou testes de habilidade. O fluxo inicia-se no **Context Builder**, que gera o **GameState** e incorpora informações relevantes do mundo ou do NPC alvo, possivelmente enriquecidas por mecanismos de *Retrieval-Augmented Generation* (RAG). A resposta é então produzida pelo **NarrativeAgent**, e, diferentemente do fluxo **SEE**, as falas do jogador podem provocar reações no mundo e avançar o tempo narrativo, sendo processadas também pelo **WorldReactionEngine**.

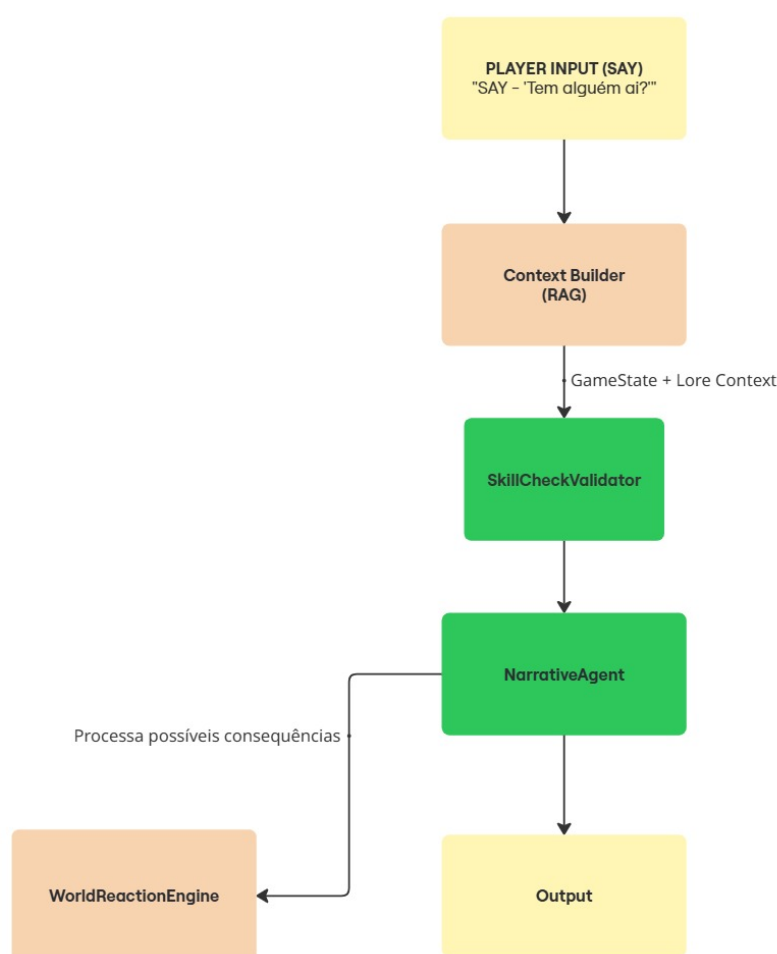


Figura 2 – Fluxograma input SAY

Por fim, as entradas do tipo **SEE** correspondem a ações perceptivas, ilustrado na **Figura 3**, nas quais o jogador solicita a descrição de elementos da cena ou de personagens presentes no ambiente. Esse fluxo é intencionalmente mais simples: o **Context Builder** fornece apenas o **GameState** atual, sem enriquecimento adicional

de lore ou histórico, uma vez que o jogador só pode perceber aquilo que está imediatamente disponível na cena. O **NarrativeAgent** então gera a descrição solicitada. Esse tipo de interação não gera consequências diretas no mundo nem provoca avanço temporal, funcionando exclusivamente como um mecanismo de observação.

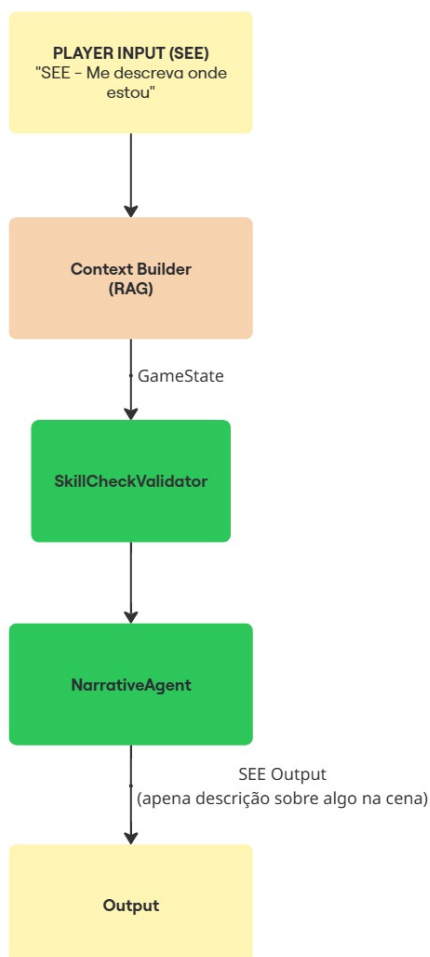


Figura 3 – Fluxograma input SEE

De forma geral, essa separação em fluxos distintos permite ao sistema equilibrar liberdade narrativa, coerência do mundo simulado e controle sobre as consequências das ações do jogador, além de reduzir a complexidade computacional em interações mais simples.

## 4.2 Persistência de Dados

A persistência de dados no AI Dungeon Master é estruturada de forma híbrida e intencionalmente desacoplada, refletindo a diversidade semântica das informações

manipuladas pelo sistema durante a execução de uma campanha narrativa interativa. Essa abordagem combina três mecanismos distintos e complementares: uma *Retrieval-Augmented Generation* (RAG) baseada em **grafos** para o **estado do mundo**, uma RAG tradicional apoiada em um **espaço vetorial** para **conhecimento narrativo e lore**, e uma **estrutura tabular** convencional para o **registro histórico das interações**.

O estado persistente do mundo de jogo é representado por meio de uma base de dados orientada a grafos, caracterizando uma RAG em grafo (Figura 4). Nesse modelo, entidades centrais como personagens, itens e localizações são modeladas como nós, enquanto suas relações, tais como presença espacial, posse, equipamentos, vínculos sociais ou estados contextuais, são representadas explicitamente por arestas tipadas. Esse formato permite consultas estruturais e contextuais sobre o mundo, possibilitando ao sistema recuperar, de forma precisa e controlada, apenas as entidades e relações relevantes para um determinado estado ou intenção do jogador. Ao ser integrada ao fluxo de geração narrativa, essa base atua como uma fonte estruturada de recuperação, fornecendo contexto factual e relacional ao modelo de linguagem sem delegar a ele a manutenção implícita do estado, reduzindo assim inconsistências e alucinações.



Figura 4 – Grafo de Entidades

Complementarmente, informações de natureza mais descritiva, abstrata ou narrativa, como lore do mundo, conhecimento específico de personagens, eventos históricos relevantes e descrições não estruturadas, são armazenadas em uma base documental indexada em um espaço vetorial, caracterizando uma RAG tradicional baseada em similaridade semântica. Nesse caso, documentos textuais são recuperados conforme a relevância para a consulta atual, levando em conta fatores como o alvo da interação, o ambiente e o contexto narrativo imediato. Esse mecanismo é especialmente utilizado em *intents* como *SAY* e *NARRATIVE*, nos quais a riqueza textual e a coerência narrativa são prioritárias, mas sem a necessidade de modificações diretas no estado mecânico do mundo.

Por fim, o sistema mantém um histórico persistente da campanha em uma estrutura tabular convencional, organizada cronologicamente. Cada entrada desse histórico registra, de forma resumida, a ação realizada pelo jogador e a resposta produzida pelo

sistema, incluindo decisões narrativas e efeitos observáveis. Diferentemente das bases de RAG, essa tabela não tem como objetivo principal a recuperação semântica, mas sim a rastreabilidade, auditoria e reconstrução do fluxo narrativo da campanha ao longo do tempo. Além disso, esse histórico serve como apoio para a construção de contextos de curto prazo, permitindo ao sistema resumir ações recentes sem a necessidade de reprocessar todo o estado do mundo.

Em conjunto, essa arquitetura de persistência reforça a separação de responsabilidades entre estado mecânico, conhecimento narrativo e memória histórica, ao mesmo tempo em que fornece conteúdo adequados para a geração de narrativas coerentes, consistentes e contextualizadas. A combinação de RAG em grafo, RAG vetorial e histórico tabular contribui diretamente para a estabilidade do sistema, para a redução de dependência implícita do modelo de linguagem e para a reprodutibilidade das interações narrativas ao longo da campanha.

#### 4.2.1 Campanha *Toy*

Em razão das limitações de escopo, tempo de desenvolvimento e da necessidade de garantir estabilidade e previsibilidade para fins de avaliação experimental, a campanha utilizada nos testes do AI Dungeon Master foi concebida como um cenário *toy*. Nesse contexto, as entidades fundamentais do mundo, tais como personagens, itens e localizações, foram previamente definidas e instanciadas por meio de *scripts* de inicialização, em vez de serem geradas de forma totalmente autônoma por modelos de linguagem. Essa decisão permitiu isolar e avaliar de maneira mais controlada os mecanismos centrais do sistema, como a resolução de intenções, a validação de ações, a atualização de estado e a geração narrativa, sem introduzir variáveis adicionais associadas à criação dinâmica de entidades. Ressalta-se, contudo, que essa escolha não limita a arquitetura proposta, a qual foi concebida para suportar, em trabalhos futuros, a geração automática e incremental de entidades e conteúdos narrativos por meio de LLMs.

A campanha de teste (`campaign_001`) compreende um conjunto de entidades interconectadas que estabelecem o estado inicial e o escopo de interatividade do mundo de jogo. A estrutura detalhada deste cenário está consolidada nas Tabelas 1, 2 e 3, que apresentam, respectivamente, as propriedades das localizações geográficas, os perfis e papéis dos personagens, e a especificação de itens e equipamentos disponíveis.

Tabela 1 – Detalhamento das Localizações e Ambiências da Campanha.

Nome	Descrição	Ambiência	Conexões
Villa Verde	Vila tranquila com colinas e rio sereno.	Pacífico (pássaros)	Taverna
Taverna do Porco	Estabelecimento rústico com cheiro de cerveja.	Animado (música)	Villa Verde, Cozinha
Cozinha	Cozinha simples com cheiro de comida caseira.	Cheiro de comida	Taverna

Fonte: Elaborada pelo autor (2026).

Tabela 2 – Atributos e Papéis dos Personagens (Agentes).

Nome	Papel	Descrição	Vida	Atributos*
John Doe	PLAYER	Aventureiro atento e curioso.	120	5 (Base)
Taverneiro	NPC	Homem robusto limpando copos.	65	0
Bêbado	NPC	Homem sujo e embriagado.	50	0

Fonte: Elaborada pelo autor (2026).

\*Atributos base incluem Força, Agilidade, Inteligência, etc.

Tabela 3 – Propriedades dos Itens e Equipamentos.

Item	Tipo	Dano	Peso	Efeito Especial
Espada Enferrujada	ARMA	15	5	-
Adaga Aterrorizante	ARMA	8	3	-
Poção de Cura	CONSUMÍVEL	0	1	Cura Leve (20 HP)
Tocha	FERRAMENTA	2	2	Fonte de Luz
Caneca de Cerveja	CONSUMÍVEL	0	1	-

Fonte: Elaborada pelo autor (2026).

#### 4.2.1.1 Estado Global e Ambiente (WorldState)

O mundo encontra-se em um estado de manhã (10:45), sob condições climáticas ensolarada, mas com iluminação clara. Essas variáveis globais servem como modificadores de contexto para as descrições geradas pela LLM e podem influenciar mecânicas de visibilidade e locomoção.

### 4.3 Processamento da Entrada do Jogador

#### 4.3.1 Player input

O sistema recebe como entrada ações do jogador expressas em linguagem natural, acompanhadas da seleção explícita do tipo de intenção principal, definida no *frontend* como **SEE**, **SAY** ou **DO**. Essa classificação inicial permite ao sistema distinguir entre ações de observação, fala e execução, direcionando o processamento para fluxos específicos da arquitetura. Embora o tipo de intenção seja informado de forma

clara pelo jogador, o conteúdo textual da ação permanece livre, ambíguo e narrativo, exigindo interpretação semântica para determinar, especialmente no caso do **DO**, qual ação concreta foi pretendida e quais regras devem ser aplicadas.

No caso do **SEE**, o jogador solicita informações perceptivas sobre o ambiente ou elementos presentes na cena atual, sem alterar o estado do mundo ou avançar o tempo narrativo. Já o **SAY** representa ações de fala dirigidas a outros personagens ou ao ambiente, sendo sempre permitido, mas potencialmente capaz de gerar reações, consequências narrativas e alterações no estado do mundo. O tipo **DO**, por sua vez, concentra as ações executivas do jogador, como mover-se, atacar, usar itens ou interagir com objetos, e demanda validações adicionais para garantir aderência às regras do sistema e coerência com o estado atual do mundo.

Apesar da seleção explícita do tipo de ação, o sistema deve lidar com entradas textuais naturalmente imprecisas, nas quais intenções podem ser implícitas ou expressas de forma indireta. Essa característica reflete a forma como jogadores humanos se comunicam em RPGs narrativos, priorizando expressividade e fluidez em detrimento de comandos formais.

Exemplos de entradas típicas incluem:

- **SEE (Observar):** Comandos voltados à percepção do ambiente e detalhes visuais.
  - *“Olho ao redor da taverna.”*
  - *“Examino o homem encostado no balcão.”*
  - *“Observo o que há sobre a mesa.”*
- **SAY (Falar):** Interações verbais e diálogos.
  - *“Pergunto ao taverneiro sobre rumores na cidade.”*
  - *“Grito para o bêbado se afastar.”*
  - *“Digo em tom ameaçador que ele deve ir embora.”*
- **DO (Agir):** Ações físicas que podem alterar o estado do mundo ou disparar mecânicas.
  - *“Ataco o goblin com minha espada.”*
  - *“Finjo atacar para intimidá-lo.”*
  - *“Pego a caneca e guardo na mochila.”*
  - *“Caminho até o balcão.”*

Esses exemplos ilustram como, mesmo com a intenção principal explicitada, o conteúdo da ação pode envolver ambiguidades narrativas, ações simbólicas ou tentativas que não correspondem diretamente a uma ação mecânica concreta. Dessa forma, o processamento da entrada do jogador é projetado para priorizar a interpretação da intenção semântica, preservando a liberdade expressiva do jogador sem comprometer a previsibilidade e a coerência do sistema.

### 4.3.2 Context Builder (Estado Básico do Jogo)

O **Context Builder** é o componente responsável por consolidar e estruturar as informações do estado atual do jogo (**GameState**) antes que qualquer interpretação semântica, validação mecânica ou geração narrativa seja realizada. Nessa etapa, sua função é estritamente a de materializar e organizar o estado persistente do mundo em uma representação explícita e consistente, que será utilizada como base para validações posteriores, quando necessárias, como no fluxo *DO*, e em todos os casos, como contexto de entrada para o *NarrativeAgent*.

Durante esse processo, o **Context Builder** não executa inferências, não interpreta intenções e não aplica regras do jogo. Ele atua exclusivamente como um componente de preparação de contexto, responsável por expor, de forma controlada, as informações relevantes do mundo de jogo aos módulos subsequentes. Nos fluxos *SEE* e *SAY*, esse **GameState** é utilizado diretamente pelo *NarrativeAgent*, sem passar por etapas de validação mecânica. Já no fluxo *DO*, o mesmo estado estruturado serve como insumo para os módulos de validação e resolução de consequências, garantindo que qualquer decisão narrativa esteja ancorada em um estado de mundo consistente e verificável.

A Tabela 5.7 apresenta um exemplo da estrutura de dados que compõe o estado do sistema (*GameState*), detalhando as variáveis de ambiente, localização e entidades presentes no mundo fictício.

#### Listing 4.1 – Exemplo de representação do estado do sistema (GameState).

```
World: manha, ensolarado, claro, 10:50

Location: Taverna do Porco
  Description: Uma taverna rustica, com cheiro de cerveja.
  Ambience: animado com risadas e musica
  Light: semiborroadado
  Weather: quente
  Connected Locations: Villa Verde, Cozinha

Location Map:
```

```
[Entrada] (->Villa Verde)
  |
[Mesas] -- [Mesa Central]
  |
[Balcao] (->Cozinha)

Characters:
- (NPC): Taverneiro [Balcao], limpando copos
- (NPC): Bebado Barulhento [Mesa Central], bebendo
- (PLAYER): John Doe [Balcao], observando

Items:
- Caneca de Cerveja [Mesa Central], sobre a mesa

Player Status:
Health: 100, Area: [Balcao]
Equipped: Espada Enferrujada, Holding: Tocha
```

Fonte: Elaborado pelo autor (2026).

Esse processo é fundamental para reduzir ambiguidades, evitar contradições narrativas e limitar o espaço de geração dos modelos, garantindo que toda interpretação ou narração esteja ancorada exclusivamente nas informações válidas do mundo de jogo naquele momento.

### 4.3.3 ActionResolverAgent

O **ActionResolverAgent** é o componente responsável por interpretar a entrada textual do jogador e classificá-la em um *Intent* explícito, que representa a ação pretendida dentro do sistema. Nesse estágio, o objetivo não é compreender plenamente as implicações narrativas ou mecânicas da ação, mas identificar, de forma estruturada, o *tipo de ação* que o jogador deseja realizar.

Para isso, o agente adota uma estratégia híbrida de classificação. Inicialmente, são utilizadas funções determinísticas baseadas em padrões linguísticos e sinônimos comuns da linguagem natural, capazes de identificar intents a partir de verbos e expressões recorrentes. Por exemplo, termos como “avanço”, “me movimento”, “caminho até” ou “vou em direção a” são associados ao intent *MOVE*; expressões como “olho ao redor”, “examino”, “observo” ao intent *SEE*; enquanto “ataco”, “golpeio” ou “investo contra” são associados ao intent *ATTACK*. Abordagens semelhantes são aplicadas para os intents *SAY* e *USEITEM*.

Caso a entrada do jogador não possa ser classificada de forma confiável por essas heurísticas iniciais, o agente aciona um mecanismo de `llm_fallback`, no qual uma LLM é utilizada exclusivamente para classificar semanticamente a ação em um dos intents suportados pelo sistema. Mesmo nesse cenário, a LLM continua restrita ao papel de classificador, não sendo responsável por validar regras ou gerar narrativa.

Se, ainda assim, a entrada não puder ser adequadamente mapeada para nenhum dos intents mecânicos disponíveis, o sistema recorre ao intent *NARRATIVE*. Esse intent funciona como um “coringa” semântico, destinado a capturar ações descritivas, subjetivas ou ambíguas que expressam intenções narrativas do jogador, mas que não correspondem diretamente a uma ação mecânica formal.

Dessa maneira o uso de LLMs mais custosas fica como “último recurso” do agente, assim preservando capacidade computacional e reduzindo tempo de requisição e processamento.

Os intents atualmente suportados pelo sistema são descritos a seguir:

- **SEE**: representa ações de observação ou inspeção do ambiente. Não possui consequências mecânicas diretas nem avança o tempo do mundo, sendo utilizado apenas para revelar informações perceptíveis ao jogador.
- **SAY**: representa ações de fala ou comunicação verbal. É sempre permitido, avança o estado do mundo e pode gerar consequências, como reações de personagens não jogáveis (NPCs) ou alterações narrativas.
- **MOVE**: representa deslocamentos físicos do jogador entre áreas ou posições. Possui consequências mecânicas diretas, como mudança de localização e possível avanço temporal.
- **ATTACK**: representa ações ofensivas direcionadas a alvos específicos. Está sujeito a validações adicionais e testes de habilidade, produzindo consequências mecânicas diretas.
- **USEITEM**: representa o uso de itens ou objetos do inventário ou do ambiente. Pode gerar efeitos diretos no estado do mundo, dependendo do item e do contexto.
- **NARRATIVE**: representa intenções narrativas livres, como gestos, atitudes subjetivas ou ações implícitas que não se enquadram claramente em ações mecânicas. Não possui consequências mecânicas diretas, mas pode gerar consequências indiretas por meio da reação narrativa do mundo.

Do ponto de vista arquitetural, essa abordagem reduz significativamente a liberdade excessiva do modelo de linguagem, impedindo que ele “invente” ações inexistentes ou aplique consequências incompatíveis com o estado atual do jogo. Ao mesmo tempo, preserva a expressividade da linguagem natural, permitindo que o jogador se comunique de forma fluida e não restrita a comandos rígidos.

Assim, o **ActionResolverAgent** atua como um intermediário fundamental entre a linguagem livre do jogador e a estrutura formal do sistema, aproximando-se de um *parser semântico flexível*, capaz de lidar com ambiguidade sem comprometer a previsibilidade, a coerência e a integridade do mundo de jogo.

#### 4.3.4 Intent Validator

Uma vez que a entrada do jogador foi classificada em um *Intent* explícito pelo **ActionResolverAgent**, torna-se necessário verificar se essa intenção é válida no contexto atual do jogo. Essa verificação é realizada pelo **Intent Validator**, componente responsável por garantir que a ação pretendida seja permitida, coerente com as regras do mundo e compatível com o estado corrente do jogo.

O **Intent Validator** opera exclusivamente sobre o **GameState** previamente construído, utilizando informações como localização do jogador, entidades presentes na cena, estado de personagens e itens disponíveis. A partir desses dados, o sistema avalia se a ação solicitada pode, de fato, ser executada naquele momento. Dessa forma, a validação não depende de interpretação subjetiva ou narrativa, mas de regras explícitas e verificáveis.

Para cada tipo de intent existe uma função de validação específica. Alguns intents possuem validações mais simples, enquanto outros exigem verificações mais rigorosas. Por exemplo, o intent *SEE* é sempre considerado válido, desde que o objeto ou entidade solicitada esteja presente ou perceptível na cena. O intent *SAY* também é sempre permitido, uma vez que falar não viola regras mecânicas do mundo, embora possa gerar consequências narrativas posteriores.

Em contrapartida, intents como *MOVE* e *ATTACK* estão sujeitos a validações mais severas. No caso de *MOVE*, o sistema verifica se o destino existe, se é acessível a partir da localização atual e se não há bloqueios lógicos ou narrativos. Para o intent *ATTACK*, são checados aspectos como a existência do alvo, sua presença na cena, seu estado (por exemplo, se já está incapacitado) e se a ação é permitida pelas regras do sistema naquele contexto. O intent *USEITEM* também exige validações relacionadas à posse do item, sua aplicabilidade e ao contexto da ação.

Quando um intent é considerado inválido, o fluxo de processamento é imediatamente interrompido. Nesse caso, o sistema retorna ao jogador uma resposta infor-

mando que a ação não pode ser realizada, explicando de forma narrativa ou mecânica o motivo da invalidação. Essa interrupção precoce é uma decisão arquitetural intencional, pois evita a execução de etapas posteriores mais custosas, especialmente chamadas a LLMs utilizadas no **NarrativeAgent**. Com isso, o sistema reduz custos computacionais e previne a geração de narrativas incoerentes baseadas em ações impossíveis.

Do ponto de vista de projeto, a existência do **Intent Validator** reforça a separação clara entre interpretação e validação. Enquanto a classificação do intent busca compreender o *que* o jogador deseja fazer, a validação responde à pergunta se essa ação pode ser realizada naquele estado do mundo. Essa distinção é fundamental para evitar que modelos de linguagem assumam o papel de árbitros das regras do jogo, garantindo maior previsibilidade, consistência mecânica e integridade do estado narrativo ao longo da campanha.

#### 4.3.5 SkillCheck Validator

Após a validação estrutural do intent, algumas ações exigem a resolução de incerteza por meio de testes mecânicos, típicos de sistemas de RPG. Essa responsabilidade é atribuída ao **SkillCheck Validator**, componente responsável por determinar se uma ação validada requer um teste de habilidade, qual atributo do personagem será utilizado, qual a dificuldade envolvida e como o resultado desse teste deve influenciar a narrativa subsequente.

O **SkillCheck Validator** não é acionado para todos os intents. Sua atuação é condicional e baseada no tipo de ação interpretada, refletindo a lógica de RPGs tradicionais, nos quais nem toda ação exige uma rolagem. A decisão sobre a necessidade de um teste é realizada por meio de regras explícitas associadas a determinados intents. Por exemplo, ações de ataque (*AttackIntent*) sempre requerem um teste baseado no atributo de força do jogador, enquanto ações de fala (*SayIntent*) direcionadas a um alvo específico podem demandar um teste de carisma. Da mesma forma, intents narrativos específicos, como intimidação ou enganação, também são associados a testes de atributos adequados, como carisma ou inteligência.

Quando um teste é requerido, o sistema executa uma rolagem simulada de um dado de vinte faces (*d20*), somando o valor obtido ao atributo correspondente do personagem, extraído diretamente do **GameState**. O valor total resultante é então comparado à dificuldade definida para aquele tipo de ação. Esse processo garante que os testes sejam resolvidos de forma determinística e reproduzível, sem delegar ao modelo de linguagem decisões mecânicas ou probabilísticas.

O resultado numérico do teste é posteriormente traduzido em um desfecho qua-

litativo, classificado em cinco categorias: *Critical Success*, *Success*, *Partial Success*, *Failure* e *Critical Failure*. Essa classificação considera tanto a margem entre o valor total e a dificuldade quanto resultados críticos baseados na rolagem do dado. Cada uma dessas categorias está associada a diretrizes narrativas explícitas, que orientam como o **NarrativeAgent** deve descrever o desfecho da ação.

Essas diretrizes não determinam a narrativa final, mas impõem limites claros ao escopo da narração. Por exemplo, um sucesso crítico permite a descrição de uma execução excepcional da ação, possivelmente com pequenas vantagens adicionais, enquanto um sucesso parcial exige a introdução de custos, limitações ou complicações narrativas. Em casos de falha ou falha crítica, o sistema orienta a introdução de consequências coerentes, evitando punições arbitrárias ou encerramentos prematuros da história. Dessa forma, o resultado do teste influencia diretamente o tom e o impacto da narrativa, sem permitir que o modelo de linguagem altere regras, estados ou crie entidades inexistentes.

Do ponto de vista arquitetural, o **SkillCheck Validator** cumpre um papel central na integração entre mecânicas tradicionais de RPG e geração narrativa assistida por LLMs. Ao externalizar a resolução de incerteza para um componente determinístico, o sistema evita narrativas excessivamente determinísticas ou arbitrárias, preserva a possibilidade de falha significativa e mantém compatibilidade conceitual com sistemas clássicos de RPG. Ao mesmo tempo, o uso de resultados qualitativos e diretrizes narrativas permite que a criatividade da geração de linguagem natural seja preservada dentro de limites controlados, reforçando a coerência e a previsibilidade do mundo de jogo

## 4.4 Construção de Contexto Específico por Intent

Após a identificação, validação e, quando aplicável, a resolução de testes de habilidade associados ao *intent* do jogador, o sistema executa a etapa de **Context Builder (Contexto Relevante)**. Diferentemente da construção inicial do *GameState*, utilizada principalmente para validações mecânicas, esta fase tem como objetivo preparar um **contexto narrativo específico e direcionado**, a ser consumido pelo *NarrativeAgent* durante a geração da resposta textual.

A função central deste componente é **selecionar, organizar e sintetizar informações do estado persistente do mundo** de acordo com o tipo de intent identificado, fornecendo ao modelo de linguagem apenas os elementos necessários para a narração da ação em curso. Essa abordagem evita a exposição indiscriminada do estado global do jogo, reduzindo ambiguidades, limitações de foco narrativo e custos computacionais associados a prompts extensos.

De forma geral, todos os contextos gerados pelo **Context Builder** podem incluir três categorias principais de informação: (i) um resumo das ações recentes, limitado a eventos relevantes para a situação atual; (ii) o conteúdo resultante de testes de habilidade, quando aplicável; e (iii) uma representação visual estruturada do estado do mundo, que descreve o ambiente imediato, entidades visíveis e suas disposições espaciais. A partir dessa base comum, o conteúdo retornado é adaptado conforme o intent específico, conforme descrito a seguir.

Para intents do tipo **SEE**, o contexto é estritamente observacional. Nessa situação, o **Context Builder** retorna apenas o conteúdo do *skillcheck*, quando existente, e a representação visual do *GameState*. Como esse tipo de ação não provoca alterações diretas no estado do mundo nem avança o tempo narrativo, não são incluídos históricos de ações ou informações adicionais sobre entidades além daquelas diretamente perceptíveis pelo jogador.

No caso do intent **MOVE**, o contexto é ampliado para incluir um resumo das últimas ações relevantes, permitindo preservar continuidade narrativa local, além do conteúdo do *skillcheck*, quando necessário. O contexto também incorpora informações sobre o **destino pretendido pelo jogador**, bem como a representação visual atual do *GameState*, possibilitando que a narrativa descreva a transição espacial de forma coerente.

Para ações do tipo **SAY**, o **Context Builder** desempenha um papel mais complexo. Além da lista resumida das ações recentes e do conteúdo do *skillcheck*, o sistema consulta mecanismos de *Retrieval-Augmented Generation* (RAG) para obter informações relevantes tanto para a consulta do jogador quanto para o alvo da interação, quando existente. Os resultados retornados pela RAG são avaliados por um **score de relevância**, que considera fatores como o personagem alvo, o ambiente atual e o contexto narrativo em curso. Apenas os trechos mais relevantes são incorporados ao contexto final, juntamente com a representação visual do *GameState*, permitindo respostas dialogais coerentes e contextualizadas.

No intent **ATTACK**, o contexto inclui o resumo das últimas ações, o conteúdo do *skillcheck* e a representação visual do estado do mundo, **acrescidos de informações detalhadas sobre o alvo da ação**, como pontos de vida, itens carregados e armas equipadas. Essas informações são essenciais para que a narrativa reflita corretamente o impacto mecânico do ataque, sem delegar à LLM decisões implícitas sobre regras ou estados críticos.

Para ações do tipo **USEITEM**, o contexto segue estrutura semelhante, incorporando a lista de ações recentes, o conteúdo do *skillcheck* e a representação visual do *GameState*, além de **informações estruturadas sobre o item utilizado**, como nome, descrição e tipo. Essa estratégia permite que o modelo de linguagem gere uma narra-

ção alinhada às propriedades do item.

Por fim, o intent **NARRATIVE**, utilizado como um mecanismo de fallback para entradas que não se enquadram claramente em ações mecânicas, recebe um contexto composto pelo resumo das ações recentes, pelo conteúdo do *skillcheck* e pela representação visual do *GameState*. Embora esse tipo de ação não produza consequências mecânicas diretas, o contexto fornecido permite que a LLM interprete a intenção narrativa do jogador e produza efeitos indiretos coerentes com o estado atual da campanha.

Do ponto de vista arquitetural, essa estratégia reforça o princípio do **contexto mínimo suficiente**, no qual cada intent recebe apenas as informações necessárias para sua correta interpretação e narração. Como resultado, o sistema reduz custos computacionais, melhora o foco narrativo das LLMs e diminui a ocorrência de inconsistências, preservando a criatividade da geração de linguagem natural sem comprometer a integridade do mundo de jogo.

## 4.5 Geração Narrativa

O **NarrativeAgent** é o componente responsável exclusivamente pela geração da narração textual apresentada ao jogador, a partir do intent validado e do contexto relevante produzido pelo *Context Builder*. Diferentemente de abordagens monolíticas baseadas em LLMs, esse agente **não participa da interpretação de intenções, da validação de regras ou da execução de alterações mecânicas no estado do mundo**, atuando apenas como mediador narrativo entre o sistema e o usuário.

Para garantir esse isolamento de responsabilidades, o NarrativeAgent opera a partir de **prompts especializados por tipo de intent**. Cada prompt é estruturado para incorporar informações do intent identificado, os elementos contextuais relevantes e, quando aplicável, os resultados de testes de habilidade (*skill check outcomes*). Essa estratégia permite ajustar o tom, o foco e o escopo da narração de acordo com a natureza da ação, evitando generalizações excessivas e reduzindo ambiguidades narrativas.

Embora o NarrativeAgent não possua autoridade para validar regras ou modificar diretamente o *GameState*, ele pode **bloquear narrativamente a execução de determinados intents em situações específicas**. Esse bloqueio não ocorre por violação mecânica, mas por **restrições narrativas plausíveis** derivadas do contexto da cena. Por exemplo, mesmo que uma ação de movimentação tenha sido validada mecanicamente, um personagem não jogável (NPC) presente na cena pode, de forma coerente com a narrativa, impedir a entrada do jogador em determinado local. Nesses casos, o NarrativeAgent produz uma descrição do impedimento, mantendo a coerência da história sem violar as regras formais do sistema.

Os *intents* que podem ser narrativamente barrados pelo NarrativeAgent incluem **ATTACK**, **MOVE** e **USEITEM**, uma vez que essas ações envolvem interação direta com o ambiente ou com outras entidades do mundo. Já *intents* como **SEE** e **SAY** não estão sujeitos a esse tipo de bloqueio, pois não implicam, por definição, alterações diretas ou invasivas no estado do mundo.

A saída do NarrativeAgent é encapsulada em uma estrutura denominada **NarrativeOutcome**, que contém três elementos principais: (i) um indicador explícito informando se o intent foi narrativamente permitido ou bloqueado; (ii) o texto narrativo gerado, descrevendo a ação, sua execução ou seu impedimento; e (iii) uma lista de **consequências indiretas**, isto é, eventos narrativos que não foram causados diretamente pelo intent do jogador, mas que surgem como reação plausível do mundo. Um exemplo típico ocorre quando, após um ataque, um NPC que não era o alvo direto da ação se desloca para outro local em resposta ao conflito.

Essas consequências indiretas não alteram imediatamente o estado mecânico do mundo, mas são posteriormente analisadas e aplicadas por componentes especializados, como o WorldReactionEngine. Dessa forma, o NarrativeAgent mantém seu papel restrito à narração e à mediação narrativa, sem assumir funções de simulação ou decisão sistêmica.

Do ponto de vista arquitetural, a separação explícita entre lógica de jogo e geração narrativa constitui um dos pilares centrais do AI Dungeon Master. Ao restringir o NarrativeAgent a um papel exclusivamente descritivo e interpretativo, o sistema evita que modelos de linguagem assumam responsabilidades críticas relacionadas à validação de regras, atualização de estado ou autorização de ações, tarefas para as quais LLMs não oferecem garantias formais de consistência ou verificabilidade. Essa decisão mitiga diretamente problemas recorrentes em abordagens monolíticas, como o *narrative drift*, a aplicação inconsistente de regras e a geração de consequências incompatíveis com o estado do mundo. Além disso, a modularização permite analisar e evoluir o componente narrativo de forma isolada, favorecendo avaliações mais controladas e reprodutíveis do impacto das LLMs na qualidade da narrativa, sem confundir falhas linguísticas com erros de modelagem do sistema. Dessa forma, a arquitetura proposta sustenta a hipótese central deste trabalho de que LLMs são mais eficazes quando atuam como mediadores narrativos dentro de um ecossistema determinístico, e não como engines completas de jogos narrativos

## 4.6 Reação do Mundo e Consequências

Em sistemas narrativos interativos, especialmente em jogos de RPG, o mundo de jogo não deve ser compreendido apenas como um cenário passivo que responde

pontualmente às ações do jogador, mas como um sistema ativo, capaz de reagir, adaptar-se e evoluir de forma autônoma a partir dos eventos ocorridos. Nesse contexto, o **WorldReactionEngine** é introduzido como o componente responsável por materializar essa noção de mundo ativo no AI Dungeon Master.

Após a geração narrativa realizada pelo *NarrativeAgent*, o **WorldReactionEngine** atua como a camada encarregada de interpretar os efeitos mecânicos e sistêmicos decorrentes das ações do jogador, bem como das consequências indiretas descritas na narrativa. Seu papel não é narrar eventos, mas aplicar transformações estruturadas e verificáveis no estado do mundo, garantindo que personagens, ambientes e relações evoluam de forma coerente ao longo do tempo.

Essa separação reforça a distinção entre **descrição narrativa e simulação do mundo**, assegurando que alterações no *GameState* não dependam exclusivamente de interpretações implícitas do texto gerado por modelos de linguagem. Dessa forma, o sistema evita que mudanças relevantes ocorram apenas no nível narrativo, sem reflexo concreto no estado persistente do mundo.

#### 4.6.1 DirectMechanicalReactor

O **DirectMechanicalReactor** é o componente responsável por aplicar, de forma imediata e determinística, as **consequências mecânicas diretas associadas a uma intenção previamente validada**. Seu objetivo central é garantir que ações permitidas pelo sistema resultem em **atualizações concretas e verificáveis no estado persistente do mundo** (*GameState*), independentemente da forma como essas ações são posteriormente descritas na narrativa.

Diferentemente de abordagens baseadas exclusivamente em modelos de linguagem, nas quais alterações de estado podem ocorrer apenas de forma implícita no texto gerado, o **DirectMechanicalReactor** opera sobre representações estruturadas do mundo, atualizando entidades, relações e variáveis de forma explícita. Essa camada atua como um mecanismo de simulação determinística, assegurando que regras do jogo sejam aplicadas de maneira consistente e reproduzível.

As reações mecânicas tratadas por esse módulo incluem, entre outras, movimentação, avanço do tempo, aplicação de dano, consumo ou uso de itens e alterações diretas no estado de personagens. Por exemplo, no caso de uma intenção de movimentação (*MoveIntent*), o **DirectMechanicalReactor** atualiza a relação entre o jogador e o local ou área de destino, registrando a mudança de posição no mundo e avançando o tempo global em uma quantidade pré-definida. Esse avanço temporal ocorre tanto em deslocamentos entre áreas distintas quanto em movimentações dentro de uma mesma área, refletindo o custo temporal da ação.

Do ponto de vista arquitetural, a existência do `DirectMechanicalReactor` permite uma separação clara entre o que acontece no mundo e como isso é narrado. Ao centralizar as reações mecânicas em um módulo determinístico, **o sistema evita narrativas contraditórias**, nas quais, em interações futuras, a história descreve efeitos que não se refletem no estado persistente do jogo. Além disso, essa abordagem reduz significativamente a dependência de inferências implícitas feitas por LLMs, aumentando a confiabilidade, previsibilidade e coerência das interações ao longo de campanhas prolongadas.

Essa camada constitui, portanto, um dos pilares do AI Dungeon Master no que diz respeito à consistência mecânica, funcionando como a base sobre a qual narrativas mais complexas e consequências emergentes podem ser construídas de forma segura e controlada.

## 5 Resultado e Discussão

### 5.1 Ambiente de Execução e Fluxo Geral do Sistema

O sistema **AI Dungeon Master** foi implementado como uma aplicação web interativa, composta por uma arquitetura cliente–servidor que separa claramente a interface de usuário (*frontend*) dos componentes responsáveis pelo processamento narrativo, validação de ações e persistência do estado do mundo (*backend*). Essa separação arquitetural permite que o sistema seja utilizado de forma contínua, suportando múltiplas campanhas narrativas independentes, cada uma representando uma instância isolada de um mundo persistente.

No backend, encontram-se os módulos descritos no capítulo de metodologia, responsáveis pela interpretação das entradas do jogador, validação de intenções, simulação mecânica, geração narrativa e aplicação de consequências diretas e indiretas. Já o frontend atua como uma camada de interação, possibilitando ao jogador criar, gerenciar e jogar campanhas narrativas por meio de uma interface gráfica acessível, sem exposição direta à complexidade interna do sistema.

A Figura 5 apresenta a tela de dashboard de campanhas, na qual são listadas todas as campanhas criadas pelo jogador. Cada campanha corresponde a um mundo narrativo independente, com seu próprio estado persistente, histórico de ações, personagens e progressão temporal. Essa funcionalidade evidencia que o sistema não se limita a uma única sessão narrativa efêmera, mas oferece suporte explícito à continuidade, reutilização e retomada de mundos narrativos ao longo do tempo. Do ponto de vista experimental, essa característica é relevante, pois permite a análise de comportamentos emergentes e consistência narrativa em execuções prolongadas.

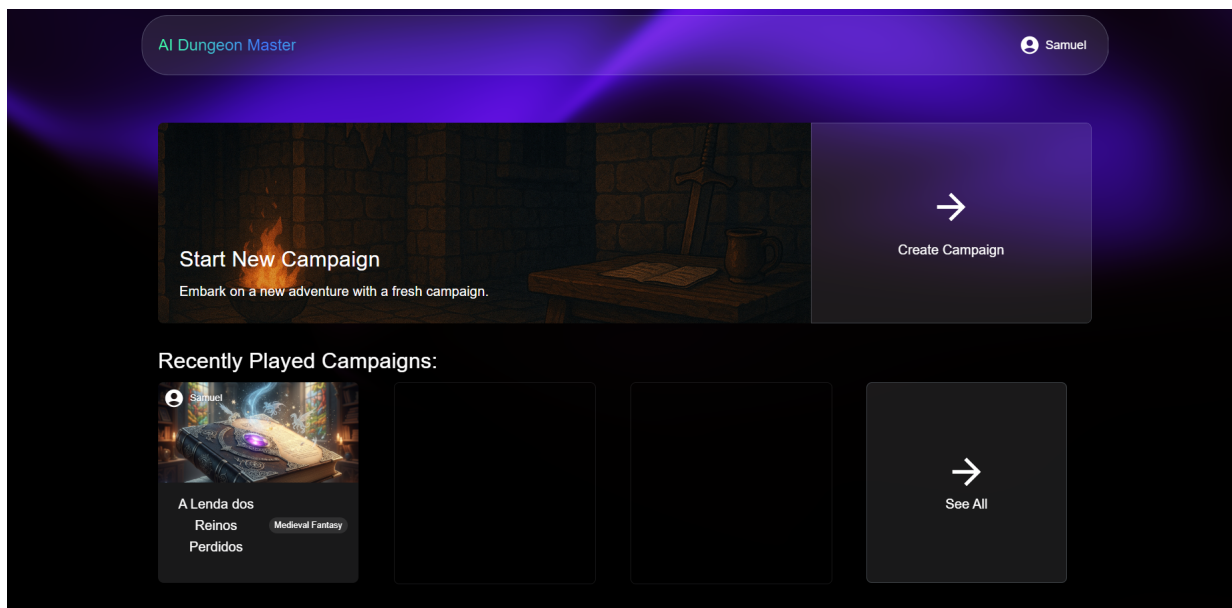


Figura 5 – Tela Dashboard de Campanhas

A Figura 6 ilustra a tela de criação de uma nova campanha, na qual são definidos os parâmetros iniciais do mundo narrativo, como ambientação, contexto inicial e informações básicas do personagem jogador. Esse processo corresponde à fase de inicialização do estado do mundo, na qual estruturas persistentes são criadas e armazenadas antes do início da interação narrativa propriamente dita. Tal abordagem reforça a separação entre configuração do mundo e execução da narrativa.

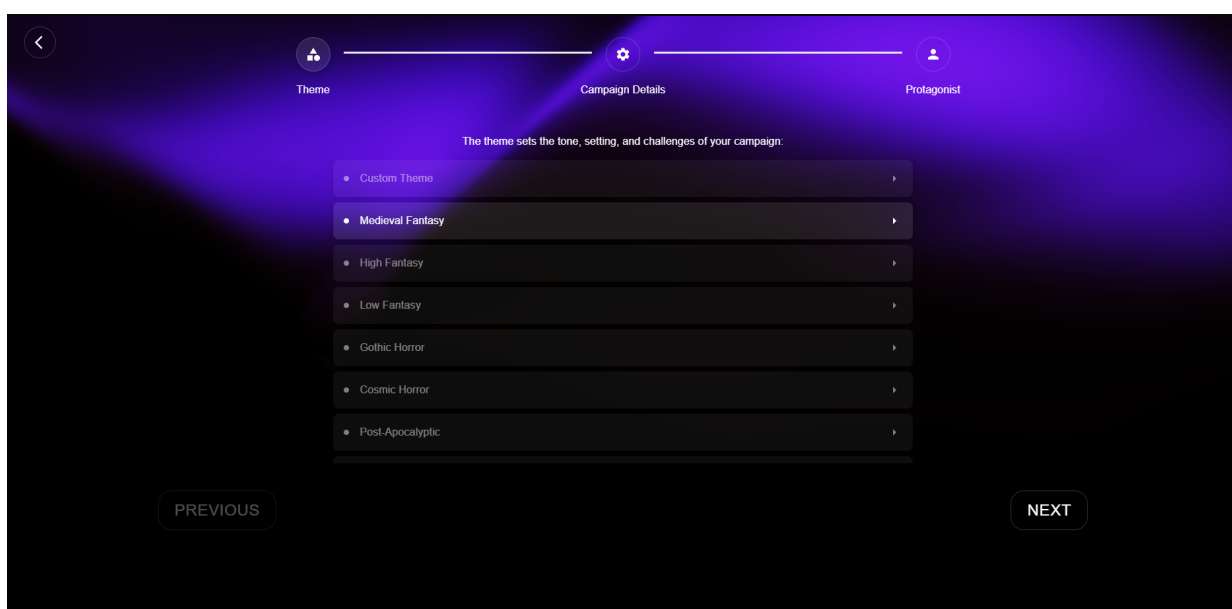


Figura 6 – Tela Criação de Campanha

Após a criação ou seleção de uma campanha, o jogador é direcionado para a interface principal de jogo, apresentada na Figura 7, onde ocorre a interação narrativa

em linguagem natural. Nessa etapa, o jogador envia ações textuais ao sistema, dentro do escopo de intenções macro pré-definidas **DO**, **SAY** e **SEE**, que são processadas de acordo com o fluxo descrito na metodologia, culminando na geração de respostas narrativas coerentes com o estado atual do mundo. Além da área principal de interação, essa interface apresenta, no canto esquerdo da tela, uma representação textual estruturada do estado do jogo, organizada em três níveis: *WorldState*, *SceneState* e *PlayerState*. Essa visualização permite ao jogador acompanhar, de forma explícita, informações globais do mundo, o estado imediato da cena atual e os atributos do personagem, incluindo o acesso ao inventário. Embora essa interface represente o ponto de contato mais visível com o usuário, sua função permanece estritamente a de mediação da interação, enquanto as decisões narrativas, validações e atualizações de estado continuam encapsuladas nos módulos internos do sistema.

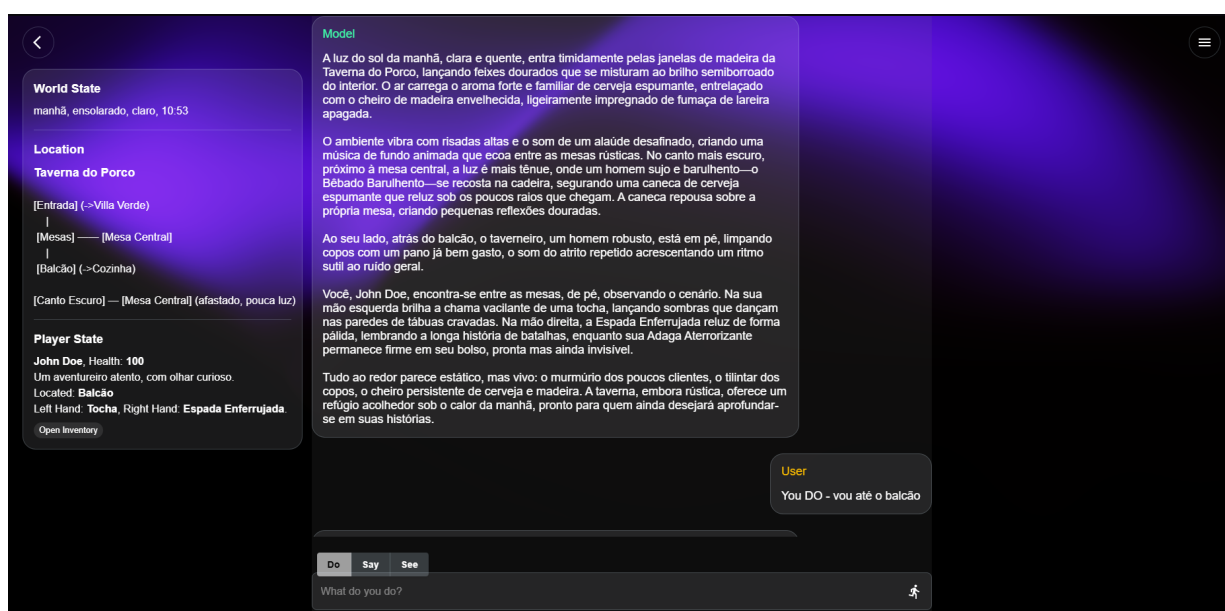


Figura 7 – Tela Campanha

Em conjunto, essas telas demonstram que o AI Dungeon Master não se restringe a um protótipo conceitual ou a uma simulação isolada, mas configura-se como um sistema operacional completo, capaz de gerenciar múltiplos mundos narrativos persistentes, inicializar contextos controlados e sustentar interações narrativas contínuas. Essa infraestrutura é fundamental para a análise dos resultados apresentados nas seções subsequentes, nas quais são discutidos aspectos como consistência de estado, controle narrativo e geração de consequências ao longo da execução das campanhas.

## 5.2 Demonstração dos Intents e suas Reações

Esta seção apresenta uma demonstração prática do funcionamento do sistema por meio da execução dos diferentes tipos de intents suportados. Cada intent repre-

sentam uma classe distinta de ação do jogador, com regras próprias de validação, impacto narrativo e efeitos sobre o estado do mundo. Os exemplos a seguir têm como objetivo evidenciar como o pipeline proposto interpreta entradas em linguagem natural, aplica validações quando necessárias e produz reações coerentes, tanto narrativas quanto mecânicas, preservando a consistência do *GameState*. Ressalta-se que o foco desta seção não está na qualidade literária da narração, mas na observação do comportamento sistêmico e da separação entre lógica, estado e narrativa.

### 5.2.1 Intent SEE — Observação do Ambiente

O *intent SEE* representa ações de observação e percepção do jogador sobre o ambiente, sem implicar intervenção direta no mundo de jogo. Diferentemente de ações que produzem efeitos mecânicos, o SEE não altera o estado do mundo nem avança o tempo narrativo, sendo utilizado exclusivamente para a obtenção de informações contextuais sobre a cena atual. Essa distinção permite que o sistema trate percepção e ação como categorias semânticas distintas, preservando a consistência do *GameState* e evitando a introdução de consequências artificiais em interações puramente observacionais.

Listing 5.1 – Demonstração da preservação do *GameState* após *Intent SEE*.

```
[Antes do SEE]
-----
World: manhã, ensolarado, claro, 10:53

Location: Taverna do Porco
  Description: Uma taverna rústica, com cheiro de cerveja e madeira
              velha.
  Ambience: animado com risadas e música
  Light: semiborroadado
  Weather: quente
  Connected Locations: Villa Verde, Cozinha
  Location Map (low-fidelity):
  (...)
-----
[Depois do SEE]
-----
World: manhã, ensolarado, claro, 10:53

Location: Taverna do Porco
  Description: Uma taverna rústica, com cheiro de cerveja e madeira
              velha.
```

```
Ambiente: animado com risadas e música
Light: semiborreado
Weather: quente
Connected Locations: Villa Verde, Cozinha
Location Map (low-fidelity):
(...)
```

Fonte: Elaborado pelo autor (2026).

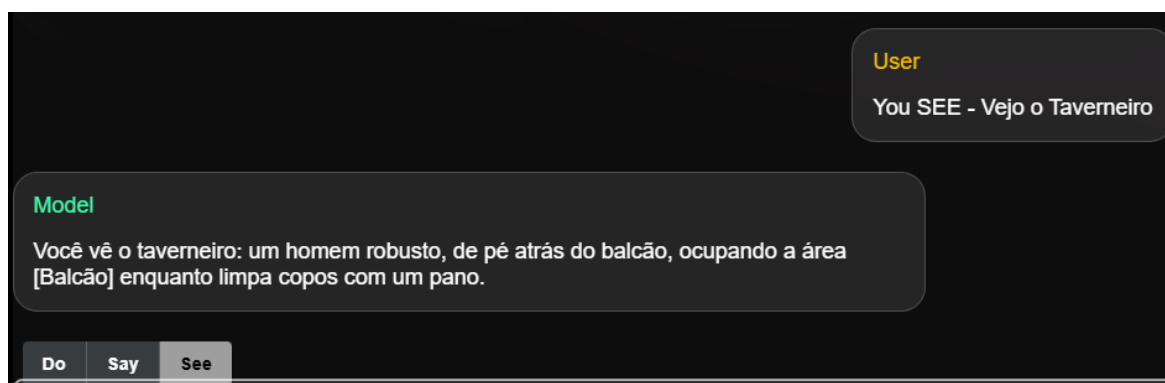


Figura 8 – Interação SEE

A demonstração do intent **SEE** evidencia que **ações de observação não produzem alterações no *GameState*** (Listing 5.1). A comparação entre o estado do mundo antes e após a execução confirma a ausência de modificações em localização, tempo ou atributos das entidades, preservando integralmente a consistência mecânica do sistema. O retorno ao jogador ocorre exclusivamente por meio da resposta narrativa apresentada na interface (Figura 8), reforçando a separação entre percepção e ação efetiva e validando o tratamento do SEE como uma **interação puramente informacional**.

### 5.2.2 Intent SAY — Interação Social

O intent **SAY** representa ações comunicativas realizadas pelo jogador, como falar, questionar ou dialogar com entidades presentes no ambiente. Diferentemente de ações mecânicas diretas, o SAY atua principalmente no plano narrativo e social, podendo envolver testes de habilidade, como carisma, e recuperação de informações relevantes por meio de mecanismos de RAG. Sua execução não implica, por padrão, alterações estruturais imediatas no *GameState*, mas pode desencadear consequências indiretas, como mudanças de atitude de NPCs ou a introdução de novos ganchos narrativos.

## Listing 5.2 – Demonstração da passagem do tempo após Intent SAY.

```
[Antes do SAY]
```

```
-----  
World: manhã, ensolarado, claro, 10:53
```

```
(...)
```

```
-----  
[Depois do SAY]
```

```
-----  
World: manhã, ensolarado, claro, 10:54
```

```
(...)
```

Fonte: Elaborado pelo autor (2026).

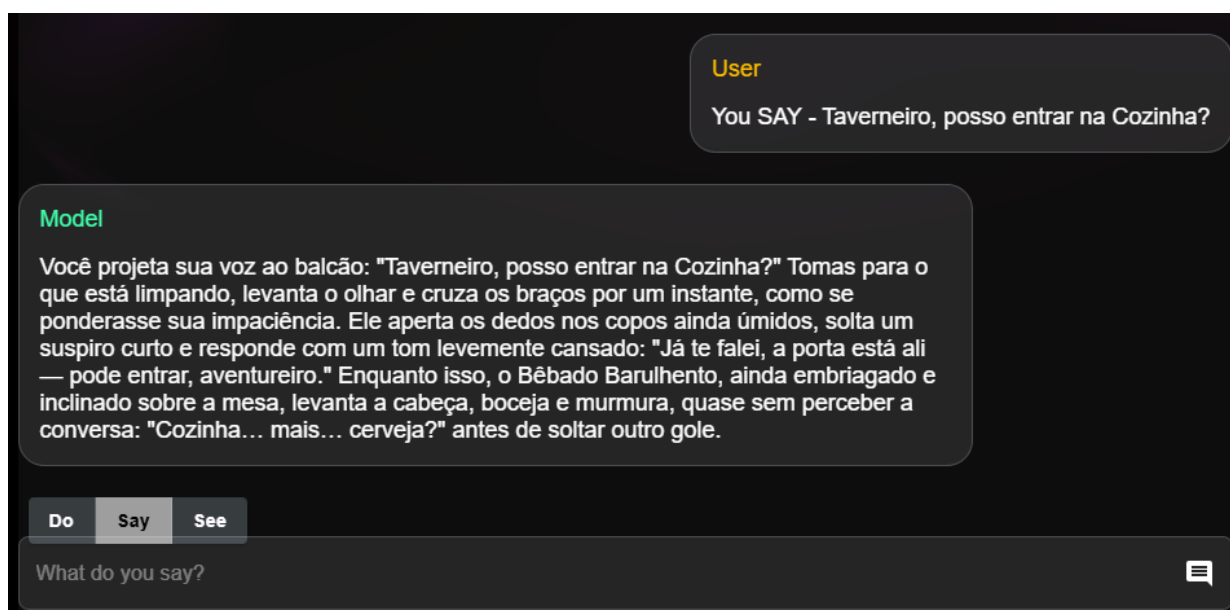


Figura 9 – Interação SAY

A demonstração do *intent SAY* evidencia que, embora não haja modificações estruturais imediatas no estado físico do mundo, sua execução resulta na progressão temporal do sistema (Listing 5.2) e na geração de respostas narrativas coerentes com o contexto social da cena (Figura 9). Observa-se que a reação verbal dos personagens presentes é construída a partir do estado atual e das informações relevantes recuperadas, preservando a consistência do *GameState* ao mesmo tempo em que permite a evolução narrativa e relacional entre as entidades envolvidas.

### 5.2.3 Intent MOVE — Alteração Espacial

O *intent* **MOVE** é responsável por representar ações de deslocamento do jogador no mundo do jogo. Diferentemente de intents puramente narrativos, sua execução implica alterações mecânicas diretas no *GameState*, incluindo a atualização da localização do personagem e o avanço do tempo do mundo. Apesar dessas modificações estruturais, a decisão sobre a permissibilidade da movimentação em termos narrativos é tratada de forma independente, garantindo a separação entre validação mecânica e coerência narrativa.

Listing 5.3 – Demonstração da alteração do GameState após Intent MOVE .

```
[Antes do MOVE]
-----
(...)
Player: John Doe, Health: 100
  Description: Um aventureiro atento, com olhar curioso.
  Area: [Balcão] <-----
  Inventory: Poção de Cura, Adaga Aterrorizante
  Equipped: Espada Enferrujada (slot: RIGHT_HAND)
  Holding: Tocha (slot: LEFT_HAND)
-----

[Depois do MOVE]
-----
(...)
Player: John Doe, Health: 100
  Description: Um aventureiro atento, com olhar curioso.
  Area: [Mesas] <-----
  Inventory: Poção de Cura, Adaga Aterrorizante
  Equipped: Espada Enferrujada (slot: RIGHT_HAND)
  Holding: Tocha (slot: LEFT_HAND)
```

Fonte: Elaborado pelo autor (2026).

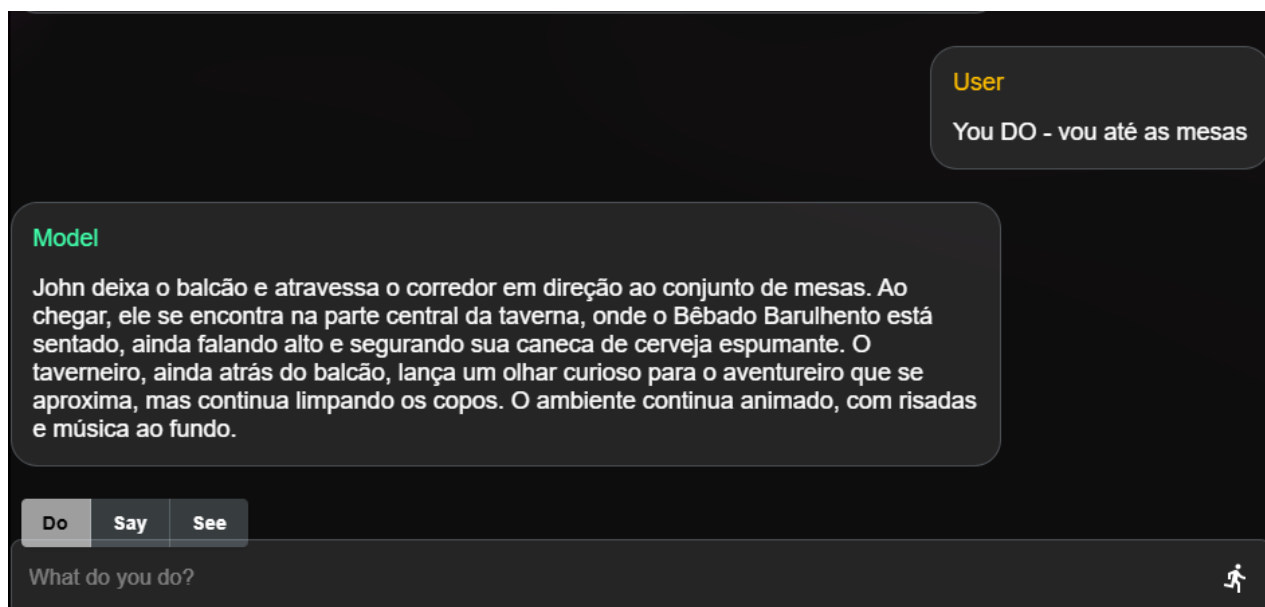


Figura 10 – Interação MOVE

A demonstração do *intent* **MOVE** evidencia a atualização consistente do *GameState* após a execução da ação, em especial no que se refere à mudança de localização do personagem do jogador. Observa-se que, após a validação mecânica e narrativa, o novo estado passa a refletir corretamente o ambiente no qual o jogador se encontra (Listing 5.3), mantendo coerência entre a representação interna do mundo e a resposta narrativa apresentada na interface (Figura 10).

#### 5.2.4 Intent ATTACK — Conflito e Consequências Mecânicas

O *intent* **ATTACK** representa ações ofensivas direcionadas a entidades específicas do mundo de jogo, envolvendo a aplicação de regras mecânicas relacionadas a combate, como verificação de alvos válidos, cálculo de dano e atualização de atributos de estado, especialmente pontos de vida. Diferentemente de intenções puramente narrativas ou observacionais, esse *intent* exige uma integração direta entre validação mecânica e resposta narrativa.

##### Listing 5.4 – Demonstração da alteração do GameState após Intent ATTACK .

```
[Estado do Node do Personagem Taverneiro antes do ATTACK]
-----
<id>: 4:0ae298d8-0254-4d8c-bf59-5071454e7d48:13
agility: 0
charisma: 0
description: "Um homem robusto limpando copos atrás do balcão."
health: 80 <----
intelligence: 0
```

```
name: "Taverneiro"  
perception: 0  
role: "NPC"  
strength: 0  
willpower: 0  
-----  
[Estado do Node do Personagem Taverneiro depois do ATTACK]  
-----  
<id>: 4:0ae298d8-0254-4d8c-bf59-5071454e7d48:13  
agility: 0  
charisma: 0  
description: "Um homem robusto limpando copos atrás do balcão."  
health: 65 <----  
intelligence: 0  
name: "Taverneiro"  
perception: 0  
role: "NPC"  
strength: 0  
willpower: 0
```

Fonte: Elaborado pelo autor (2026).

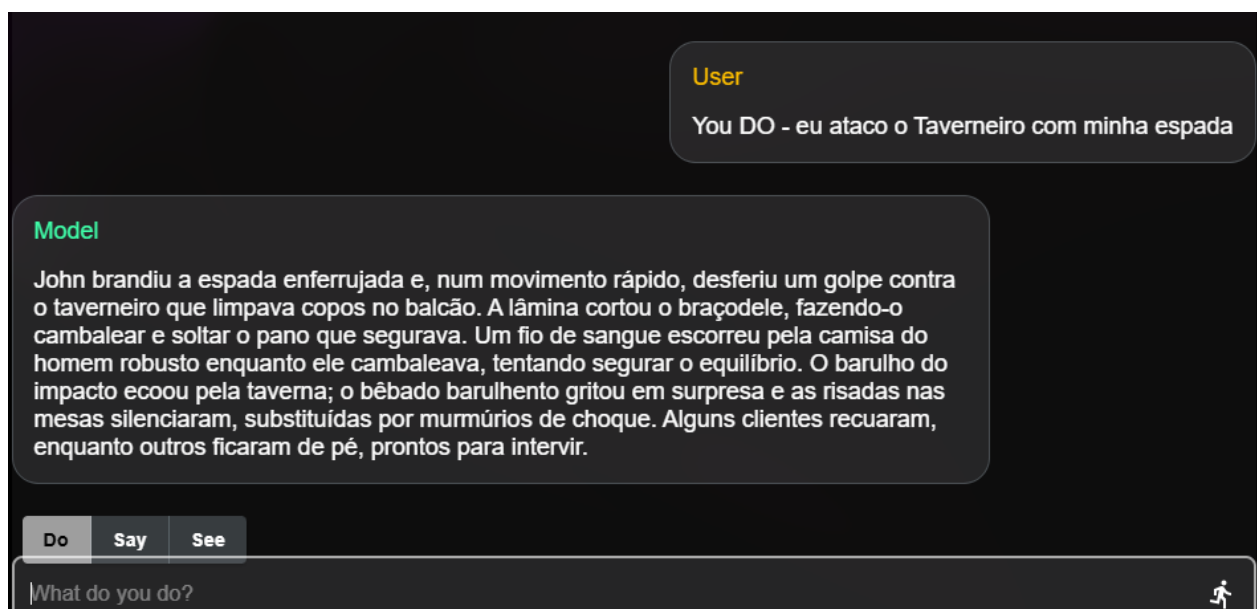


Figura 11 – Interação ATTACK

A demonstração do *intent* **ATTACK** evidencia a correta propagação dos efeitos de uma ação ofensiva sobre o estado do mundo. A partir da mensagem do jogador que dispara a intenção de ataque contra o personagem Taverneiro, observa-se a atualização explícita do atributo de vida no nó correspondente do grafo que representa

o mundo de jogo (Listing 5.4) , refletindo a consequência mecânica da ação. Paralelamente, a resposta narrativa retornada ao frontend descreve o impacto do ataque no contexto da cena (Figura 11).

### 5.2.5 Intent USEITEM — Manipulação de Inventário e Estado

O *intent USEITEM* representa ações do jogador relacionadas à manipulação direta de itens, incluindo uso, equipagem, armazenamento ou descarte, produzindo efeitos concretos sobre o estado do personagem e do mundo. Diferentemente de intenções puramente narrativas, esse tipo de ação está diretamente associado a alterações mecânicas bem definidas, como mudanças em atributos, inventário ou equipamentos ativos. A seguir, é apresentada a demonstração do *intent USEITEM*, evidenciando como o sistema valida a ação, atualiza o estado persistente e, posteriormente, traduz essas mudanças em uma resposta narrativa coerente ao jogador.

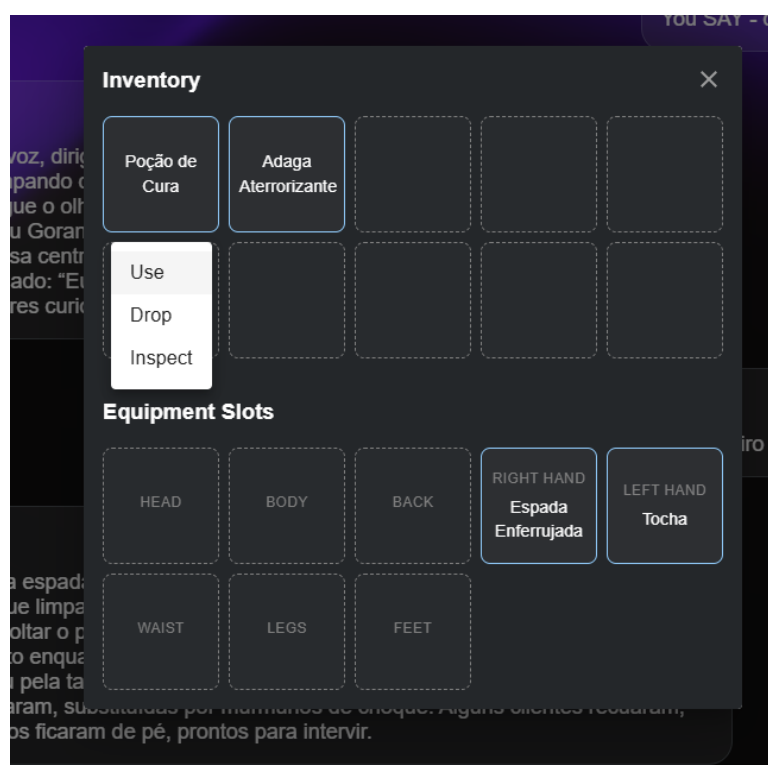


Figura 12 – Uso da Poção de Cura

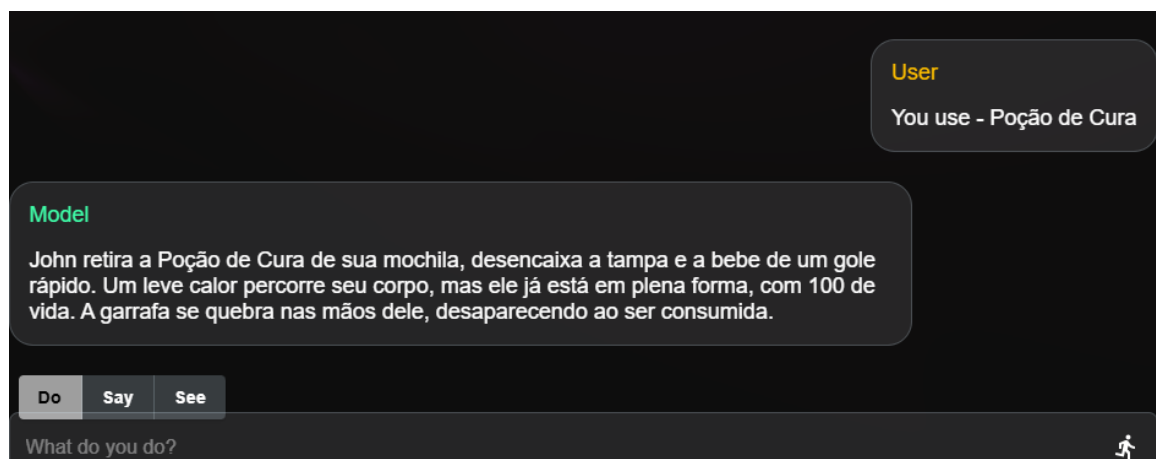


Figura 13 – Interação USEITEM

Listing 5.5 – Demonstração da alteração do GameState após Intent USEITEM .

```
[Antes do USEITEM]
-----
(...)
Player: John Doe, Health: 80 <-----
  Description: Um aventureiro atento, com olhar curioso.
  Area: [Balcão]
  Inventory: Poção de Cura, Adaga Aterrorizante
  Equipped: Espada Enferrujada (slot: RIGHT_HAND)
  Holding: Tocha (slot: LEFT_HAND)
-----

[Depois do USEITEM]
-----
(...)
Player: John Doe, Health: 100 <-----
  Description: Um aventureiro atento, com olhar curioso.
  Area: [Mesas]
  Inventory: Poção de Cura, Adaga Aterrorizante
  Equipped: Espada Enferrujada (slot: RIGHT_HAND)
  Holding: Tocha (slot: LEFT_HAND)
```

Fonte: Elaborado pelo autor (2026).

A demonstração do *intent* **USEITEM** evidencia a integração entre validação mecânica, atualização do estado persistente e geração narrativa. No exemplo apresentado, a poção é consumida com sucesso (Figura 12), resultando na aplicação imediata de seus efeitos ao personagem e na remoção do item do inventário. Essa alteração no estado do sistema pode ser observada detalhadamente no Listing 5.5, que ilustra a atualização do *GameState* no momento em que a poção de cura incrementa os pontos de

vida do jogador. Adicionalmente, conforme ilustrado na Figura 12, a interface permite ao usuário selecionar ações específicas para cada item, como *Use*, *Drop* ou *Inspect*, garantindo versatilidade na gestão do inventário. Em seguida, o sistema produz uma descrição narrativa compatível com a ação realizada (Figura 13), reforçando a relação de causalidade entre mecânica e narrativa.

### 5.2.6 Intent NARRATIVE — Ação Livre e Flexível

O *intent NARRATIVE* representa uma categoria mais aberta e flexível de interação, utilizada quando a ação do jogador não se enquadra de forma clara em intents mecânicos específicos, como *MOVE*, *ATTACK* ou *USEITEM*. Nesse caso, a entrada do jogador é interpretada prioritariamente em seu valor narrativo, permitindo descrições, atitudes ou comportamentos que enriquecem a cena sem produzir, de forma imediata, alterações mecânicas diretas no estado do mundo. Esse intent atua, portanto, como um mecanismo de expressão narrativa livre, preservando a agência do jogador e ampliando a expressividade da interação, ao mesmo tempo em que mantém a coerência com o contexto e os limites estabelecidos pelo sistema.

#### Listing 5.6 – Intent NARRATIVE classificado pelo ActionResolverAgent

```
Validated Intent :
  type=<PlayerIntentType.NARRATIVE: 'NARRATIVE'>
  confidence=0.96
  raw_text='olho de maneira ameaçadora para todos na Taverna do
    Porco' decisor='LLM_INFERRED'
  narrative_act=<NarrativeActType.INTIMIDATION: 'intimidation'>
  target='todos na Taverna do Porco'
  inferred_tone='agressivo'
  inferred_intent='intimidation'
  emotion='aggressiveness'
```

Fonte: Elaborado pelo autor (2026).

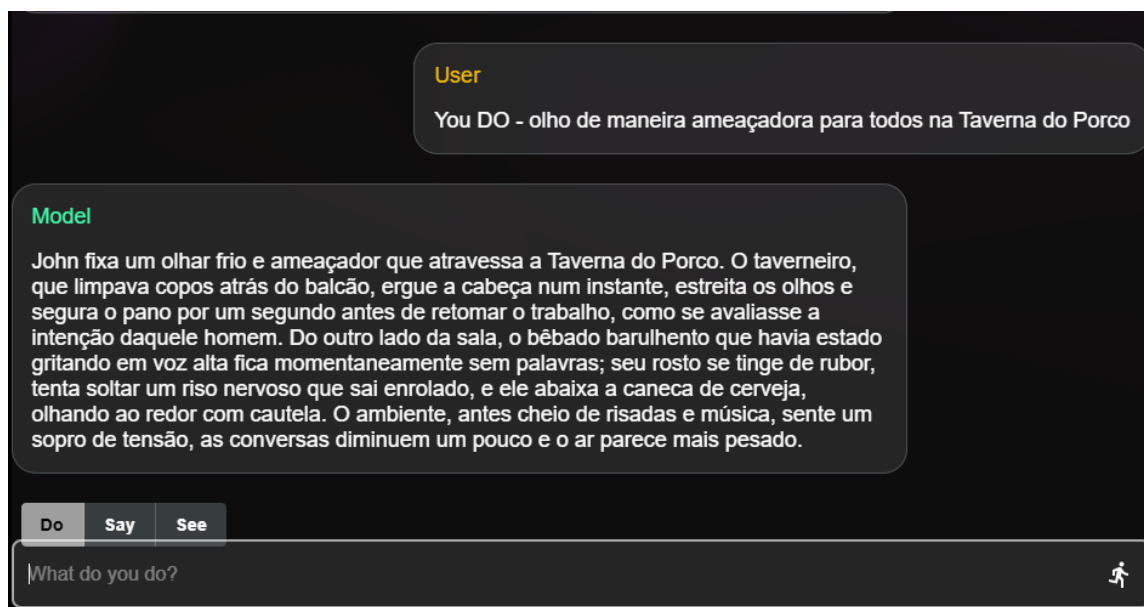


Figura 14 – Interação NARRATIVE

Listing 5.7 – Lista de Consequências Indiretas retornadas pelo NarrativeAgent para essa Ação

```
[
  {
    "type": "NARRATIVE_HOOK",
    "target": "Taverneiro",
    "details": "Visivelmente surpreendido, pausa a limpeza por um
      instante, estreitando os olhos antes de continuar."
  },
  {
    "type": "NARRATIVE_HOOK",
    "target": "Bêbado Barulhento",
    "details": "Fica ruborizado, tenta rir mas a voz falha, abaixa
      a caneca e observa cautelosamente."
  }
]
```

Fonte: Elaborado pelo autor (2026).

A demonstração do *intent* **NARRATIVE** evidencia o papel desse mecanismo como um elemento de fechamento conceitual do conjunto de intents suportados pelo sistema. Conforme apresentado, a entrada do jogador foi corretamente classificada pelo *ActionResolverAgent* como narrativa (Listing 5.5), resultando em uma resposta textual coerente com o contexto da cena (Figura 14) e na geração de um conjunto explícito de consequências indiretas (Listing 5.6), representadas em um array estrutu-

rado. Embora essas consequências não impliquem, necessariamente, alterações mecânicas imediatas no estado do mundo, elas estabelecem ganchos narrativos e sinais claros para possíveis reações futuras do sistema. Com isso, esta seção consolida a proposta geral de interpretação de intenções do *AI Dungeon Master*, demonstrando como diferentes tipos de interação, desde ações estritamente mecânicas até expressões narrativas livres, são tratados de forma consistente, controlada e extensível, reforçando a separação entre lógica, narrativa e evolução do mundo apresentada ao longo deste trabalho.

### 5.3 Modelos de Linguagem Utilizados e seus Papéis no Sistema

Durante o desenvolvimento e a execução do sistema *AI Dungeon Master*, foram empregados diferentes modelos de linguagem de grande porte (*Large Language Models* - LLMs), cada um associado a responsabilidades específicas dentro da arquitetura proposta. Conforme detalhado na Tabela 4, essa escolha deliberada de múltiplos modelos segue o princípio de adequação entre a complexidade da tarefa e a capacidade do modelo, buscando o equilíbrio entre qualidade narrativa, custo computacional e desempenho geral do sistema. Dessa forma, a atribuição de cada modelo às suas respectivas funcionalidades baseou-se em uma avaliação qualitativa autoral, selecionando as ferramentas que, segundo minha percepção técnica, apresentaram o comportamento mais refinado e aderente às necessidades de cada componente.

O ambiente de execução baseia-se no *Ollama* como infraestrutura para uso local de LLMs, permitindo a execução de modelos leves de forma eficiente e com baixo custo operacional. Modelos locais são priorizados para tarefas de classificação simples e processamento semântico restrito, enquanto modelos de maior porte, executados em ambiente *cloud*, são reservados para tarefas que demandam maior capacidade de compreensão contextual e geração narrativa sofisticada.

O *NarrativeAgent*, responsável exclusivamente pela geração de descrições narrativas e consequências textuais, utiliza o modelo `gpt-oss:120b-cloud`, em razão de sua maior capacidade de coerência discursiva, consistência estilística e interpretação de contextos complexos. Esse agente atua estritamente na camada narrativa, não participando de decisões mecânicas ou validações de regras de jogo.

Por sua vez, o *ActionResolverAgent*, encarregado da classificação das entradas do jogador em *intents* estruturados, adota uma estratégia híbrida. Para entradas diretas e linguisticamente simples, é utilizado o modelo `gemma3:4b`, executado localmente via *Ollama*, visando latência reduzida. Em cenários de ambiguidade semântica, onde a classificação inicial não atinge o limiar de confiança estabelecido, caracterizando o *fallback* semântico, o sistema recorre ao modelo de maior escala (`gpt-oss:120b-cloud`)

para uma inferência aprofundada, permitindo a identificação precisa de intenções complexas, inclusive do tipo *NARRATIVE*.

Essa distribuição de responsabilidades entre modelos distintos reforça a separação entre lógica, interpretação semântica e narrativa, evidenciando uma abordagem pragmática e escalável para a implementação de LLMs em sistemas interativos de RPG.

Tabela 4 – Modelos de Linguagem Utilizados no AI Dungeon Master

Agente / Componente	Modelo	Ambiente	Papel no Sistema
ActionResolverAgent	gemma3:4b	Local	Classificação semântica de intents simples (SEE, MOVE, SAY, ATTACK, USE).
ActionResolverAgent (fallback)	gpt-oss:120b	Cloud	Classificação avançada de intents ambíguos e inferência do intent <i>NARRATIVE</i> .
NarrativeAgent	gpt-oss:120b	Cloud	Geração narrativa, descrição de consequências e bloqueios narrativos.

Fonte: Elaborada pelo autor (2026).

Tabela 5 – Análise comparativa de performance e viabilidade entre os modelos.

Modelo	Ambiente	Vantagens e Performance	Limitações e Viabilidade
gpt-oss:120b	Cloud (Ollama)	Alto desempenho em tarefas complexas; retorno rápido; narrativas de alta qualidade.	Recurso pago; cota limitada de chamadas por hora; dependência de conectividade externa.
gemma3:4b	Local (Ollama)	Chamadas ilimitadas; custo operacional zero; total privacidade dos dados.	Tempo de resposta elevado para a escala do modelo; baixa qualidade em tarefas de maior complexidade.

Fonte: Elaborada pelo autor (2026).

## 5.4 Discussão

### 5.4.1 Aspectos que funcionaram adequadamente

O principal ponto positivo identificado nesta pesquisa foi **a separação clara de responsabilidades entre os módulos mecânicos e narrativos**. A divisão entre agentes responsáveis pela validação de ações, atualização de estado e geração narrativa mostrou-se eficaz na mitigação de problemas recorrentes em abordagens fundamentadas exclusivamente em LLMs, tais como alucinações factuais e perda de coerência temporal.

Outro aspecto bem-sucedido consistiu na adoção de uma **representação explícita do mundo por meio de grafos**, permitindo que personagens, itens e localizações mantivessem relações persistentes e verificáveis. Essa estrutura facilitou tanto a validação mecânica das ações quanto a geração de narrativas contextualizadas, reduzindo inconsistências críticas, como o deslocamento indevido de personagens (*teleporte*) ou o uso de itens ausentes no inventário do jogador.

A utilização de interfaces estruturadas no *frontend*, incluindo a visualização do estado do mundo, inventário e localização atual, também contribuiu positivamente para a experiência do usuário. Esses elementos reforçaram a percepção de causalidade e aproximaram a interação da dinâmica tradicional de um RPG de mesa. Do ponto de vista arquitetural, essa decisão consolidou a separação entre a camada de apresentação e a lógica de domínio, impedindo que o *frontend* assumisse responsabilidades narrativas ou mecânicas indevidas, garantindo a integridade do fluxo de dados no sistema.

#### 5.4.2 Limitações e aspectos que não funcionaram plenamente

A despeito dos resultados positivos alcançados, foram identificadas limitações pertinentes ao modelo proposto. Primeiramente, destaca-se a complexidade adicional inerente à arquitetura híbrida; a orquestração entre múltiplos agentes, validadores e relatores eleva o custo de desenvolvimento e depuração ao ser confrontada com soluções monolíticas fundamentadas exclusivamente em LLMs. Em particular, a sincronização entre eventos narrativos e as atualizações de estado demanda rigoroso controle para mitigar o risco de inconsistências transitórias durante o fluxo de execução.

Outra limitação observada refere-se à natureza da campanha *toy* utilizada, cujas entidades e relações foram previamente definidas via *scripts*. Embora tal abordagem tenha sido adequada ao escopo deste projeto e ao cronograma disponível, ela evidencia que a geração integralmente autônoma de mundos complexos ainda requer mecanismos suplementares de validação, sob o risco de comprometer a consistência global do universo ficcional.

Ademais, a geração de consequências indiretas mediada por LLMs, embora promissora, revelou-se um ponto de sensibilidade técnica. Em determinados cenários, foi imperativo restringir o formato e o escopo das saídas do modelo para evitar efeitos colaterais incoerentes com o *WorldState*. Essa constatação corrobora a premissa de que os modelos de linguagem devem atuar como geradores condicionados a uma base de dados estruturada, e não como fontes de verdade definitivas do sistema.

### 5.4.3 Pitfalls e Lições Aprendidas

A implementação do *AI Dungeon Master* permitiu a identificação de armadilhas técnicas (*pitfalls*) e o aprendizado de lições fundamentais para a engenharia de sistemas baseados em LLMs. Os principais pontos observados são detalhados a seguir:

- a) **Instabilidade na Geração de Dados Estruturados:** Observou-se que, embora modelos de linguagem sejam capazes de produzir saídas em formatos como JSON, não se pode confiar em uma taxa de sucesso de 100% na validade sintática dessas estruturas. Mesmo com *prompt engineering* rigoroso, o sistema deve prever mecanismos de *parsing* robustos e tratamentos de exceção para lidar com respostas malformadas que poderiam interromper o fluxo do *engine* de jogo.
- b) **A Dependência da Qualidade na Recuperação (RAG):** A eficácia de um sistema de Geração Aumentada de Recuperação (RAG) é diretamente proporcional à relevância das informações recuperadas. O uso irrestrito ou a recuperação de fragmentos de baixa similaridade semântica introduz ruído no contexto. Esse excesso de informações irrelevantes induz a LLM a integrar dados falsos na narrativa, resultando em respostas incoerentes ou desconexas do estado atual da campanha.
- c) **Limitações do Controle por Prompt e Alucinações de Conhecimento:** Verificou-se que diretivas negativas nos *prompts* (ex: "Use SOMENTE as informações fornecidas") possuem eficácia limitada diante da natureza probabilística dos modelos. Na ausência de dados detalhados sobre entidades específicas na campanha *toy*, como o histórico profundo de um NPC, o *NarrativeAgent* tende a preencher lacunas informativas com conhecimentos genéricos de seu pré-treinamento. Essa tendência à "alucinação de conhecimento" reforça que o controle puramente textual é insuficiente para garantir a estrita fidelidade a um universo fechado, exigindo uma base de dados mais densa para minimizar a inventividade do modelo.
- d) **Gargalos de Latência em Arquiteturas Multi-Agentes:** A decomposição do sistema em múltiplos agentes (como o *ActionResolver* e o *NarrativeAgent*) introduz uma latência cumulativa. Cada chamada de API ou processamento local via *Ollama* soma-se ao tempo total de resposta. Em um ambiente de jogo, essa demora pode quebrar a imersão do usuário (*flow*), revelando um conflito entre a complexidade da validação mecânica e a necessidade de interatividade em tempo real.
- e) **A Deriva de Estilo e Tom Narrativo:** Manter a consistência na "voz" do mestre de jogo é um desafio quando se alterna entre diferentes modelos (como o *Gemma* e o *GPT-OSS*). Observou-se que modelos de menor escala tendem a ser mais

diretos e técnicos, enquanto modelos maiores são mais descritivos. Essa oscilação estilística pode gerar uma experiência fragmentada, onde a narração de um evento simples parece desconexa da descrição de um evento complexo, exigindo um esforço adicional na padronização dos *system prompts*.

- f) **A "Cegueira de Contexto" por Janela de Memória:** Apesar do uso de grafos para persistência, a janela de contexto da LLM ainda é limitada. Se o sistema enviar um histórico muito longo para garantir a continuidade, o modelo pode sofrer do fenômeno de *Lost in the Middle* (perda de atenção em informações localizadas no meio do contexto enviado). Isso exige uma estratégia de sumarização ou seleção criteriosa de fatos, pois fornecer "toda a memória" do grafo de uma vez pode, paradoxalmente, levar o agente a ignorar detalhes cruciais da cena atual.

#### 5.4.4 Síntese da discussão

Em síntese, os resultados obtidos e a análise crítica subsequente indicam que a arquitetura proposta para o *AI Dungeon Master* representa uma evolução incremental frente aos sistemas narrativos fundamentados exclusivamente em modelos de linguagem de grande porte. Ao consolidar o papel da LLM como um componente especializado de síntese textual e interpretação semântica, em vez de utilizá-la como o núcleo lógico do sistema, obteve-se um maior determinismo factual, previsibilidade operacional e aderência às mecânicas de jogo preestabelecidas.

Apesar de a arquitetura híbrida introduzir desafios relacionados à complexidade de orquestração e à escalabilidade na geração autônoma de conteúdo, as lições aprendidas ratificam a validade da proposta. O desacoplamento entre a lógica de estado (baseada em grafos) e a camada narrativa (baseada em agentes) demonstra ser um caminho promissor para mitigar as limitações intrínsecas das LLMs. Em última análise, este trabalho estabelece uma fundamentação sólida para futuras investigações que visem o equilíbrio entre a flexibilidade da linguagem natural e o rigor dos sistemas baseados em regras.

## 6 Trabalhos Futuros

A natureza modular e extensível do *AI Dungeon Master* abre precedentes para diversas frentes de pesquisa e desenvolvimento. Com o intuito de superar as limitações identificadas e expandir as fronteiras da narrativa emergente assistida por IA, propõem-se os seguintes trabalhos futuros:

- a) **Geração Autônoma e Procedural de Universos:** Uma evolução natural deste trabalho consiste na criação de sistemas capazes de gerar campanhas completas sem intervenção humana prévia. A partir de parâmetros mínimos, como temática (*setting*), descrição sucinta do mundo e arquétipo do personagem, o sistema utilizaria LLMs de alta escala para instanciar, de forma autônoma, o grafo de entidades inicial. Isso incluiria a definição de *lore*, estados de conhecimento de NPCs, topologia de localizações e eventos históricos persistentes, garantindo que o mundo gerado possua uma profundidade técnica maior, ou equivalente, a campanha *toy* apresentada neste estudo.
- b) **Refinamento de Causalidade e Consequências Indiretas:** Propõe-se o desenvolvimento de um módulo especializado, operando de forma análoga ao *Action-ResolverAgent*, voltado para a análise de impacto sistêmico. Este agente seria responsável por avaliar a interação entre a ação do jogador e a resposta narrativa, identificando consequências secundárias ou indiretas que não foram previstas pela regra imediata. O resultado dessa inferência seria traduzido em atualizações automáticas no *GameState*, permitindo que o estado do mundo reflita de forma mais rica e complexa as escolhas do usuário, tornando o mundo mais "vivo" e reativo.
- c) **Colaboração Híbrida Humano-IA (Coautoria):** Uma vertente promissora é a inclusão de um Mestre de Jogo (*Dungeon Master* - DM) humano no ciclo de controle. Nesta configuração, o sistema atuaria como um copiloto narrativo, permitindo que o DM intervenha em pontos cruciais da trama ou forneça direcionamentos de alto nível para os eventos da campanha. A LLM, por sua vez, encarregaria-se da micro-narrativa e da manutenção das regras estruturadas, transformando o sistema em uma ferramenta de coautoria em tempo real que potencializa a criatividade humana através da automação algorítmica.
- d) **Exploração de Modelos de Voz e Multimodalidade:** Adicionalmente, a integração de tecnologias de *Text-to-Speech* (TTS) com modelos de voz expressivos permitiria que a narração e os diálogos dos NPCs fossem apresentados de forma

audível, aumentando a imersão sensorial. Aliado a isso, o uso de modelos de difusão para a geração dinâmica de artes visuais dos cenários e personagens, baseadas no estado atual do grafo, consolidaria o *AI Dungeon Master* como uma plataforma multimídia completa de RPG.

## 7 Conclusão

Este trabalho apresentou o *AI Dungeon Master*, um sistema arquitetural para jogos de RPG narrativos assistidos por *Large Language Models* (LLMs), concebido para equilibrar a criatividade da geração de linguagem natural com a necessidade de consistência lógica, aderência a regras e previsibilidade das consequências das ações do jogador. A proposta partiu da constatação de que abordagens baseadas exclusivamente em LLMs, embora expressivas do ponto de vista narrativo, tendem a apresentar instabilidades factuais, violações de causalidade e perda gradual de coerência em interações prolongadas.

A arquitetura desenvolvida fundamenta-se na separação explícita de responsabilidades entre os componentes do sistema, distinguindo claramente a interpretação semântica das ações do jogador, a validação mecânica dessas ações, a atualização do estado do mundo e a geração narrativa. Essa organização mostrou-se essencial para evitar que decisões críticas sobre o funcionamento do jogo fossem delegadas diretamente aos modelos de linguagem, reduzindo a ocorrência de comportamentos imprevisíveis e inconsistentes.

A adoção de um fluxo de interação baseado em *intents* permitiu estruturar ações em linguagem natural de forma controlada, sem comprometer a expressividade do jogador. A demonstração prática dos diferentes tipos de *intents* evidenciou que o sistema é capaz de lidar com interações variadas, como observação, diálogo, movimentação, combate e uso de itens, mantendo a coerência entre entrada do usuário, validações internas e resposta narrativa. Essa abordagem contribuiu para tornar o comportamento do sistema mais transparente e alinhado às expectativas típicas de jogos de RPG.

No que diz respeito à persistência do mundo de jogo, a combinação de diferentes estratégias mostrou-se eficaz. A representação do mundo por meio de uma estrutura em grafo permitiu modelar entidades e suas relações de forma explícita, enquanto a utilização de uma base vetorial para armazenamento de *lore* e conhecimento contextual possibilitou a recuperação de informações narrativas relevantes ao longo da campanha. Complementarmente, o histórico temporal das interações forneceu um registro estruturado das ações do jogador e das respostas do sistema, contribuindo para a continuidade narrativa.

Os resultados experimentais e a avaliação qualitativa indicaram que a manutenção de um estado de jogo explícito, validado mecanicamente, reduz significativamente problemas comuns em sistemas puramente baseados em LLMs, como alucinações factuais e mudanças arbitrárias de cenário. A comparação com uma solução existente

evidenciou que a previsibilidade das consequências e a presença de interfaces estruturadas, como mapas e inventários, aproximam a experiência do *AI Dungeon Master* daquela observada em RPGs de mesa tradicionais.

Apesar dos avanços alcançados, o trabalho apresenta limitações inerentes ao seu escopo. A campanha utilizada para demonstração foi construída como um cenário controlado, com entidades previamente definidas por meio de *scripts*, o que restringe a avaliação de geração totalmente autônoma de mundos e narrativas em larga escala. Ainda assim, essa escolha permitiu uma análise mais rigorosa da arquitetura proposta e de seus mecanismos centrais de validação e persistência.

Como perspectivas futuras, destacam-se a ampliação da autonomia sistêmica na geração procedimental de campanhas e a implementação de módulos voltados ao processamento de consequências indiretas, visando elevar o determinismo das interações complexas no *GameState*. Adicionalmente, a investigação de modelos de coautoria híbrida, integrando a figura de um mestre humano ao ciclo de controle da IA, e a realização de testes com grupos de usuários mais amplos e diversificados apresentam-se como etapas fundamentais.

Conclui-se que o *AI Dungeon Master* demonstra que a aplicação de LLMs em jogos narrativos é mais eficaz quando inserida em uma arquitetura que privilegia estado explícito, validação mecânica e persistência estruturada. Ao tratar os modelos de linguagem como componentes especializados, e não como a única fonte de decisão, o sistema contribui para o avanço de plataformas narrativas interativas mais consistentes, confiáveis e adequadas a campanhas de longa duração.

## Referências

- ACHIAM, J. et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023. Citado 2 vezes nas páginas 9 e 20.
- BARNETT, S. et al. Seven failure points when engineering a retrieval augmented generation system. In: *Proceedings of the IEEE/ACM 3rd International Conference on AI Engineering-Software Engineering for AI*. [S.l.: s.n.], 2024. p. 194–199. Citado na página 23.
- BARTLE, R. A. *Designing virtual worlds*. [S.l.]: New Riders, 2004. Citado 3 vezes nas páginas 24, 26 e 27.
- BJÖRK, S.; HOLOPAINEN, J. Games and design patterns. *The game design reader: A rules of play anthology*, MIT Press, Cambridge MA, p. 410–437, 2005. Citado na página 26.
- BOMMASANI, R. et al. *On the Opportunities and Risks of Foundation Models*. 2022. Disponível em: <<https://arxiv.org/abs/2108.07258>>. Citado na página 20.
- BROWN, T. et al. Language models are few-shot learners. *Advances in neural information processing systems*, v. 33, p. 1877–1901, 2020. Citado na página 20.
- CHEN, J. et al. *From Persona to Personalization: A Survey on Role-Playing Language Agents*. 2024. Disponível em: <<https://arxiv.org/abs/2404.18231>>. Citado na página 9.
- ELIAS, G. S.; GARFIELD, R.; GUTSCHERA, K. R. *Characteristics of games*. [S.l.]: MIT Press, 2012. Citado 3 vezes nas páginas 24, 25 e 26.
- FAN, A.; LEWIS, M.; DAUPHIN, Y. Hierarchical neural story generation. *arXiv preprint arXiv:1805.04833*, 2018. Citado 2 vezes nas páginas 20 e 21.
- FINE, G. A. *Shared fantasy: Role playing games as social worlds*. [S.l.]: University of Chicago Press, 2002. Citado 2 vezes nas páginas 23 e 24.
- GAO, Y. et al. Retrieval-augmented generation for large language models: A survey. *arXiv preprint arXiv:2312.10997*, v. 2, n. 1, 2023. Citado na página 22.
- GARNELO, M.; SHANAHAN, M. Reconciling deep learning with symbolic artificial intelligence: representing objects and relations. *Current Opinion in Behavioral Sciences*, Elsevier, v. 29, p. 17–23, 2019. Citado na página 27.
- HIGGINS, J. Janet h. murray, hamlet on the holodeck: The future of narrative in cyberspace (new york: The free press, 1997), 324 pp. isbn 0-684-82723-9. *Convergence*, Sage Publications Sage CA: Thousand Oaks, CA, v. 4, n. 4, p. 128–130, 1998. Citado 2 vezes nas páginas 23 e 24.
- HU, C. et al. Chatdb: Augmenting llms with databases as their symbolic memory. *arXiv preprint arXiv:2306.03901*, 2023. Citado na página 28.

ISBISTER, K. *Better game characters by design: A psychological approach*. [S.l.]: CRC Press, 2022. Citado na página 24.

Ji, Z. et al. Survey of hallucination in natural language generation. *ACM computing surveys*, ACM New York, NY, v. 55, n. 12, p. 1–38, 2023. Citado 3 vezes nas páginas 21, 26 e 27.

JUUL, J. *Half-real: Video games between real rules and fictional worlds*. [S.l.]: MIT press, 2011. Citado na página 25.

Latitude Inc. *AI Dungeon: Interactive AI-Generated Storytelling Game*. 2019. Software online. Acesso em: 10 jan. 2026. Disponível em: <<https://aidungeon.com>>. Citado na página 15.

LEANDRO, J. et al. Geneva: Generating and visualizing branching narratives using llms. In: IEEE. *2024 IEEE Conference on Games (CoG)*. [S.l.], 2024. p. 1–5. Citado na página 9.

LEWIS, P. et al. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems*, v. 33, p. 9459–9474, 2020. Citado na página 22.

MONTFORT, N. *Twisty little passages: An approach to interactive fiction*. [S.l.]: Mit Press, 2005. Citado 2 vezes nas páginas 23 e 24.

MU, N. et al. Can llms follow simple rules? *arXiv preprint arXiv:2311.04235*, 2023. Citado 3 vezes nas páginas 9, 11 e 21.

PARK, J. S. et al. Generative agents: Interactive simulacra of human behavior. In: *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*. New York, NY, USA: Association for Computing Machinery, 2023. (UIST '23). ISBN 9798400701320. Disponível em: <<https://doi.org/10.1145/3586183.3606763>>. Citado 6 vezes nas páginas 9, 21, 22, 25, 26 e 28.

PENG, L.; SHANG, J. Codifying character logic in role-playing. *arXiv preprint arXiv:2505.07705*, 2025. Citado 3 vezes nas páginas 9, 11 e 18.

RIEDL, M. O.; BULITKO, V. Interactive narrative: An intelligent systems approach. *Ai Magazine*, v. 34, n. 1, p. 67–67, 2013. Citado 6 vezes nas páginas 21, 22, 23, 24, 25 e 28.

SURÍS, D.; MENON, S.; VONDRICK, C. Vipergpt: Visual inference via python execution for reasoning. In: *Proceedings of the IEEE/CVF international conference on computer vision*. [S.l.: s.n.], 2023. p. 11888–11898. Citado na página 28.

SWANSON, R.; GORDON, A. S. Say anything: Using textual case-based reasoning to enable open-domain interactive storytelling. *ACM Transactions on Interactive Intelligent Systems (TiiS)*, ACM New York, NY, USA, v. 2, n. 3, p. 1–35, 2012. Citado na página 25.

SYLVESTER, T. *Designing games: A guide to engineering experiences*. [S.l.]: O'Reilly Media, 2013. Citado 2 vezes nas páginas 23 e 25.

- TEAM, G. et al. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*, 2023. Citado na página 9.
- TEAM, G. et al. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv preprint arXiv:2403.05530*, 2024. Citado na página 20.
- TEKINBAS, K. S.; ZIMMERMAN, E. *Rules of play: Game design fundamentals*. [S.l.]: MIT press, 2003. Citado 4 vezes nas páginas 23, 24, 25 e 26.
- TYCHSEN, A. et al. The game master. In: *Proceedings of the second Australasian conference on Interactive entertainment*. [S.l.: s.n.], 2005. p. 215–222. Citado 2 vezes nas páginas 24 e 25.
- VASWANI, A. et al. Attention is all you need. *Advances in neural information processing systems*, v. 30, 2017. Citado 2 vezes nas páginas 20 e 27.
- WANG, L. et al. Characterbox: Evaluating the role-playing capabilities of llms in text-based virtual worlds. In: *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*. [S.l.: s.n.], 2025. p. 6372–6391. Citado 6 vezes nas páginas 9, 10, 11, 12, 13 e 17.
- YANNAKAKIS, G. N.; TOGELIUS, J. *Artificial intelligence and games*. [S.l.]: Springer, 2018. v. 2. Citado 4 vezes nas páginas 21, 22, 23 e 26.
- YU, P. et al. Rpgbench: Evaluating large language models as role-playing game engines. *arXiv preprint arXiv:2502.00595*, 2025. Citado 6 vezes nas páginas 9, 10, 11, 12, 13 e 16.
- ZHAO, W. X. et al. A survey of large language models. *arXiv preprint arXiv:2303.18223*, v. 1, n. 2, 2023. Citado 2 vezes nas páginas 21 e 27.