

# Wumpus Verse: A Web Platform Prototype for Development and Evaluation of Agents in the Wumpus World

José R. M. Pereira<sup>1</sup>, Williams D. E. Duarte<sup>2</sup>, Pedro P. V. Barroso<sup>1</sup>, Otávio N. Teixeira<sup>1,2</sup>

<sup>1</sup>Campus Universitário de Tucuruí (CAMTUC)  
Universidade Federal do Pará (UFPA)  
Tucuruí – PA – Brazil

<sup>2</sup>Núcleo de Desenvolvimento da Amazônia (NDAE)  
Universidade Federal do Pará (UFPA)  
Tucuruí – PA – Brazil

jose.r.maciél@outlook.com, {pedrovbarroso087, onoura, davidduart04}@gmail.com

**Abstract.** *This paper presents the prototype of a web platform designed to provide an intuitive, pleasant, and practical testing environment for intelligent agents within the Wumpus World. Aimed at experimentation and research in Artificial Intelligence, the tool allows users to freely customize environments and configure different types of agents. The system enables the execution of combined experiments with performance metrics generation, while ensuring the persistence and reproduction of past simulations. The platform simplifies practical experimentation by providing a visual ecosystem that brings users closer to artificial intelligence concepts.*

**Resumo.** *Este trabalho apresenta o protótipo de uma plataforma web projetada para oferecer um ambiente de testes intuitivo, agradável e prático para agentes inteligentes no ambiente do Mundo do Wumpus. Voltada para a experimentação e à pesquisa em Inteligência Artificial, a ferramenta permite que o usuário customize ambientes livremente e configure diferentes tipos de agentes. O sistema viabiliza a execução de experimentos combinados com geração de métricas de desempenho, além de garantir a persistência e a reprodução de simulações passadas. A plataforma simplifica a experimentação prática, fornecendo um ecossistema visual que aproxima usuários de conceitos de inteligência artificial.*

## 1. Introdução

O ensino e a experimentação em Inteligência Artificial (IA) frequentemente utilizam ambientes simplificados que permitem explorar conceitos fundamentais sem a complexidade inerente aos problemas do mundo real. Nesse contexto, o Mundo do Wumpus destaca-se como um dos exemplos clássicos para o estudo de agentes inteligentes e processos de tomada de decisão, combinando regras relativamente simples com desafios relacionados à percepção parcial do ambiente, raciocínio sob incerteza e planejamento. O conceito tem origem no jogo Hunt the Wumpus, proposto por [Yob 1975], com o objetivo de criar um desafio de exploração e sobrevivência em um ambiente desconhecido e dinâmico, envolvendo elementos como armadilhas, um inimigo oculto e a necessidade de localização espacial do jogador. Posteriormente, [Russell and Norvig 2021] incorporaram esse cenário ao contexto da Inteligência Artificial, consolidando o Mundo do Wumpus como um

ambiente clássico para o estudo de agentes inteligentes, representação do conhecimento e mecanismos de inferência lógica.

No Mundo do Wumpus, o agente navega em um grid parcialmente observável contendo poços, uma criatura hostil e ouro, devendo coletar o ouro e retornar à saída sem ser eliminado. O agente utiliza percepções locais (como brisa para indicar poços adjacentes e fedor para proximidade do Wumpus) para inferir perigos, e dispõe de flechas limitadas para eliminar a criatura, o que exige decisões estratégicas ao longo da exploração.

O desempenho do agente é avaliado por meio de um sistema de pontuação associado às ações executadas durante a simulação, incluindo movimentação, utilização de flechas, coleta do ouro e sobrevivência. Esse mecanismo permite a comparação objetiva entre diferentes estratégias e abordagens de tomada de decisão.

O Mundo do Wumpus é amplamente empregado para demonstrar o comportamento de agentes inteligentes em ambientes parcialmente observáveis e incertos, permitindo a investigação de diferentes estratégias de resolução. Além do estudo de agentes baseados em lógica, o ambiente pode ser adaptado para pesquisas envolvendo ambientes dinâmicos, algoritmos evolutivos, sistemas multiagentes e outras abordagens relacionadas à Inteligência Artificial. Sua flexibilidade faz com que seja utilizado tanto em atividades de ensino quanto em projetos de pesquisa.

Embora o Wumpus World seja amplamente utilizado como benchmark, a ausência de uma plataforma unificada que padronize a configuração de ambientes, a definição de agentes e a coleta de métricas dificulta a comparação objetiva entre diferentes abordagens e a reprodução de experimentos. Pesquisadores frequentemente precisam reimplementar o ambiente ou adaptar código legado, o que consome tempo e introduz vieses.

Diante desse cenário, este trabalho apresenta o projeto “Wumpus Verse”, um protótipo de uma plataforma web que elimina a necessidade de uma instalação local preservando sua flexibilidade e potencial para experimentação. A ferramenta oferece uma interface gráfica amigável e um conjunto de funcionalidades que permitem aos usuários sem experiência em programação criar e personalizar mundos, configurar diferentes tipos de agentes inteligentes, executar experimentos e armazenar os resultados obtidos. Além disso, por funcionar em ambiente web, a plataforma amplia a acessibilidade e fornece um ecossistema integrado para atividades de ensino, aprendizagem e pesquisa envolvendo agentes inteligentes.

## **2. Trabalhos Relacionados**

Como mencionado anteriormente, os trabalhos relacionados ao Mundo do Wumpus concentram-se, em sua maioria, no desenvolvimento e na avaliação de estratégias para agentes inteligentes, utilizando o ambiente como um cenário de testes para diferentes abordagens de Inteligência Artificial. Em geral, o foco dessas pesquisas está nos algoritmos e nos resultados obtidos pelos agentes, enquanto aspectos relacionados à acessibilidade, à integração das funcionalidades e ao suporte à experimentação por usuários finais recebem menor atenção.

Com o objetivo de identificar plataformas voltadas à execução e avaliação de agentes por meio de interfaces acessíveis ao usuário, foi realizada uma pesquisa bibliográfica abrangendo trabalhos que apresentassem simuladores ou ambientes de experimentação

para o Mundo do Wumpus. Embora tenham sido encontradas diversas propostas de implementação do ambiente, observou-se que a maioria delas é destinada a fins específicos de pesquisa, frequentemente limitada a interfaces textuais ou à interação direta por meio de código. Além disso, as ferramentas que oferecem algum nível de interação gráfica ou configuração por parte do usuário correspondem, em sua maioria, a propostas desenvolvidas há mais de uma década e apresentam funcionalidades restritas quando comparadas às possibilidades oferecidas por plataformas web modernas.

Nesse contexto, a análise dos trabalhos encontrados é relevante para compreender como essas soluções foram concebidas, quais funcionalidades disponibilizam aos usuários e quais limitações ainda permanecem. Essa discussão também permite evidenciar a contribuição da plataforma proposta neste trabalho, que busca integrar, em um único ambiente web, recursos de criação de mundos, configuração de agentes, execução de experimentos e gerenciamento dos resultados obtidos.

[Julien and Machado 2015] apresentam uma implementação do Mundo do Wumpus baseada em um sistema multiagente utilizando o framework JADE e programação lógica em Prolog. O trabalho concentra-se na modelagem do comportamento dos agentes, nos mecanismos de comunicação e nos algoritmos de inferência utilizados para a resolução do problema, disponibilizando uma interface gráfica para visualização do ambiente. Diferentemente da proposta apresentada neste trabalho, a ferramenta é concebida como um suporte à implementação e avaliação da arquitetura multiagente, não contemplando recursos voltados a experimentações em lote.

[Salehi et al. 2012] propõem um ambiente de testes baseado em uma versão estendida do Mundo do Wumpus, projetada para simular restrições presentes em cenários reais de trabalho cooperativo entre agentes. A plataforma introduz novos elementos de complexidade, como múltiplos contextos, sensores imperfeitos e compartilhamento de informações entre agentes, sendo implementada em Java utilizando o framework JADE. Os autores destacam ainda a flexibilidade do ambiente e a disponibilidade de uma interface de configuração para apoiar a avaliação de problemas relacionados à cooperação, coordenação, planejamento e negociação em sistemas multiagentes. Embora a proposta apresente mecanismos de configuração e interação com o usuário, seu principal objetivo é servir como um ambiente especializado para a investigação de estratégias de cooperação em sistemas multiagentes.

[Li et al. 2025] apresentam o LLM-Cave, um ambiente baseado no Mundo do Wumpus desenvolvido para avaliação das capacidades de raciocínio e tomada de decisão de grandes modelos de linguagem. A plataforma adapta o problema clássico para uma interação totalmente textual, fornecendo ao agente descrições estruturadas do ambiente e métricas quantitativas de desempenho, como taxa de sucesso, recompensa média e custo computacional. O trabalho utiliza esse ambiente para investigar estratégias de inferência, como Chain of Speculation e Planner-Critic, aplicadas à navegação e resolução do problema. Embora o LLM-Cave forneça um ambiente flexível para a avaliação de agentes baseados em modelos de linguagem, seu foco está na análise do desempenho e das estratégias de inferência desses agentes.

Trabalhos como o ALP4AI [Lasisi et al. 2022] e a abordagem de Bryce [Bryce 2011] exploram o uso do Mundo do Wumpus no ensino de IA, oferecendo interfa-

ces gráficas e simuladores para atividades práticas com algoritmos de busca e inferência lógica. Embora compartilhem o objetivo de tornar o ambiente mais acessível, ambas as propostas são voltadas exclusivamente ao contexto educacional, com suporte limitado a experimentação em lote, persistência de dados e comparação objetiva de agentes, lacunas que o Wumpus Verse busca preencher ao integrar esses recursos em uma plataforma web unificada.

Embora o Mundo do Wumpus seja amplamente empregado como referência em ensino e pesquisa, a maioria das implementações existentes direciona-se a nichos específicos (multiagentes, LLMs ou algoritmos clássicos) e oferece suporte limitado à gestão integrada de experimentos. O Wumpus Verse complementa essas propostas ao concentrar, em uma única plataforma web, funcionalidades de criação de cenários, configuração de agentes, execução, persistência e análise de desempenho, atendendo tanto a fins educacionais quanto à pesquisa experimental.

### **3. Metodologia**

A metodologia adotada para o desenvolvimento da plataforma Wumpus Verse baseou-se na separação entre frontend, backend e lógica dos agentes, seguindo uma arquitetura cliente-servidor de três camadas. Esta seção descreve os requisitos estabelecidos para o sistema, a arquitetura e tecnologias empregadas, os agentes implementados, os mecanismos de persistência de dados e a interface desenvolvida para a plataforma.

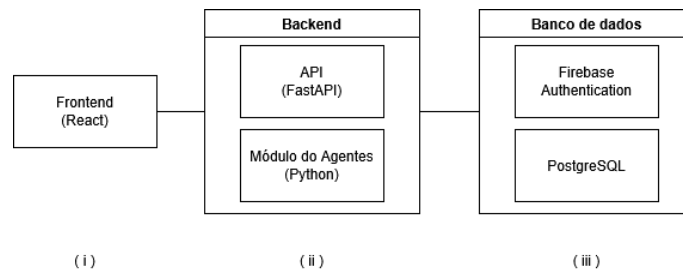
#### **3.1. Requisitos, Arquitetura e Tecnologias**

Para orientar o desenvolvimento da plataforma Wumpus Verse, foram definidos requisitos funcionais e não funcionais (Tabela 1) que garantem flexibilidade, personalização e boa experiência ao usuário, além de uma arquitetura e tecnologias que agilizem e tornem o processo eficiente.

Com base nesses requisitos, a arquitetura da plataforma foi projetada seguindo o modelo cliente-servidor de três camadas, conforme ilustrado na Figura 1. A escolha das tecnologias para cada camada considerou critérios de produtividade, integração e maturidade do ecossistema. A camada de frontend (I), desenvolvida com a biblioteca React, é responsável pela interface gráfica e pelo consumo da API REST. O React foi selecionado por oferecer componentização, Virtual DOM e um ecossistema maduro, fatores que agilizam o desenvolvimento de interfaces interativas. A camada de backend (II), implementada em Python com o framework FastAPI, gerencia as requisições, coordena as simulações e controla o acesso ao banco de dados. A combinação de Python com FastAPI foi escolhida por proporcionar agilidade no desenvolvimento e integração direta com os agentes (que também são implementados em Python) eliminando a sobrecarga de comunicação entre linguagens e permitindo o reuso de código entre a API e os agentes. Por fim, a camada de persistência (III) é composta por dois sistemas distintos: o Firebase Authentication, responsável pelo gerenciamento de usuários, e o PostgreSQL, que armazena os dados específicos da aplicação. Essa separação de responsabilidades, aliada à escolha tecnológica adequada, permite um desenvolvimento modular e facilita a manutenção e evolução do sistema.

| Código | Tipo          | Descrição                                                                               |
|--------|---------------|-----------------------------------------------------------------------------------------|
| RF01   | Funcional     | Criação de grids de qualquer formato (não apenas configurações quadradas).              |
| RF02   | Funcional     | Definição de parâmetros, objetivos e funções de avaliação para agentes.                 |
| RF03   | Funcional     | Execução de agentes personalizados em ambientes salvos.                                 |
| RF04   | Funcional     | Salvamento de resultados e histórico de cada simulação.                                 |
| RF05   | Funcional     | Geração de relatórios com pontuação, indicadores e caminho percorrido.                  |
| RF06   | Funcional     | Exportação de dados em JSON (ambientes, agentes e execuções).                           |
| RNF01  | Não Funcional | Multiplataforma: funcionamento em navegadores modernos (Chrome, Firefox, Edge, Safari). |
| RNF02  | Não Funcional | Responsividade parcial para dispositivos móveis (funcionalidades reduzidas).            |
| RNF03  | Não Funcional | Disponibilidade 24/7 durante o período de avaliação.                                    |
| RNF04  | Não Funcional | Tempo de resposta menos de 5 segundos por requisição.                                   |

**Tabela 1. Requisitos funcionais e não funcionais da plataforma Wumpus Verse.**



**Figura 1. Diagrama da arquitetura do sistema.**

### 3.2. Agentes

A plataforma Wumpus Verse disponibiliza três tipos de agentes com diferentes estratégias de decisão, todos projetados para maximizar a pontuação final, seja coletando o ouro, eliminando o Wumpus ou retornando à saída. A implementação desses agentes permite que os usuários comparem diferentes abordagens para o mesmo problema, facilitando a compreensão das vantagens e limitações de cada estratégia.

O **Agente Reativo Aleatório** implementa uma abordagem simples baseada em regras. A cada passo, o agente recebe as percepções da sala atual. Se detecta ouro, o agente o coleta imediatamente. Se detecta fedor, existe uma chance de 10% de o agente atirar ou mover-se em uma direção aleatória. Este agente ignora completamente a percepção de brisa, o que o torna vulnerável a cair em poços, mas serve como um baseline para comparação com estratégias mais sofisticadas.

O **Agente Baseado em Modelo** mantém um estado interno, ou crença, que é atualizado continuamente com base nas percepções recebidas. Salas consideradas seguras são exploradas, enquanto salas com indícios de perigo são evitadas. Este agente utiliza o algoritmo A\* para planejar rotas ótimas entre sua posição atual e os destinos desejados. O agente pode ser configurado com quatro perfis de comportamento distintos, que determinam sua estratégia de priorização:

- **Perfil Assassino:** prioriza a localização e eliminação do Wumpus, mesmo que isso implique assumir riscos maiores.
- **Perfil Garimpeiro:** prioriza a coleta do ouro como objetivo principal, evitando confrontos desnecessários.
- **Perfil Explorador:** continua explorando o ambiente mesmo após cumprir seu objetivo inicial, buscando mapear completamente o cenário.
- **Perfil Corajoso:** está disposto a arriscar entrar em salas consideradas perigosas, confiando em sua capacidade de reagir a imprevistos.

O **Agente Evolutivo** emprega um algoritmo genético para otimizar uma sequência de ações (cromossomo), permitindo ao usuário configurar parâmetros populacionais, taxas genéticas e uma função fitness personalizada, combinação linear de passos, coleta de ouro e sobrevivência. A cada geração, os indivíduos são executados no ambiente, e os melhores são selecionados, recombinados e mutados; ao final, o melhor cromossomo define a política de ação do agente.

### 3.3. Armazenamento e Backend

O backend da plataforma Wumpus Verse é responsável por expor a API REST e gerenciar a persistência dos dados, assegurando que todas as informações geradas pelos usuários sejam armazenadas de forma segura e consistente. Para isso, sua arquitetura integra três aspectos fundamentais: a estrutura da API, a integração direta com os agentes e os mecanismos de persistência.

Desenvolvida em Python com o framework FastAPI, a API disponibiliza endpoints que cobrem todas as funcionalidades da plataforma. Os principais endpoints estão listados na Tabela 2, que apresenta as rotas, métodos HTTP e descrições das operações disponíveis. A comunicação entre cliente e servidor utiliza JSON como formato de dados, e a maioria das requisições é autenticada via token JWT integrado ao Firebase Authentication, garantindo que apenas usuários autorizados possam acessar e modificar seus dados.

Complementando a estrutura da API, a integração com os agentes foi projetada para otimizar o desempenho e a confiabilidade do sistema. Como os agentes também são implementados em Python, o backend os importa diretamente como módulos, eliminando a sobrecarga e a complexidade da comunicação entre linguagens diferentes. A simulação ocorre em um ambiente controlado (classe `Environment`), que recebe o agente e o grid, executa as ações passo a passo e coleta métricas como número de passos, ouro coletado, Wumpus morto e sobrevivência. Ao final da simulação, o sistema calcula a pontuação final do agente de acordo com as regras estabelecidas. Esse desenho permite a reciclagem de código entre a API e os agentes, reduzindo significativamente a complexidade e os possíveis pontos de falha.

Por fim, no que tange à persistência de dados, a plataforma adota dois sistemas de armazenamento complementares para gerenciar diferentes aspectos das informações. O

| <b>Rota</b>                                  | <b>Método</b> | <b>Descrição</b>                              |
|----------------------------------------------|---------------|-----------------------------------------------|
| /environment/list-user,<br>/environment/user | GET, POST     | Lista ou cria ambientes do usuário            |
| /environment/execution                       | POST          | Executa um agente em um ambiente (recebe IDs) |
| /auth/register,<br>/auth/login               | POST          | Registro/login, retorna token JWT             |
| /agents/user,<br>/agents/agents              | POST, GET     | Cria ou lista agentes personalizado           |
| /execution/user,<br>/execution/list-user     | POST, GET     | Salva ou lista execuções                      |
| /execution/statics                           | POST          | Analisa agentes em um ambiente (estatísticas) |

**Tabela 2. Endpoints da API para gerenciamento de ambientes, agentes e execuções.**

**Firestore Authentication** é responsável pelo gerenciamento de cadastro, login e redefinição de senhas de usuários, fornecendo uma infraestrutura pronta e segura que elimina a necessidade de implementar esses mecanismos do zero. O **PostgreSQL**, por sua vez, é utilizado para armazenar os dados específicos da aplicação, incluindo ambientes (grid, posições), agentes (parâmetros, tipo, função fitness) e execuções (histórico, ouros coletados, Wumpus mortos, pontuação final). A escolha do PostgreSQL garante consistência transacional e integridade referencial, características essenciais para um sistema que lida com dados complexos e inter-relacionados. Dessa forma, a combinação dessas duas ferramentas proporciona uma base robusta e escalável para todas as operações de gerenciamento de usuários e experimentos realizados na plataforma.

### 3.4. Frontend

frontend da plataforma foi desenvolvido utilizando a biblioteca JavaScript React, permitindo a construção de uma interface web dinâmica e responsiva. A arquitetura da interface foi organizada em diferentes módulos, cada um responsável por uma etapa do fluxo de utilização do sistema, desde o acesso inicial até a execução e análise dos experimentos. Esta seção descreve os principais módulos da interface, relacionando cada funcionalidade aos requisitos funcionais estabelecidos na Seção 3.1.

A **tela de apresentação** constitui a página inicial da plataforma, fornecendo acesso às principais funcionalidades disponíveis, atendendo ao requisito de disponibilidade e acesso intuitivo ao sistema. Complementarmente, a seção **Saiba Mais** reúne informações sobre o projeto, sua motivação, a equipe de desenvolvimento e uma breve contextualização sobre o problema do Mundo do Wumpus.

O sistema possui um módulo de **autenticação** composto pelas telas de login e registro, responsáveis pelo gerenciamento do acesso dos usuários. O cadastro inclui verificação de e-mail, e apenas usuários autenticados podem acessar as funcionalidades de gerenciamento e execução de experimentos. Este módulo atende indiretamente aos requisitos de persistência (RF04) ao garantir que os dados sejam associados a usuários específicos.

Para a criação e administração dos cenários, a plataforma disponibiliza as telas **Mundos** e **Editor de Mundos**, atendendo ao requisito RF01 (Ambientes Personalizados). A primeira apresenta a lista de ambientes previamente criados pelo usuário, permitindo sua visualização, edição, exclusão, importação e exportação. O editor, por sua vez, possibilita a construção de mundos bidimensionais personalizados, nos quais podem ser posicionadas as entidades do jogo, como Wumpus, poços e ouro. Embora os cenários possam assumir diferentes formatos, sua estrutura é baseada em uma malha de salas quadradas interconectadas. A Figura 2 apresenta exemplos dessas interfaces.

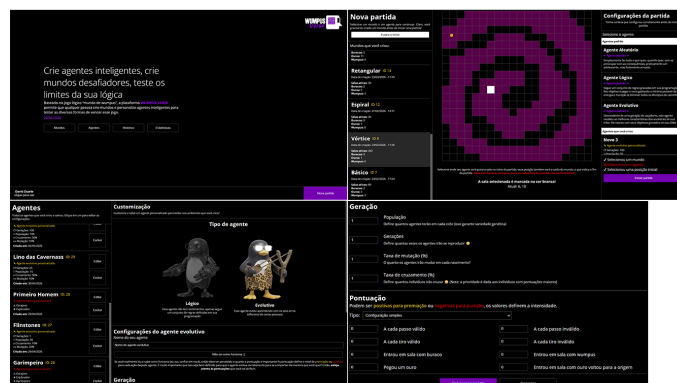


Figura 2. Exemplo de telas da plataforma.

A tela **Agentes** permite criar e editar agentes lógicos e evolutivos (RF02). Para agentes lógicos, o usuário seleciona perfis de comportamento (assassino, garimpeiro, explorador, corajoso); para evolutivos, define parâmetros genéticos e função fitness personalizada, podendo experimentar combinações e observar os efeitos no comportamento.

As telas **Histórico** e **Estatísticas** (RF04 e RF05) gerenciam os resultados experimentais, permitindo recuperar execuções salvas e comparar agentes por métricas como pontuação média, passos e disparos. Para iniciar uma simulação, a tela **Nova Partida** (RF03) define o mundo, o agente e a posição inicial; na tela **Execução**, o trajeto do agente é reproduzido graficamente a partir do roteiro gerado pelo backend, com controles para navegação, repetição, ajuste de parâmetros e visualização das ações (Figura 3).

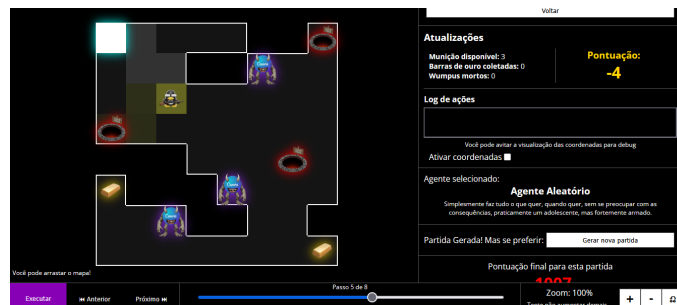


Figura 3. Interface de execução da plataforma durante uma simulação.

Além disso, o usuário pode armazenar a execução em sua conta para futuras análises, bem como exportar ou importar partidas por meio de arquivos, atendendo ao requisito RF06 (Exportação). A funcionalidade de exportação permite que os dados sejam salvos

em formato JSON, enquanto a importação possibilita a reconstrução de ambientes e execuções previamente exportados.

Conforme mencionado anteriormente, a plataforma foi desenvolvida seguindo princípios de responsividade, permitindo seu acesso também por meio de dispositivos móveis. Entretanto, determinadas funcionalidades, como a criação e edição de mundos, permanecem disponíveis apenas em ambientes com teclado e mouse, uma vez que a definição da estrutura do cenário depende de interações de precisão baseadas no uso simultâneo do cursor e de comandos do teclado. Dessa forma, a adaptação para dispositivos móveis prioriza a consulta e a execução de experimentos, preservando a usabilidade das funcionalidades que demandam maior nível de interação. A Figura 4 apresenta a interface da plataforma em um dispositivo móvel, evidenciando a adaptação do layout para telas de menor dimensão.

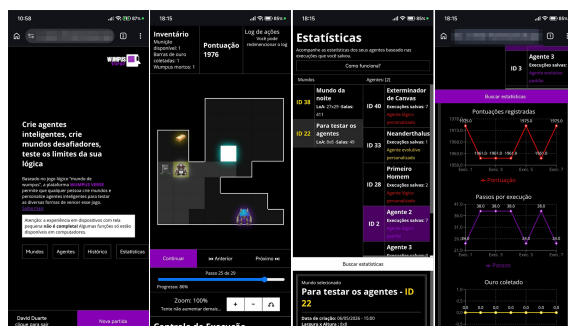


Figura 4. Interface responsiva da plataforma em dispositivo móvel.

## 4. Análise Experimental

A avaliação da plataforma Wumpus Verse foi conduzida em duas frentes: (i) verificação do funcionamento dos módulos e integração (testes funcionais) e (ii) análise quantitativa do desempenho dos agentes implementados, com o objetivo de demonstrar a utilidade da ferramenta para experimentação comparativa em pesquisa.

Os testes funcionais, executados manualmente com base em cenários de uso, cobriram as operações essenciais da plataforma: cadastro e autenticação, criação e edição de mundos, configuração de agentes, execução de simulações, persistência de resultados e importação/exportação de dados. Todas essas funcionalidades apresentaram comportamento esperado, atestando a maturidade do protótipo para uso em experimentos controlados. Os detalhes dos casos de teste estão disponíveis no repositório do projeto.

### 4.1. Análise Quantitativa de Desempenho dos Agentes

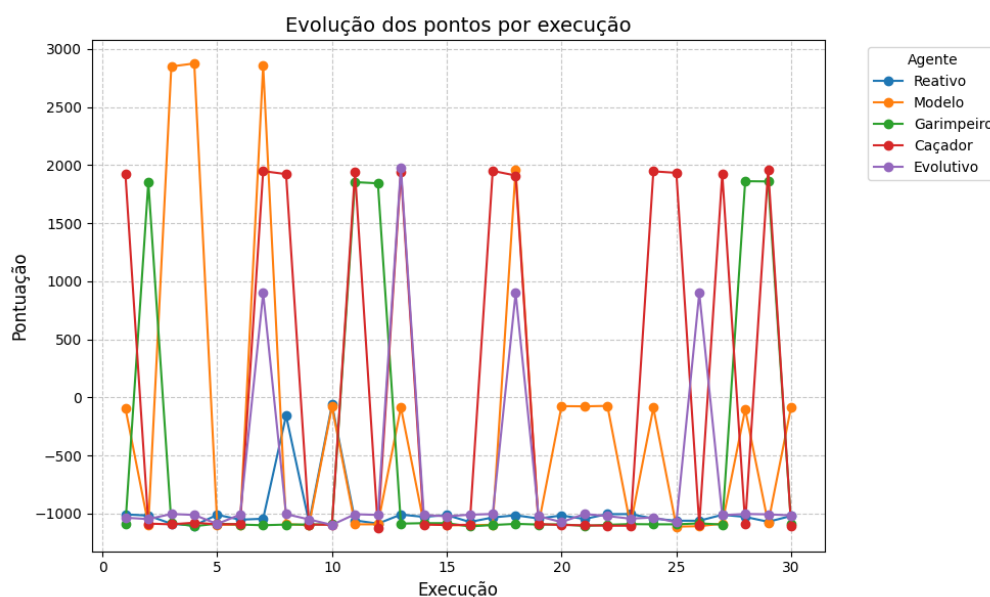
Para avaliar a capacidade da plataforma como ambiente de *benchmarking*, foram conduzidos experimentos com cinco agentes de diferentes estratégias em um cenário padronizado. O ambiente utilizado, ilustrado na Figura 3, consiste em um grid  $8 \times 8$  com 3 Wumpus, 3 poços e 2 ouros, dispostos de forma a criar um nível mediano de complexidade. A sala  $(0, 0)$ , destacada em branco, foi definida como ponto inicial para todas as simulações.

Foram testados cinco agentes, abrangendo diferentes níveis de sofisticação:

- **Reativo Aleatório:** agente sem memória, com decisões aleatórias baseadas em percepções.

- **Baseado em Modelo (padrão):** com todos os perfis ativados (Garimpeiro, Caçador, Explorador, Corajoso).
- **Garimpeiro:** prioriza a coleta de ouro.
- **Caçador:** prioriza a eliminação do Wumpus.
- **Evolutivo:** algoritmo genético (população 100, 1000 gerações, crossover 50%, mutação 5%) com função *fitness* linear combinando passos, coleta de ouro e sobrevivência.

Cada agente foi executado 30 vezes no mesmo ambiente. A Figura 5 apresenta a evolução das pontuações ao longo das execuções, e a Tabela 3 resume as estatísticas descritivas.



**Figura 5. Evolução das pontuações por execução para cada agente.**

| Agente     | Média   | Desvio Padrão | Máximo | Mediana | Mínimo |
|------------|---------|---------------|--------|---------|--------|
| Reativo    | -978,5  | 238,32        | -62    | -1031,5 | -1103  |
| Modelo     | -294,9  | 1263,43       | 2876   | -1087,0 | -1113  |
| Garimpeiro | -602,1  | 1117,57       | 1862   | -1092,0 | -1111  |
| Caçador    | 15,5    | 1487,29       | 1963   | -1091,0 | -1125  |
| Evolutivo  | -735,53 | 779,57        | 1979   | -1012,5 | -1098  |

**Tabela 3. Estatísticas descritivas das pontuações (30 execuções por agente).**

Os resultados revelam diferenças substanciais entre as estratégias. O **agente reativo** apresenta o pior desempenho (média  $-978,5$ ), confirmando a ineficácia de decisões puramente aleatórias. O **agente baseado em modelo padrão** obtém média intermediária ( $-294,9$ ), mas com alta variabilidade (desvio padrão de  $1263,43$ ), indicando sucessos esporádicos (máximo de  $2876$ ) em meio a muitas falhas (mediana  $-1087$ ).

O **agente Caçador** alcançou a melhor média ( $15,5$ ), sugerindo que, neste cenário, a eliminação do Wumpus é uma estratégia particularmente eficaz. Contudo, seu elevado

desvio padrão (1487,29) decorre do perfil *Corajoso*, que, ao incentivar a exploração de salas perigosas mesmo após atingir objetivos parciais, gora execuções bem-sucedidas com mortes tardias, penalizando a pontuação final. O **Garimpeiro** apresentou desempenho inferior ao modelo padrão (média -602, 1), possivelmente porque a priorização do ouro o expõe a riscos desnecessários neste ambiente.

O **agente evolutivo**, apesar do custo computacional mais elevado, apresentou a segunda melhor mediana (-1012, 5), superando o modelo padrão e o garimpeiro. Sua média de -735, 53 e desvio padrão de 779,57 indicam maior consistência, com menos execuções extremamente negativas. A pontuação máxima de 1979 demonstra que o algoritmo genético é capaz de encontrar soluções de alta qualidade, embora ainda haja espaço para melhorias na robustez.

## 4.2. Discussão

Os experimentos demonstram que a plataforma Wumpus Verse é capaz de produzir resultados comparativos reproduzíveis entre diferentes agentes, fornecendo métricas objetivas (pontuação, passos, sobrevivência) que permitem avaliar o impacto de distintas estratégias de decisão. A flexibilidade na configuração de agentes e ambientes, aliada à persistência e exportação de dados, torna a ferramenta adequada para pesquisas que demandam experimentação sistemática e *benchmarking* em cenários variados.

Os testes funcionais, embora não detalhados neste artigo, validam a robustez da implementação para as operações de criação, execução e gerenciamento, enquanto os resultados quantitativos evidenciam a utilidade da plataforma para análises comparativas. Como trabalhos futuros, planeja-se ampliar o conjunto de métricas, suportar experimentos em lote e integrar novos tipos de agentes, consolidando o Wumpus Verse como um ambiente de referência para a comunidade de IA.

## 5. Conclusão

Este trabalho apresentou o Wumpus Verse, um protótipo de plataforma web desenvolvido para apoiar a criação, execução e avaliação de agentes inteligentes no ambiente do Mundo do Wumpus. A proposta surgiu da necessidade de disponibilizar uma ferramenta acessível e integrada que permitisse a experimentação com agentes inteligentes sem exigir dos usuários conhecimentos avançados de programação ou configuração de ambientes de teste.

A plataforma foi projetada para oferecer recursos que abrangem todo o ciclo de experimentação, incluindo a criação de mundos personalizados, a configuração de diferentes tipos de agentes, a execução de simulações, o armazenamento de resultados e a geração de estatísticas de desempenho. Além disso, sua implementação em ambiente web amplia a acessibilidade da ferramenta, permitindo sua utilização para a realização de experimentação em lote e padronização de métricas.

Os testes realizados demonstraram o funcionamento adequado das principais funcionalidades do sistema, evidenciando a integração entre os módulos desenvolvidos e a capacidade da plataforma de fornecer um ambiente consistente para a realização de experimentos envolvendo agentes inteligentes.

Como perspectivas futuras, pretende-se ampliar as funcionalidades da plataforma de modo a torná-la ainda mais flexível e personalizável. Uma das principais evoluções

planejadas está relacionada ao modelo de construção dos ambientes. Atualmente, embora os mundos possam assumir diferentes formatos por meio da adição de salas em posições arbitrárias, sua estrutura ainda permanece limitada a uma malha bidimensional. Como extensão, pretende-se permitir a criação de conexões manuais entre salas, de forma semelhante a diagramas ou grafos, possibilitando a construção de ambientes com topologias mais complexas e menos restritas ao modelo tradicional do Mundo de Wumpus.

Também estão previstas melhorias nos mecanismos de execução das simulações, oferecendo maior controle sobre parâmetros experimentais e ampliando as opções de configuração disponíveis aos usuários. Dessa forma, espera-se proporcionar maior flexibilidade para a realização de diferentes tipos de experimentos e análises.

Outra funcionalidade de interesse é o suporte a múltiplos agentes atuando simultaneamente em um mesmo ambiente. Essa extensão permitirá a investigação de cenários cooperativos e competitivos, ampliando significativamente as possibilidades de estudo e avaliação de estratégias de tomada de decisão.

Por fim, prevê-se a incorporação de recursos voltados à interação entre usuários, como compartilhamento de mundos e agentes, sistemas de classificação e mecanismos de colaboração.

Com essas evoluções, espera-se consolidar o Wumpus Verse como uma plataforma cada vez mais completa e acessível para atividades de ensino, pesquisa e experimentação envolvendo agentes inteligentes.

O código-fonte da plataforma e os experimentos descritos estão disponíveis em <https://wumpus-verse-frontend.vercel.app/>.

## Referências

- Bryce, D. (2011). Wumpus world in introductory artificial intelligence. *Journal of Computing Sciences in Colleges*, 27(2):58–65.
- Julien, D. L. L. and Machado, A. (2015). The wumpus world: Implementing a multi-agent system. Technical report, Technical Report, Salvador, Brazil.
- Lasisi, R. O., Philips, C., and Hartnett, N. (2022). ALP4AI: Agent-based learning platform for introductory artificial intelligence. In *Proceedings of the 14th International Conference on Agents and Artificial Intelligence (ICAART 2022)*, pages 1–9.
- Li, H., Li, Z., Huang, W., and Guo, X. (2025). LLM-Cave: A benchmark and light environment for large language models reasoning and decision-making system. arXiv preprint arXiv:2511.22598.
- Russell, S. and Norvig, P. (2021). *Artificial Intelligence: A Modern Approach*. Pearson, 4 edition.
- Salehi, S., Saffar, M. T., Taghiyareh, F., and Badie, K. (2012). A multi-context dynamic test bed for simulating real-world constraints in agents' teamwork. In *Proceedings of the 6th International Symposium on Telecommunications (IST 2012)*, pages 1–6. IEEE.
- Yob, G. (1975). Hunt the wumpus. *Creative Computing*, 1(5):51–54.