



**UNIVERSIDADE DO MINDELO  
DEPARTAMENTO DE ENGENHARIA INFORMÁTICA**

**CURSO DE LICENCIATURA EM ENGENHARIA INFORMÁTICA E  
SISTEMAS COMPUTACIONAIS**

**TRABALHO DE INTELIGÊNCIA ARTIFICIAL**

**Ano letivo 2025/2026 – 3º Ano**

**Tema: Relatório Técnico do Projeto Final – Desenvolvimento de  
um Agente Inteligente para o Jogo Reversi utilizando  
Aprendizagem por Reforço**

**Autor: Aulindo Correia, nº 5720**

**Professor: Estanislau Lima**

**Mindelo, 2025**

## Índice

Índice .....	1
1. Introdução .....	2
2. Enquadramento do Projeto .....	3
2.1 Inteligência Artificial em Jogos .....	4
2.2 Aprendizagem por Reforço .....	4
2.3 O jogo Reversi como Ambiente de Aprendizagem .....	4
3. Descrição do Sistema Desenvolvido .....	5
3.1 Arquitetura Geral .....	5
3.2 Tecnologias Utilizadas .....	6
4. Modelação do Agente Inteligente .....	6
4.1 Definição do Agente .....	6
4.2 Representação do Estado .....	6
4.3 Espaço de Ações .....	6
4.4 Funções de Características .....	7
4.5 Função de Avaliação .....	7
4.6 Estratégia de Exploração .....	7
5. Processo de Treino .....	7
5.1 Ambiente de Treino .....	7
5.2 Parâmetros de Treino .....	7
5.3 Armazenamento do Conhecimento .....	8
6. Avaliação Experimental .....	8
6.1 Metodologia de Avaliação .....	8
6.2 Adversários Utilizados .....	8
6.3 Resultados Obtidos .....	8
6.4 Análise dos Resultados .....	8
7. Interface Gráfica e Experiência do Utilizador .....	8
8. Discussão .....	10
9. Conclusão .....	11
10. Trabalho Futuros e Melhorias .....	12
12. Referências Bibliográficas .....	12

## 1. Introdução

A Inteligência Artificial (IA) tem desempenhado um papel fundamental no desenvolvimento de sistemas capazes de tomar decisões autónomas em ambientes complexos. Entre as várias áreas de aplicação da IA, os jogos de tabuleiro constituem um domínio privilegiado para o estudo e experimentação de agentes inteligentes, uma vez que apresentam regras bem definidas, ambientes observáveis e objetivos claros.

No âmbito da unidade curricular de Inteligência Artificial, foi proposto o desenvolvimento de um sistema inteligente, utilizando técnicas de Machine Learning e neste contexto a Aprendizagem por Reforço (Reinforcement Learning – RL) destaca-se como uma abordagem eficaz para o desenvolvimento de agentes capazes de aprender a partir da experiência, sem necessidade de supervisão explícita. O agente interage com o ambiente, executa ações e recebe recompensas, ajustando progressivamente o seu comportamento com o objetivo de maximizar o retorno acumulado.

O presente projeto tem como objetivo o desenvolvimento de um agente inteligente adaptativo para o jogo Reversi (também conhecido como Othello), integrando técnicas de Aprendizagem por Reforço num jogo totalmente funcional com interface gráfica. O agente inicia o processo de aprendizagem sem conhecimento prévio e evolui ao longo do treino, sendo posteriormente avaliado contra diferentes tipos de adversários.

## 2. Enquadramento do Projeto

### 2.1 Inteligência Artificial em Jogos

Os jogos de tabuleiro são frequentemente utilizados como ambientes de teste para algoritmos de IA, pois envolvem tomada de decisão sequencial, planeamento e avaliação de estados futuros. Um agente inteligente, neste contexto, é definido como uma entidade capaz de perceber o estado do ambiente e escolher ações que maximizem a probabilidade de atingir um objetivo.

### 2.2 Aprendizagem por Reforço

A Aprendizagem por Reforço baseia-se na interação contínua entre um agente e um ambiente. Em cada passo, o agente observa o estado atual, executa uma ação e recebe uma recompensa. Com base nesta informação, o agente ajusta a sua política de decisão.

Os principais componentes da Aprendizagem por Reforço são:

- **Estado:** representação da situação atual do ambiente;
- **Ação:** decisão tomada pelo agente;
- **Recompensa:** feedback numérico recebido após a ação;
- **Política:** estratégia que define a escolha de ações;
- **Exploração vs Exploitação:** equilíbrio entre testar novas ações e utilizar conhecimento já adquirido.

### 2.3 O Jogo Reversi como Ambiente de Aprendizagem

O Reversi é um jogo de tabuleiro para dois jogadores, jogado num tabuleiro 8×8. Os jogadores alternam jogadas, colocando peças que capturam as peças adversárias segundo regras bem definidas. O jogo termina quando não existem mais jogadas possíveis.

As características do Reversi tornam-no adequado à Aprendizagem por Reforço:

- espaço de estados grande;
- decisões estratégicas de curto e longo prazo;
- ausência de informação oculta;
- resultado claramente definido (vitória, derrota ou empate).

### 3. Descrição do Sistema Desenvolvido

#### 3.1 Arquitetura Geral

O sistema foi desenvolvido de forma modular, separando claramente as responsabilidades incluindo treino, avaliação e interação com o utilizador dos componentes:

- Interface Gráfica (reversi.py): responsável pela interação com o utilizador, renderização do tabuleiro, menus e botões;
- Lógica do Jogo (game\_logic.py): implementação das regras do Reversi, validação de jogadas e cálculo de pontuações;
- Agente de Aprendizagem por Reforço (rl\_agent.py): definição do agente, funções de avaliação, treino e avaliação;
- Módulo de Treino e Avaliação (train.py): execução do treino e recolha de estatísticas.

Esta separação permite uma maior clareza, reutilização de código e facilidade de manutenção.

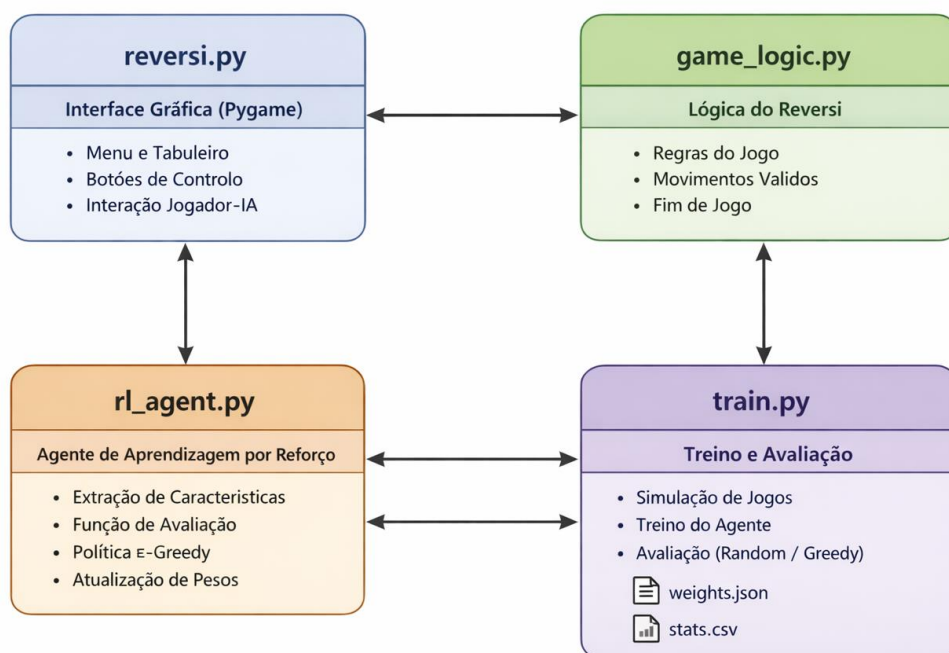


Figura 1 – Arquitetura geral do sistema

#### 3.2 Tecnologias Utilizadas

Foram utilizadas as seguintes tecnologias e bibliotecas:

- **Python** - escolhido pela sua ampla utilização em Inteligência Artificial e pela simplicidade da sua sintaxe, que facilita a implementação e compreensão dos algoritmos de Aprendizagem por Reforço.
- **Pygame** - foi utilizada para o desenvolvimento da interface gráfica do jogo, permitindo criar um ambiente interativo onde é possível observar, em tempo real, o comportamento do agente inteligente.
- **Matplotlib** - Para a análise e apresentação dos resultados experimentais, que possibilitou a visualização da evolução do desempenho do agente ao longo do treino.
- **JSON** e **CSV** foram utilizados para o armazenamento de dados, permitindo guardar os pesos aprendidos pelo agente e as estatísticas de avaliação de forma simples, portátil e facilmente reutilizável.

## 4. Modelação do Agente Inteligente

### 4.1 Definição do Agente

O agente inteligente corresponde ao jogador que utiliza as peças pretas. O seu objetivo é maximizar a probabilidade de vitória, aprendendo com base na experiência acumulada ao longo das partidas.

### 4.2 Representação do Estado

O estado do ambiente é representado pelo tabuleiro do jogo, modelado como uma matriz 8×8. Cada célula pode assumir três valores:

- peça preta (agente);
- peça branca (adversário);
- espaço vazio.

### 4.3 Espaço de Ações

O conjunto de ações possíveis corresponde às jogadas válidas disponíveis em cada estado, de acordo com as regras do Reversi.

## 4.4 Funções de Características

Para permitir a aprendizagem, o agente utiliza uma representação baseada em características extraídas do tabuleiro após cada jogada:

- **Diferença de peças** entre o agente e o adversário;
- **Mobilidade**, medida pelo número de jogadas válidas disponíveis;
- **Ocupação de cantos**, considerados posições estrategicamente fortes;
- **Ocupação de bordas**, excluindo os cantos.

Estas características são normalizadas para manter valores comparáveis.

## 4.5 Função de Avaliação

A função de avaliação consiste numa combinação linear das características, ponderadas por pesos aprendidos durante o treino. O valor resultante indica a qualidade de um estado após a execução de uma jogada.

## 4.6 Estratégia de Exploração

Foi utilizada uma política  $\epsilon$ -greedy, em que:

- com probabilidade  $\epsilon$  o agente escolhe uma jogada aleatória (exploração);
- com probabilidade  $1-\epsilon$  escolhe a melhor jogada conhecida (exploração).

O valor de  $\epsilon$  decresce gradualmente ao longo do treino.

# 5. Processo de Treino

## 5.1 Ambiente de Treino

O treino é realizado através de partidas simuladas sem interface gráfica, permitindo a execução rápida de múltiplos episódios. O agente joga contra um adversário aleatório durante o treino.

## 5.2 Parâmetros de Treino

- Número total de jogos de treino: 4000;
- Taxa de aprendizagem ( $\alpha$ ): 0.05;

- $\epsilon$  inicial: 1.0
- $\epsilon$  final: 0.1

### 5.3 Armazenamento do Conhecimento

Os pesos aprendidos são guardados no ficheiro weights.json, enquanto as estatísticas de avaliação são armazenadas em stats.csv.

## 6. Avaliação Experimental

### 6.1 Metodologia de Avaliação

A avaliação do agente é realizada sem atualização dos pesos, garantindo que o desempenho medido reflete apenas o conhecimento adquirido. Cada avaliação consiste em 2000 jogos, utilizando diferentes sementes aleatórias para garantir reprodutibilidade.

### 6.2 Adversários Utilizados

- **Adversário Aleatório:** seleciona jogadas válidas de forma aleatória;
- **Adversário Greedy:** escolhe sempre a jogada que maximiza o ganho imediato de peças.

### 6.3 Resultados Obtidos

Os resultados mostram uma rápida convergência do desempenho contra o adversário aleatório, com taxas de vitória superiores a 80%. Contra o adversário greedy, o agente não obteve vitórias, evidenciando limitações de generalização.

### 6.4 Análise dos Resultados

A rápida convergência contra o adversário aleatório indica que o agente conseguiu aprender uma política eficaz para esse tipo de comportamento. Por outro lado, o desempenho nulo contra o adversário greedy revela uma limitação da aprendizagem baseada exclusivamente num único tipo de adversário durante o treino, fenómeno conhecido como sobreajuste (overfitting).



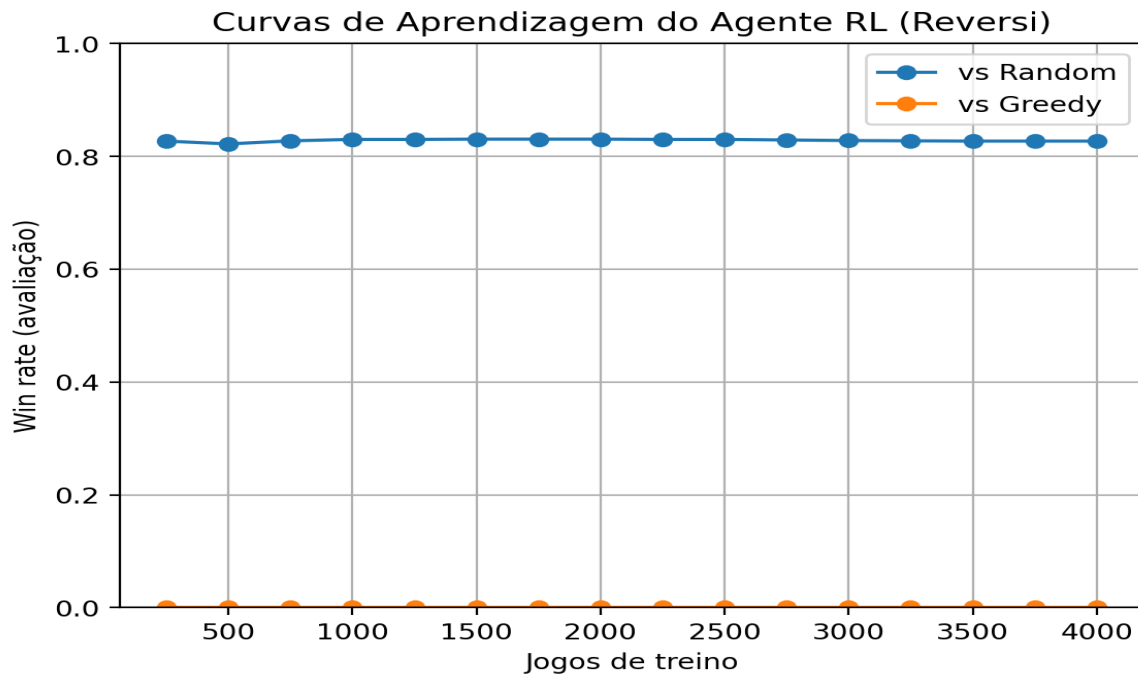


Figura 2 – Curva de aprendizagem do agente

## 7. Interface Gráfica e Experiência do Utilizador

A interface gráfica foi desenvolvida com foco na usabilidade e clareza visual. O sistema inclui:

- menu inicial com botões de controlo;
- painel informativo da IA, exibindo dados do treino;
- destaque visual das jogadas do jogador e do agente;
- botões dedicados para iniciar um novo jogo e sair da aplicação.

Estas funcionalidades permitem ao utilizador acompanhar o comportamento do agente de forma intuitiva.

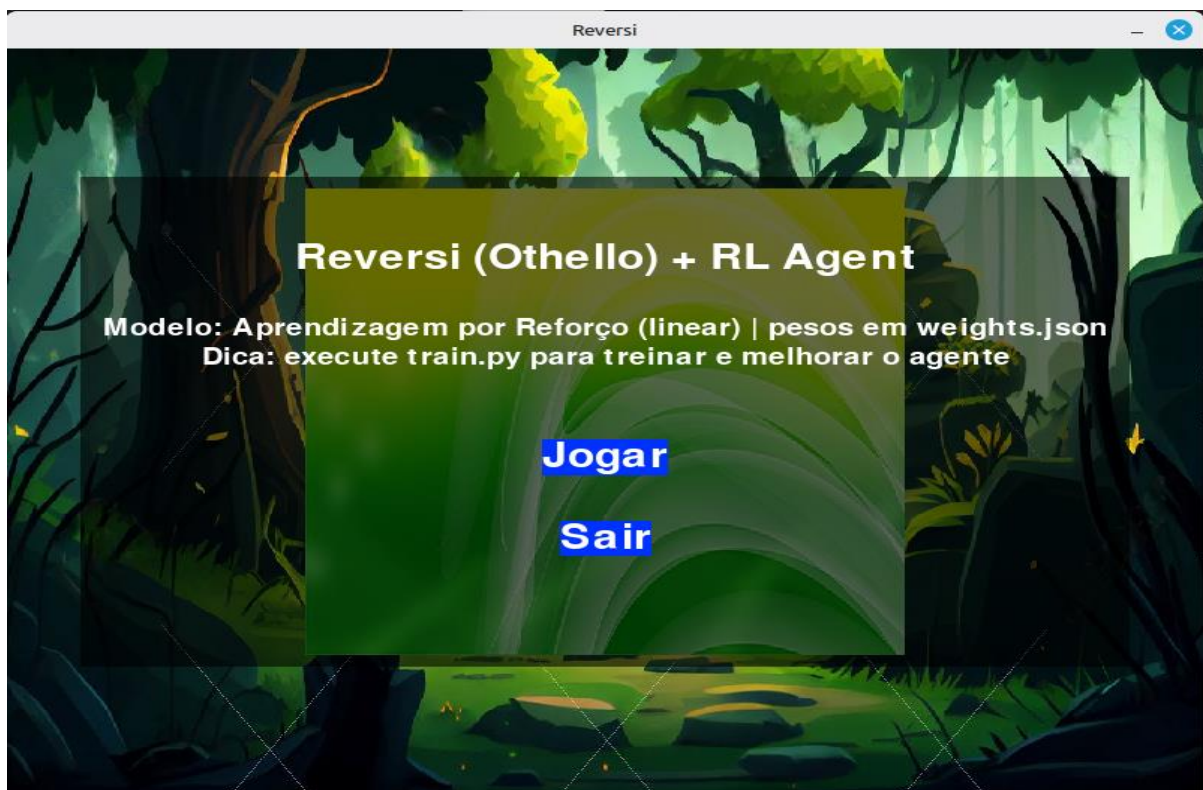


Figura 3 – Interface gráfica do jogo – Menu Inicial

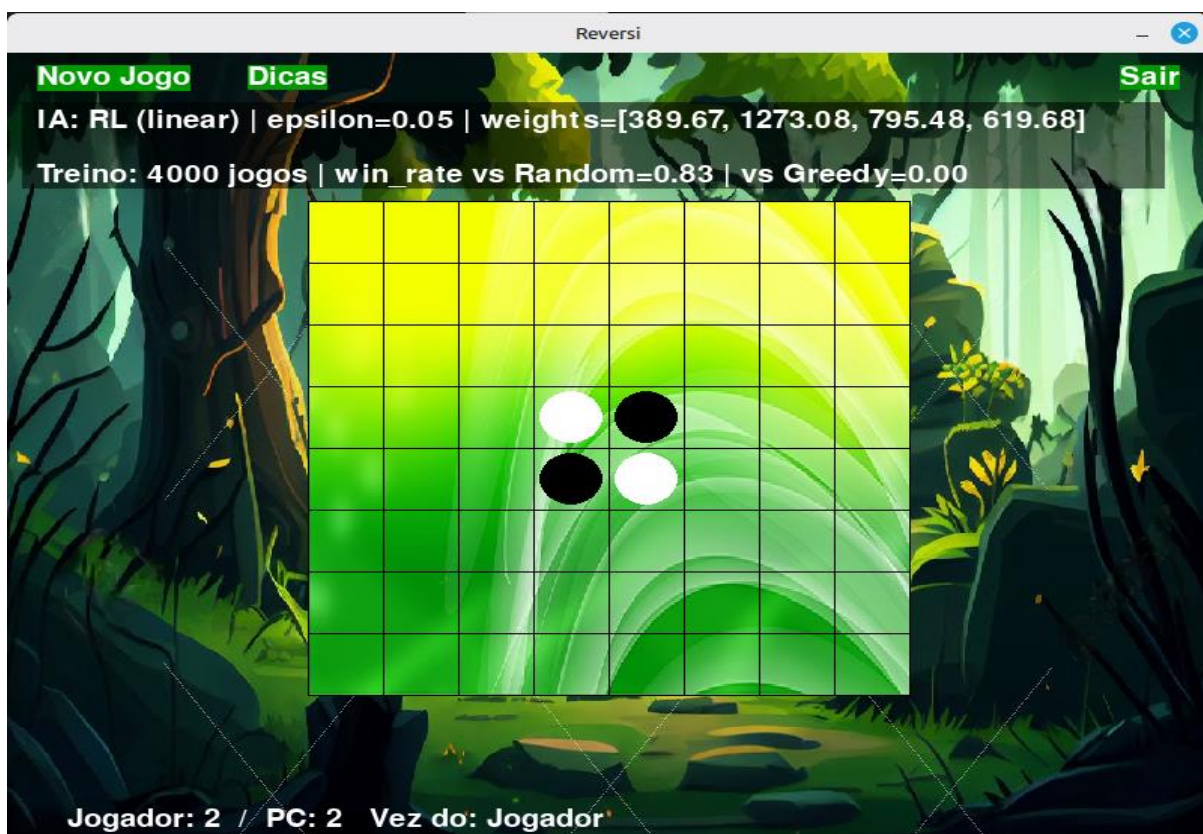


Figura 4 – Interface gráfica do jogo – Tabuleiro modo início

## 8. Discussão

O projeto demonstra a viabilidade da utilização de Aprendizagem por Reforço para o desenvolvimento de agentes inteligentes em jogos de tabuleiro. A abordagem adotada apresenta resultados consistentes, embora limitada pela simplicidade da função de avaliação e pela ausência de diversidade de adversários durante o treino.

## 9. Conclusão

Neste trabalho foi desenvolvido um agente inteligente adaptativo para o jogo Reversi, utilizando técnicas de Aprendizagem por Reforço. O agente demonstrou capacidade de aprendizagem progressiva, melhorando o seu desempenho ao longo do treino e alcançando resultados consistentes contra adversários aleatórios.

A implementação adotou uma abordagem simples e eficiente, adequada às limitações de recursos computacionais disponíveis, privilegiando a clareza do modelo e a interpretabilidade das decisões do agente. A escolha das ferramentas revelou-se apropriada, permitindo a integração do agente num jogo totalmente funcional com interface gráfica.

Em conclusão, o projeto cumpriu os objetivos propostos, validando a aplicação prática dos conceitos de Inteligência Artificial abordados na disciplina e estabelecendo uma base sólida para futuras melhorias.

## 10. Trabalhos Futuros e Melhorias

Apesar dos resultados positivos obtidos, existem várias melhorias que podem ser implementadas para aumentar o desempenho e a robustez do agente inteligente. Uma das principais extensões consiste em treinar o agente contra múltiplos tipos de adversários, incluindo estratégias mais fortes, de forma a melhorar a sua capacidade de generalização.

Outra melhoria relevante seria a utilização de self-play, permitindo que o agente aprenda jogando contraversões anteriores de si próprio, uma abordagem comum em sistemas de Aprendizagem por Reforço mais avançados. Adicionalmente, a substituição da função de avaliação linear por modelos mais expressivos, como redes neurais, poderia permitir a aprendizagem de estratégias mais complexas.

Do ponto de vista experimental, seria igualmente pertinente explorar diferentes conjuntos de características, ajustar hiper parâmetros do treino e analisar o impacto de recompensas intermédias. Por fim, melhorias na interface gráfica e na visualização do processo de aprendizagem poderiam tornar o sistema mais informativo e interativo.

## 12. Referências Bibliográficas

- Ian Millington & John Funge: *Artificial Intelligence for games, R. Learning*.
- Georgios N. Yannakakis and Julian Togelius: *A.I.G., Reinforcement Learning*.
- CS188 – UC Berkeley: Artificial Intelligence.
- Documentação oficial do Pygame.