

**Universidade do Minho — Unidade Curricular Lógica e
Inteligência Artificial**

Trabalho Prático de Prolog

Quem Quer Ser Milionário

Professores: José Machado e Carlos Alves

Data de Entrega: **11 de Janeiro 2026**

1. Objetivo Geral

Desenvolver um jogo interativo em **Prolog** inspirado no formato televisivo “*Quem Quer Ser Milionário?*”, aplicando de forma integrada os conceitos de programação lógica estudados:

- Programação orientada ao padrão;
- Recursividade;
- Manipulação de listas e estruturas compostas;
- Inferência lógica;

O projeto deve evidenciar rigor lógico, clareza estrutural e criatividade, explorando tanto a vertente técnica como estética.

2. Descrição Geral do Projeto

O programa deverá simular o jogo *Quem Quer Ser Milionário?*, contendo:

- 20 perguntas progressivas em dificuldade e valor monetário;
- Patamares de segurança (mínimo dois);
- Três ajudas lógicas (por exemplo: 50/50, ajuda do público, telefone);
- Controlo recursivo do fluxo do jogo;

- Interface textual criativa (ASCII art, cores ANSI, animações simples).

O predicado principal do jogo será:

?- `jogar.`

3. Requisitos Técnicos

3.1 Base de Conhecimento

Cada pergunta deverá ser representada como um predicado que contenha:

- Texto da pergunta;
- Quatro opções (A, B, C, D);
- Resposta correta;
- Nível de dificuldade e valor monetário.

As perguntas deverão ser armazenadas numa lista percorrida recursivamente.

3.2 Mecânica de Jogo

- O jogador começa com €0 e progride mediante respostas corretas;
- Pode desistir ou usar ajudas a qualquer momento;
- Em caso de erro, mantém o último patamar alcançado;
- O ciclo de jogo deverá ser implementado de forma recursiva.

3.3 Ajudas

Deverão existir três ajudas distintas, implementadas como predicados independentes.
Exemplos:

- **50/50:** elimina duas respostas incorretas;
- **Ajuda do público:** simula uma distribuição percentual;
- **Telefone:** fornece uma sugestão com incerteza lógica.

Cada ajuda deve usar inferência lógica e ser reutilizável no código.

3.4 Regras de Inferência

O jogo deverá demonstrar explicitamente os três tipos de raciocínio lógico:

Modus Ponens (raciocínio válido positivo) *Se o jogador respondeu corretamente, então progride para a próxima pergunta.*

```
implica(resposta_certa, progresso).  
verdadeiro(resposta_certa) :- jogador_acertou.  
conclusao(progresso) :-  
    implica(resposta_certa, progresso),  
    verdadeiro(resposta_certa).
```

Modus Tollens (raciocínio válido negativo) *Se o jogador não progrediu, então a resposta não era correta.*

```
implica(resposta_certa, progresso).  
falso(progresso) :- jogador_errou.  
conclusao(nao_resposta_certa) :-  
    implica(resposta_certa, progresso),  
    falso(progresso).
```

Modus Mistaken (falácia intencional) *Se o jogador progrediu, então a resposta era correta.*

```
implica(resposta_certa, progresso).  
verdadeiro(progresso).  
conclusao_incorreta(resposta_certa) :-  
    implica(resposta_certa, progresso),  
    verdadeiro(progresso).
```

Esta regra deve ser comentada e justificada no relatório como exemplo de **raciocínio falacioso**.

4. Requisitos Criativos e Expansões Opcionais

Os grupos que excederem os requisitos mínimos terão valorização na nota final. Recomenda-se a inclusão de:

- **Design ASCII** (logótipos, molduras, barras de progresso);

- **Narrativa temática** (por exemplo: “Milionário Científico”, “Milionário do Futebol”);
- **Modos de jogo adicionais** (treino, rápido, competitivo);
- **Armazenamento de pontuação** ou ranking de jogadores;
- **Som simbólico e efeitos textuais** (pausas, simulações de tempo real).

Projetos criativos e tecnicamente sólidos serão fortemente valorizados.

5. Relatório Técnico e Criativo

O relatório (máx. 12 páginas) deverá acompanhar o código e documentar o desenvolvimento.

5.1 Estrutura sugerida

- **Capa:** Título, unidade curricular, autores, data.
- **Introdução:** Objetivos e enquadramento.
- **Estrutura Lógica:** Descrição dos predicados, recursividade, listas e regras lógicas.
- **Criatividade:** Descrição dos elementos gráficos, interativos e narrativos.
- **Testes e Demonstração:** Execuções de exemplo e comportamentos das ajudas.
- **Conclusão:** Reflexão sobre aprendizagem e melhorias futuras.

6. Entrega

- `milionario.pl` – código principal;
- `perguntas.pl` – base de dados (opcional);
- `README.txt` – instruções de execução;
- `relatorio.pdf` – relatório técnico e criativo.

7. Critérios de Avaliação

Critério	Peso
Funcionamento técnico e ausência de erros	20%
Aplicação correta dos conceitos lógicos (Ponens, Tollens, Mistaken, recursividade, listas)	25%
Estrutura, clareza e documentação do código	15%
Criatividade e grafismo textual	20%
Qualidade e profundidade do relatório	20%

8. Nota Final

O trabalho deve demonstrar domínio da lógica, clareza estrutural e criatividade. Os grupos que fizerem mais do que o exigido — seja pela complexidade lógica, inovação das ajudas ou riqueza estética — terão **majoração na avaliação final**.

“Pensar logicamente é dominar o jogo invisível da razão.

Programar com criatividade é transformar esse jogo em arte.”

Bom Trabalho