

Trabalho 3: Jogo 21 com Java RMI

Objetivo

Desenvolver uma aplicação distribuída utilizando Java RMI (Remote Method Invocation) para simular o jogo de cartas 21 (Blackjack). O projeto deve implementar um modelo cliente-servidor no qual os jogadores interagem remotamente com o servidor do jogo para jogar partidas individuais contra o *dealer* (ou mesa).

Breve descrição das regras do jogo

O Blackjack (ou "21") é um jogo de cartas no qual o objetivo do jogador é alcançar uma pontuação o mais próxima possível de 21, sem ultrapassá-la.

- Cada carta tem um valor:
- Números de 2 a 10 valem seu próprio valor.
- Valetes (J), Damas (Q) e Reis (K) valem 10 pontos cada.
- O Ás (A) pode valer 1 ou 11 pontos, o que for mais vantajoso para a mão, sem que o total ultrapasse 21.

Dinâmica Básica do Jogo:

1. O jogador recebe duas cartas iniciais, ambas com a face para cima.
2. O *dealer* também recebe duas cartas, mas geralmente uma com a face para cima e outra para baixo.
3. O jogador então tem as seguintes ações possíveis:
 - **Pedir carta (Hit):** Solicitar mais uma carta para tentar aumentar sua pontuação.
 - **Manter (Stand):** Não pedir mais cartas e manter a pontuação atual.
4. Se a pontuação do jogador ultrapassar 21, ele "estoura" e perde a rodada imediatamente.
5. Após o jogador decidir "Manter", é a vez do *dealer*. O *dealer* revela sua carta virada para baixo.
6. **Comportamento do Dealer:** O *dealer* segue uma regra fixa: ele deve pedir cartas até que sua pontuação total seja 17 ou mais. Uma vez que atinja 17 ou mais, ele obrigatoriamente para.
7. **Determinação do Vencedor:**

- Se o *dealer* estourar e o jogador não, o jogador vence.
- Se o jogador tiver uma pontuação maior que a do *dealer* (e ambos não tiverem estourado), o jogador vence.
- Caso contrário, o *dealer* vence. Em caso de empate, o *dealer* também vence (nesta versão simplificada).

Como funciona o *dealer*

O *dealer* é um oponente automatizado controlado pelo servidor. Ele segue as regras padrão do Blackjack:

- O *dealer* recebe duas cartas no início da rodada, uma visível e uma oculta.
- Deve comprar cartas até atingir pelo menos 17 pontos.
- Se alcançar 17 ou mais pontos, para automaticamente.
- Se ultrapassar 21 pontos, o *dealer* perde automaticamente.
- Após o *dealer* jogar, o servidor compara sua pontuação com a do jogador e define o vencedor da rodada.

Como funciona o jogador

O jogador interage com a aplicação por meio de um cliente RMI. Seu comportamento é controlado manualmente, com base nas seguintes ações:

- Ao iniciar a rodada, o jogador recebe duas cartas visíveis.
- Pode pedir carta (recebe uma nova do servidor) ou parar (encerra sua jogada).
- Pode pedir quantas cartas quiser, mas se ultrapassar 21 pontos, perde automaticamente.
- Após parar, o *dealer* joga e o servidor informa o resultado.

O jogador não vê as cartas dos outros jogadores. Apenas a primeira carta do *dealer* é visível no início da rodada.

Descrição Geral

O jogo deve seguir as regras descritas acima e permitir múltiplas rodadas individuais entre jogadores e o *dealer*. Cada jogador joga de forma isolada.

Arquitetura Esperada

O sistema deve ser dividido em três partes principais:

1. Servidor RMI: mantém o baralho, *dealer*, gerencia jogadores e lógica do jogo.
2. Interface Remota: define os métodos disponíveis para os clientes.

3. Cliente RMI: permite ao jogador interagir com o servidor e visualizar o resultado.

Funcionalidades mínimas obrigatórias

- Conexão de jogadores por nome.
- Início de rodada com duas cartas para o jogador.
- Ação de "pedir carta" e "parar".
- Lógica do *dealer* automática (parar com 17 ou mais).
- Cálculo e exibição do resultado da rodada.
- Exibição das cartas visíveis do *dealer*.
- Isolamento entre jogadores (jogos individuais contra o *dealer*).

Extras opcionais (1 ponto)

- Registro do histórico de vitórias/derrotas por jogador.
- Interface gráfica (Swing/JavaFX).
- Suporte a múltiplos jogadores simultâneos.
- Implementação de apostas fictícias.

Critérios de Avaliação – Funcionalidade mínimas (9 pontos)

Funcionamento correto da aplicação – 6,0 pontos

Organização do código – 0,5 pontos

Uso correto de Java RMI – 1,0 ponto

Comentários e documentação – 0,5 pontos

Relatório – 1,0 ponto

Entrega

- Código-fonte completo do projeto.
- Instruções de compilação e execução.
- Relatório simples (1–3 páginas) explicando as decisões de implementação.