

# Jogo de Damas Multiplayer (JAVA - TCP/IP)

*Sistema cliente/servidor em Java que permite jogar damas por rede usando sockets TCP. Projeto dividido em camadas: rede, modelo e interface gráfica. O servidor gerencia partidas, valida movimentos e emparelha jogadores; o cliente fornece interface Swing para interação*

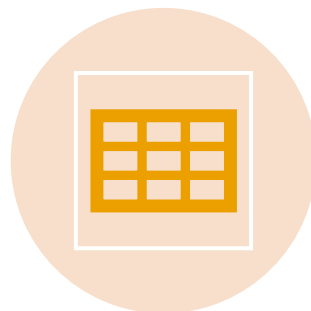
# Arquivos principais

```
src/  
├─ network/  
│   └─ GameProtocol.java      # Protocolo de comunicação  
├─ model/  
│   └─ GameState.java        # Lógica do jogo (validação server-side)  
├─ server/  
│   └─ CheckersServer.java    # Servidor de partidas  
└─ form/  
    └─ TabuleiroForm.java     # Cliente com interface gráfica
```

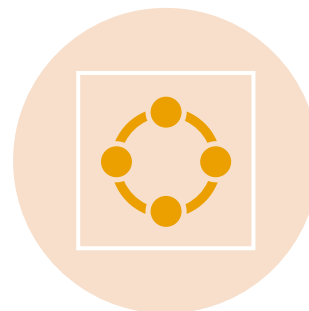
# Arquitetura (visão geral)



CAMADA DE REDE:  
PARSING/MONTAGEM DE  
MENSAGENS TEXTUAIS SIMPLES  
SEPARADAS POR |.



CAMADA DE MODELO: GAMESTATE  
MANTÉM O TABULEIRO, VALIDA  
MOVIMENTOS, APLICA REGRAS DE  
CAPTURA OBRIGATÓRIA E PROMOÇÃO  
A DAMA



CAMADA DE INTERFACE:  
TABULEIROFORM RODA NA EDT  
(SWING); COMUNICAÇÃO COM  
THREADS SEPARADAS PARA I/O.



SERVIDOR: POOL DE THREADS PARA  
ACEITAR CLIENTES; CADA PARTIDA  
RODA EM SUA PRÓPRIA EXECUÇÃO  
COM SINCRONIZAÇÃO DO ESTADO.

---

# Protocolo de comunicação (resumo)

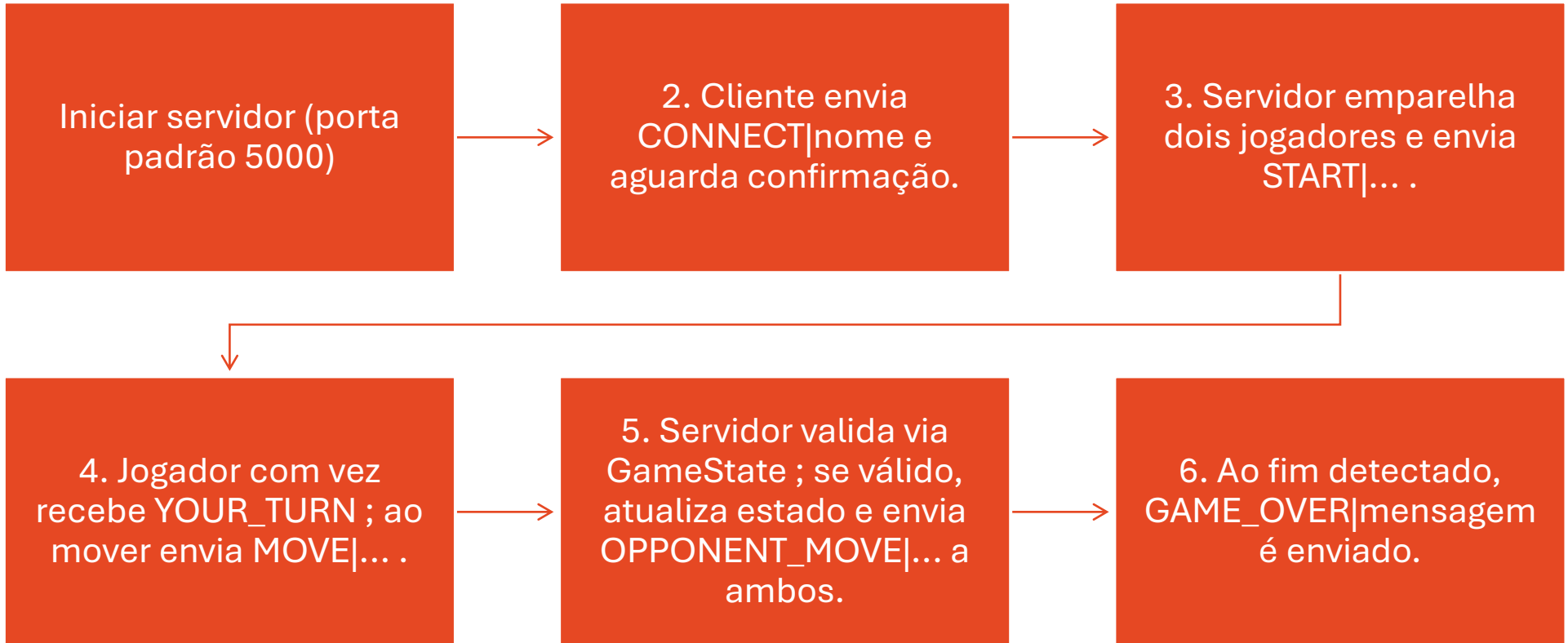
Mensagens –  
Cliente ->  
Servidor

- CONNECT|nomeJogador
- MOVE|r1,c1,r2,c2
- DISCONNECT

Mensagens –  
Servidor ->  
Cliente

- CONNECTED , WAIT\_FOR\_PLAYER
- START|WHITE / START|BLACK
- YOUR\_TURN , MOVE\_INVALID , OPPONENT\_MOVE|r1,c1,r2,c2
- GAME\_OVER|mensagem , ERROR|mensagem

# Fluxo de jogo (passo a passo)



---

# Pontos técnicos relevantes



Sincronização: proteção do estado de cada partida para evitar condições de corrida.



Validação no servidor: regra de captura obrigatória e múltiplas capturas aplicadas no backend.



UI: uso correto da EDT para evitar congelamento; atualização do modelo via `SwingUtilities.invokeLater`.



Escalabilidade: `ThreadPool` para aceitar conexões; cada sala de jogo isolada.

# Como compilar e executar (Windows)



Compilar:

```
.\compile.bat  
ou javac -d bin -encoding UTF-  
8 src\network\*.java  
src\model\*.java  
src\server\*.java src\form\*.java
```



Executar servidor:

```
.\run-server.bat ou java -cp bin  
server.CheckersServer (porta  
5000)
```



Executar cliente:

```
.\run-client.bat ou java -cp bin  
form.TabuleiroForm  
Conectar clientes em  
127.0.0.1:5000 para testes  
locais.
```

---

# Problemas comuns e soluções

01

Address already in use — porta 5000 ocupada: encerrar processo ou alterar a porta.

02

Conexão falhando — verificar firewall, IP e servidor em execução.

03

Jogo não inicia — confirmar que dois clientes estão conectados; checar logs do servidor.

---



---

# Vamos Jogar ?

