



Universidade Federal de Roraima

SERVIDOR DE JOGOS MULTI-THREAD PARA MATCHES (TORNEIO DE BATALHA POR TURNOS)

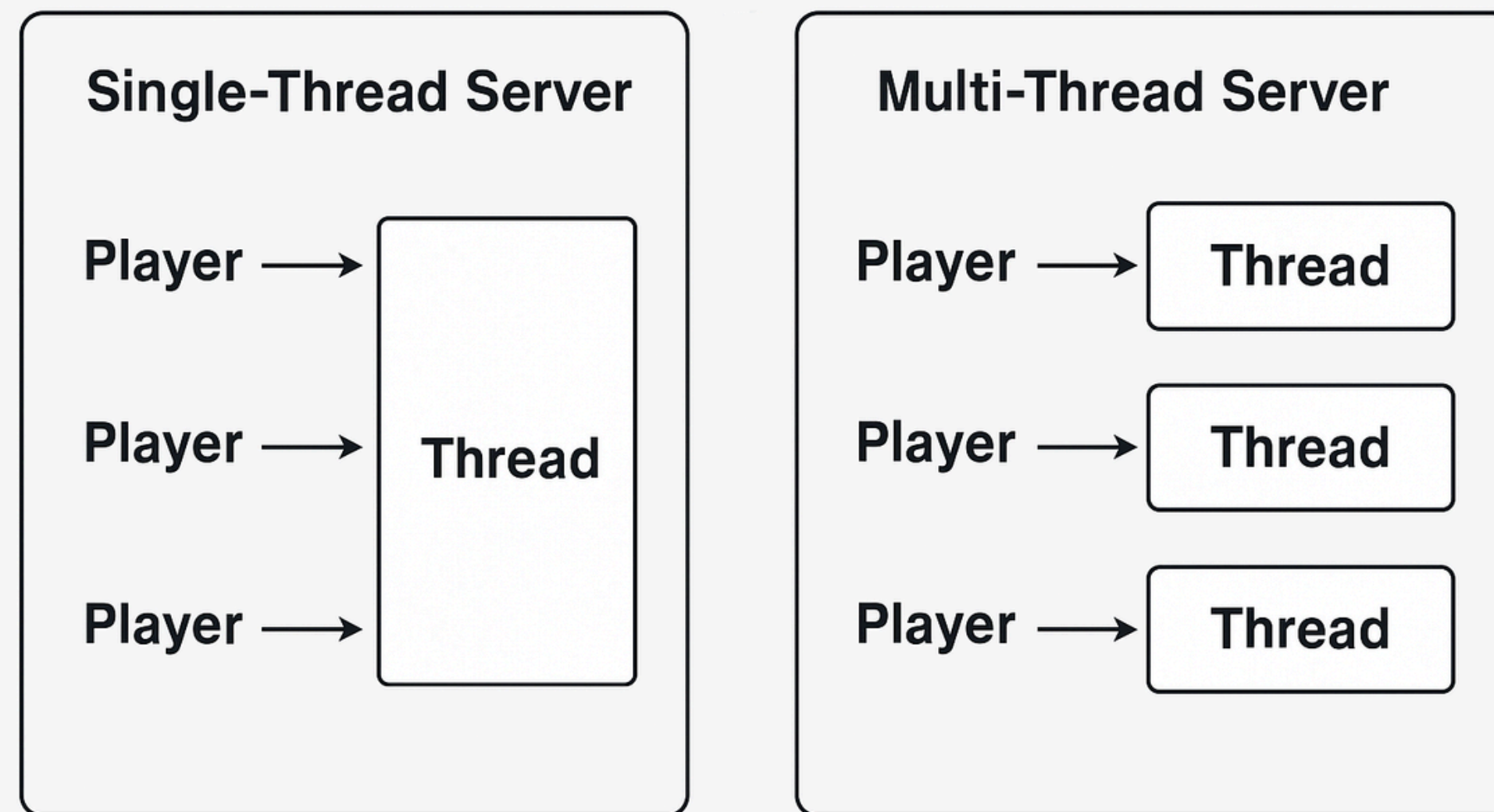
**Filipe Gabriel,
Marcus Vinícius**

Introdução

Contexto: Servidores de jogos geralmente enfrentam desafios ao processar ações em tempo real.

Problema: Servidores single-thread podem criar gargalos e ineficiências.

Objetivo: Implementar um servidor multi-thread para gerenciar jogos de torneio por turnos com desempenho otimizado.

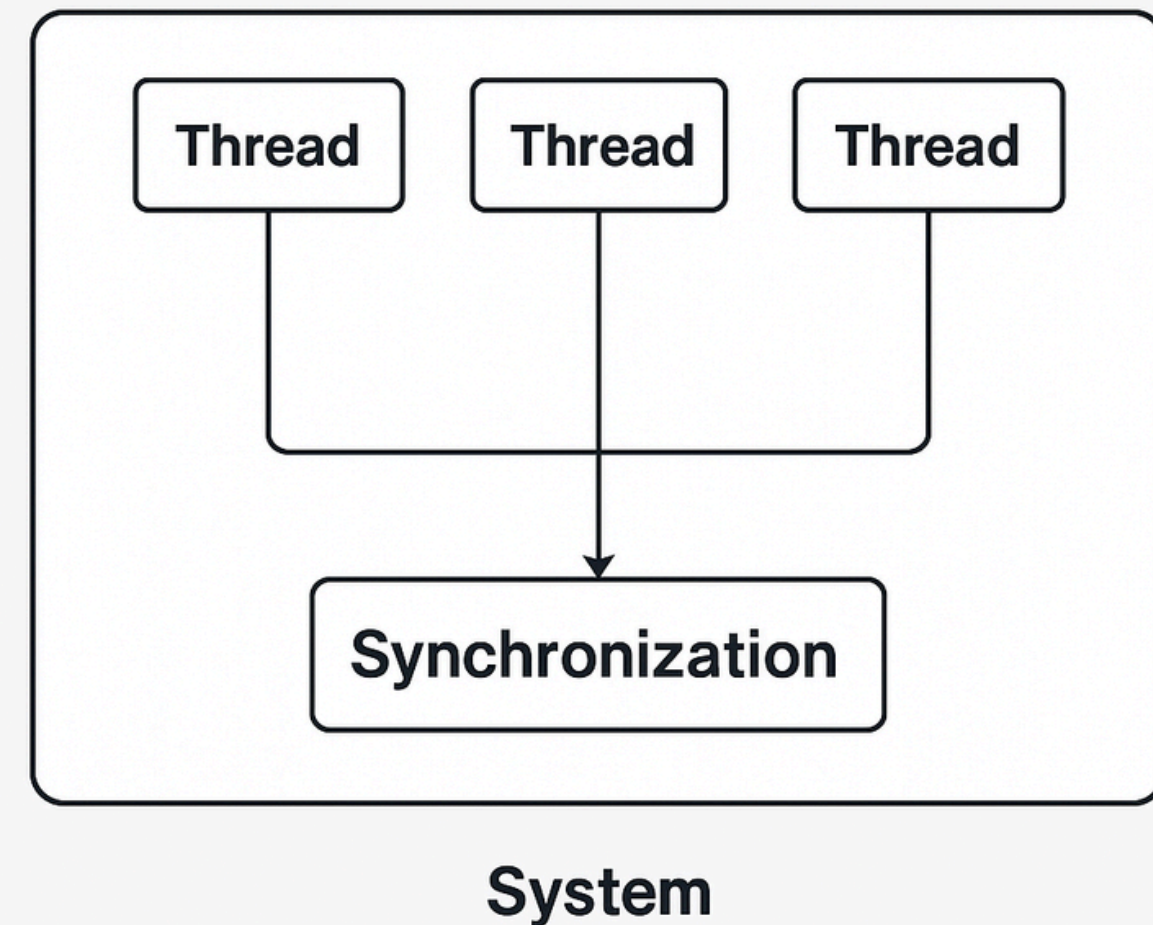


Arquitetura do Projeto

Componentes Principais:

- **Threads:** Worker threads, threads de jogador, thread do torneio.
- **Sincronização:** Uso de mutexes para gerenciar recursos compartilhados.
- **Estruturas de Dados:** Jogador, torneio, fila de eventos.

Architecture, highlighting threads and synchronization



Fluxo Geral:

Conexão → Pareamento → Rodadas de batalha → Campeão.

Funcionalidades

Multi-threaded Server

Gerencia conexões e ações paralelamente.

Sistema de Batalha

- Escolhas de ataque/defesa.
- Dano com chance de crítico e defesa.

Torneio Completo

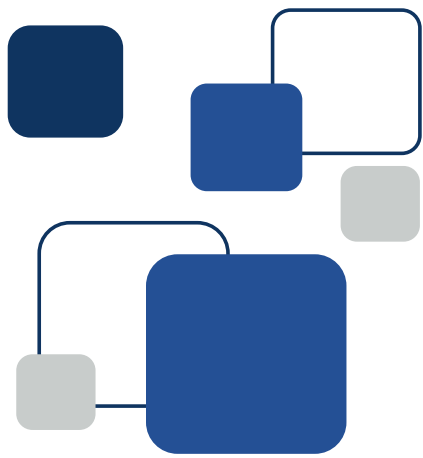
Rodadas eliminatórias até restar um campeão.

Timeout

Tempo limite para conexões de novos jogadores.

Comunicação TCP:

Mensagens em tempo real entre cliente e servidor.



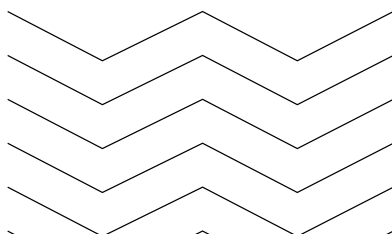
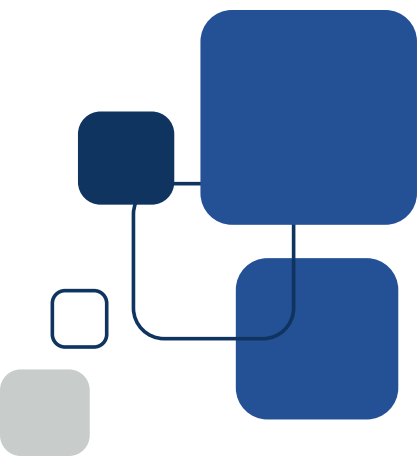
Exemplo de Execução

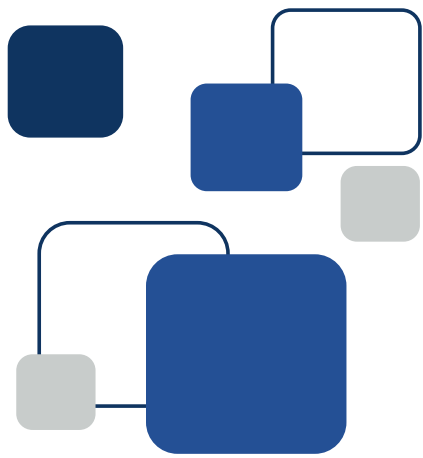
Servidor

Cliente

```
filipe@filipe-Inspiron-3501:~/games_server$ gcc servidor.c -o servidor -lpthread
filipe@filipe-Inspiron-3501:~/games_server$ ./servidor
Aguardando 4 jogadores se conectarem...
Jogador 1 conectado!
Jogador 2 conectado!
Jogador 3 conectado!
Jogador 4 conectado!
Iniciando semifinal 1: Jogador 1 vs Jogador 2
Iniciando semifinal 2: Jogador 3 vs Jogador 4
Confronto finalizado: VENCEDOR: Jogador 3
Confronto finalizado: VENCEDOR: Jogador 2
Iniciando FINAL: Jogador 2 vs Jogador 3
Confronto finalizado: VENCEDOR: Jogador 3
O CAMPEÃO é o Jogador 3
```

```
@filipe-Inspiron-3501:~/games_server$ gcc cliente.c -o cliente
@filipe-Inspiron-3501:~/games_server$ ./cliente
ado ao servidor!
r 1 ataca Jogador 2 causando 1 de dano! HP de 2: 9
r 2 ataca Jogador 1 causando 1 de dano! HP de 1: 9
r 1 ataca Jogador 2 causando 4 de dano! HP de 2: 5
r 2 ataca Jogador 1 causando 4 de dano! HP de 1: 5
r 1 ataca Jogador 2 causando 1 de dano! HP de 2: 4
r 2 ataca Jogador 1 causando 1 de dano! HP de 1: 4
r 1 ataca Jogador 2 causando 3 de dano! HP de 2: 1
r 2 ataca Jogador 1 causando 4 de dano! HP de 1: 0
OR: Jogador 2
or desconectado
conexão.
```

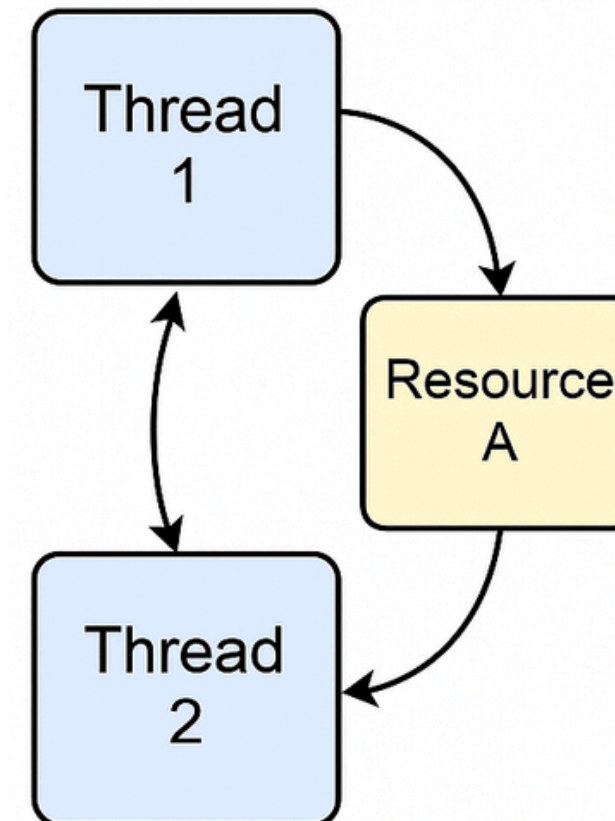




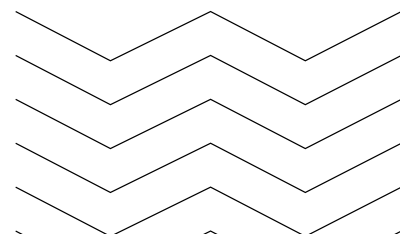
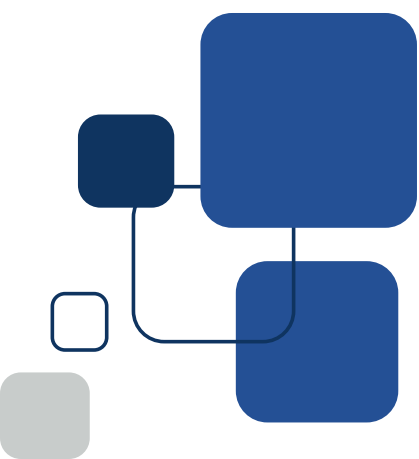
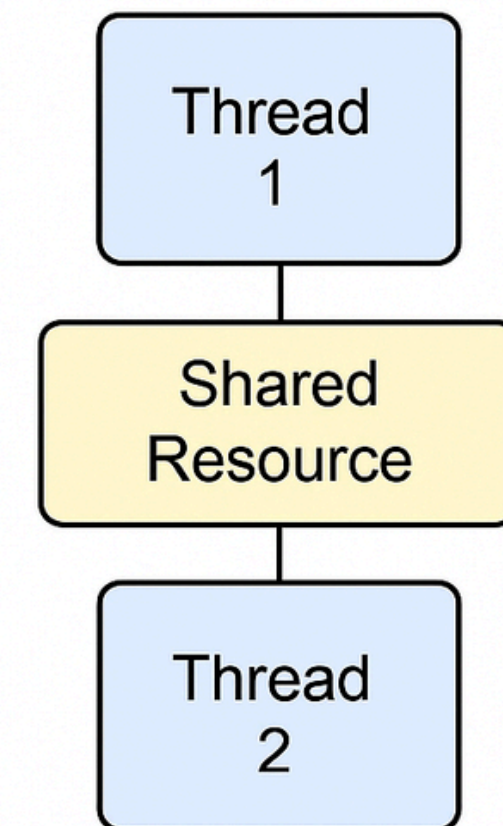
Desafios Enfrentados

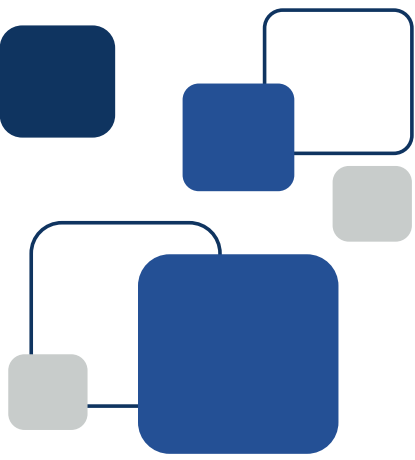
- Sincronização de threads para evitar condições de corrida.
- Gerenciamento de recursos compartilhados (HP, estado do torneio).
- Design modular para suportar expansões futuras.
- Testes em cenários com múltiplos jogadores conectados simultaneamente.

Deadlock



Race Condition





OBRIGADO PELA ATENÇÃO!!!!!!

