

# MiniBit: Implementação de um Sistema de Compartilhamento Cooperativo de Arquivos com Estratégias Distribuídas

## Objetivo

Desenvolver, em grupo, um sistema distribuído inspirado nos princípios do BitTorrent, com foco em:

- **Divisão de arquivos em blocos;**
- **Compartilhamento entre múltiplos peers;**
- Uso de **estratégias reais** de distribuição (rarest first + olho por olho);
- Um **tracker central simples** para descoberta de peers.

## Requisitos funcionais obrigatórios

1. **Divisão de arquivo em blocos:**
  - O arquivo a ser compartilhado deve ser dividido em blocos numerados (block\_0, block\_1, ...).
2. **Peers distribuídos com blocos parciais:**
  - Cada peer inicia com um subconjunto aleatório dos blocos.
  - Quando o peer entrar na rede ele vai receber do tracker um subconjunto aleatório de blocos.
  - Observação: Isso não faz parte do funcionamento do bittorrent, mas vai agilizar a convergência da rede.
3. **Comunicação entre peers (P2P):**
  - Peers devem trocar blocos entre si sem depender de um servidor central de conteúdo.
4. **Tracker central simples:**
  - Um servidor central que, ao ser consultado por um peer, retorna um **subconjunto aleatório de peers disponíveis** (exceto ele mesmo).
  - Se a rede tiver **menos de 5 peers**, o tracker retorna **todos os peers**.
5. **Rarest First:**
  - O algoritmo de seleção de blocos deve priorizar aqueles **menos comuns entre os peers conhecidos**.

#### 6. Olho por olho (tit-for-tat simplificado):

- A cada intervalo de 10 segundos, cada peer deve **desbloquear (unchoke) um novo peer aleatório** entre os que conhece, para iniciar a troca de blocos.
- Se esse novo peer possuir **blocos considerados raros** (ou seja, menos comuns entre os demais peers conhecidos), ele poderá ser promovido a um dos **4 peers prioritários (unchoked fixos)**.
- A cada rodada, o conjunto de peers desbloqueados deve conter no máximo **4 usuários fixos + 1 otimista (optimistic unchoke)**.
- A decisão de manter um peer na lista é baseada em **quantos blocos raros ele possui em relação aos outros**.
- Peers fora dessa lista ficam **choked** (bloqueados para envio).
- **Observação importante:** esta não é a implementação fiel do algoritmo tit-for-tat usado no BitTorrent real, que leva em conta a taxa de upload. Esta versão foi **adaptada para funcionar em um ambiente de simulação local (em uma única máquina), onde medir taxa de envio real seria inviável**.

#### 7. Encerramento controlado:

- O sistema deve permitir que um peer se desligue **somente após reconstruir o arquivo completo**.

### Fora de escopo:

- Arquivo .torrent;
- DHT (Distributed Hash Table);
- Estratégias de choking/unchoking complexas;
- Interface gráfica.

### Entregas obrigatórias

#### 1. Código-fonte comentado, com:

- Scripts para inicializar múltiplos peers e o tracker;
- Execução local ou em rede (Docker ou múltiplos terminais/hosts);
- Logs mostrando o progresso dos blocos.

#### 2. Relatório técnico (PDF) com:

- Diagrama da arquitetura;
- Descrição do protocolo de comunicação (mensagens trocadas, estados);

- Estratégia para rarest first e tit-for-tat;
- Resultados de testes (ex: tempo médio até completar, número de mensagens, etc.);
- Dificuldades enfrentadas;
- Reflexão individual de cada membro (1 parágrafo por integrante).

### 3. Apresentação:

- Demonstração funcional do sistema;
- Explicação técnica;
- Respostas a perguntas da banca.

## Grupo

- Máximo: **4 alunos**.