

Diego Barros Aguirre - 41624866  
Henrique da Silva Alves - 31615961

## O que é o projeto?

O projeto é uma simulação do relógio de Berkeley produzido na linguagem JAVA. Para a comunicação dos processos é utilizado sockets com o protocolo UDP junto com THREADS para a coordenação simultânea de todos os relógios. Todas as operações matemáticas são feitas convertendo as datas em Long.

## Como o programa funciona?

Existem 4 classes:

**SincRelogio:** é a classe principal que definirá qual instância será criada através dos argumentos passados pela linha de comando, -m: master ou -s: slave.

**Slave:** é um servidor UDP que fica executando o tempo todo esperando a comunicação do master para realizar ações. O slave tem o papel de enviar o seu horário local quando solicitado e ajustar sua hora em relação ao master de acordo com a diferença recebida.

Quando um slave é instanciado, uma variável chamada passo é instanciada. Essa variável é a quantidade de tempo em minutos que é definido randomicamente (de 1 a 4) para o acréscimo do relógio local a cada rodada de execução.

**Master:** é o gerenciador de Slaves que realiza os cálculos necessários e garante a consistência de hora de todo o sistema. Essa classe lê o arquivo slaves.txt e pega todos os slaves que deverão ser coordenados. Os dados dos slaves são usados para instanciar objetos ThreadM para cada slave detectado, que responsável pela comunicação a fim de receber todos os dados necessários. Após a definição do delta o relógio local do master é atualizado e é passado os ajustes individuais para cada relógio para garantir a consistência das horas.

Diferente da classe Slave o passo do master é definido como 1 minuto e a cada rodada o relógio é atualizado com esse passo, independente do delta.

**ThreadM:** é um cliente UDP que é responsável pela comunicação do master com os slaves. Ele busca a hora local do slave, calcula a sua diferença com a hora do master e salva como uma variável local. Caso essa diferença for -9223372036854775 significa que não foi possível estabelecer uma comunicação com o slave, então ele deve ser ignorado. A última ação é a de envio do ajuste do relógio local do slave de acordo com a mensagem passada pelo master.