

[br.com.threadsBasico](#): Aqui abordamos apenas o funcionamento de **Threads no seu nascimento com Java**. Então não usamos nada sofisticado como **Executores, Latches e etc...**

O projeto vai te falar sobre **o que são Threads**, como você as inicia e as controla, os **problemas básicos que podem ocorrer ao se utiliza-las** e como você pode fazer **uma Thread conversar com outra**. Novamente, estamos abordando Threads no seu nascimento em Java, então usamos bastante o **Synchronized** e classes que não são tão interessantes quanto as atuais disponíveis.

Aqui tem uma **implementação legal de pesquisa em vários arquivos de maneira paralela**.

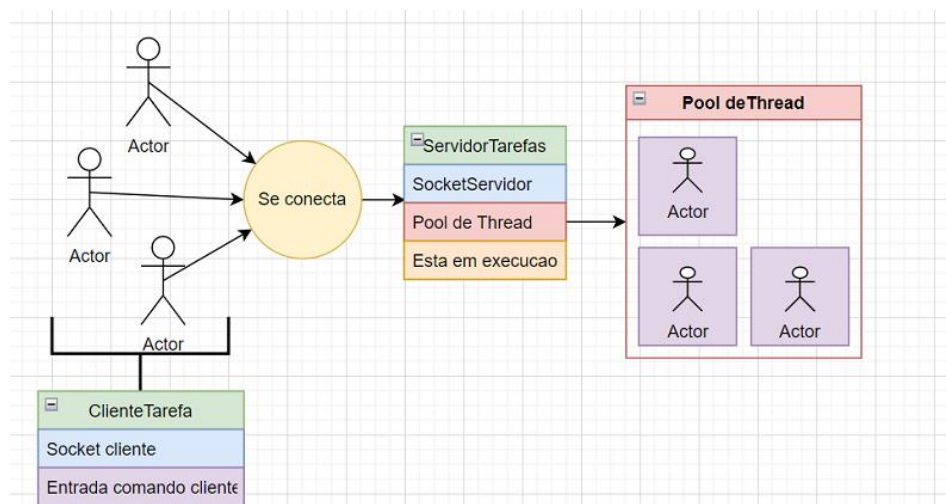
Vale para saber conceitos Básicos, mas a diversão mesmo está no próximo projeto.

[br.com.threadsConcorrente](#): Sabendo conceitos básicos sobre threads, podemos agora elevar o nível da programação concorrente **usando o pacote "Concurrent" do java**, esse pacote traz algumas funcionalidades ao se utilizar Threads. Além de passar um pouco por **Socket**.

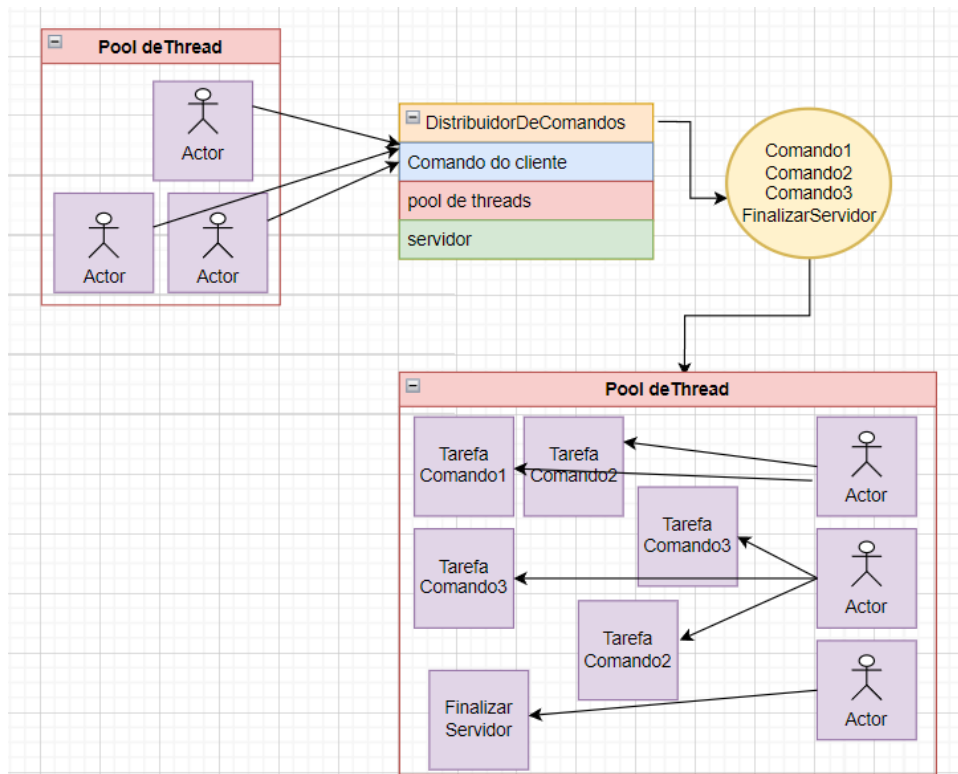
Abordamos sobre o **ExecutorService (um pool de Threads)**, tipos **Atomic (Um atributo que ao ser manipulado por Threads modificara na memória principal e não na memória de cache da Thread)**, uma fila que é **ThreadSafe (BlockingQueue)**, **Callables (Tarefas que retornam algo)**, **Future (objeto que pega o retorno das Callable)** e um pouco de alguns **padrões de projeto**.

Para um melhor entendimento usamos uma aplicação onde as threads serão usadas inúmeras vezes, a aplicação é um projeto onde um **CLIENTE** envia comandos para um **SERVIDOR** executar e o servidor também devolve resposta para o cliente sobre os comandos que ele passou.

Basicamente teremos um **pool de Thread no servidor**, para cada **socket de cliente** o servidor usara uma **Thread** do **pool**:



Cada **Thread** nesse **pool** (cada socket de cliente conectado) irá chamar um distribuidor de comandos, que irá **identificar o comando do cliente** e executa-lo, esses comandos também são rodados em paralelo no nosso pool:



BONUS: AQUI AINDA TEMOS UM EXEMPLO CONTROLADOR DE THREAD **100% FUNCIONAL E CENTRALIZADO** QUE SEPARA AS THREADS POR BLOCOS DE REGISTROS BASEADOS NO ID INICIAL E FINAL!!!