



07. Produzindo e Consumindo com Threads

📅 Date	@08/09/2022
▼ Categoria	Java
▼ Curso	Threads em Java 2: programação concorrente avançada

Tópicos

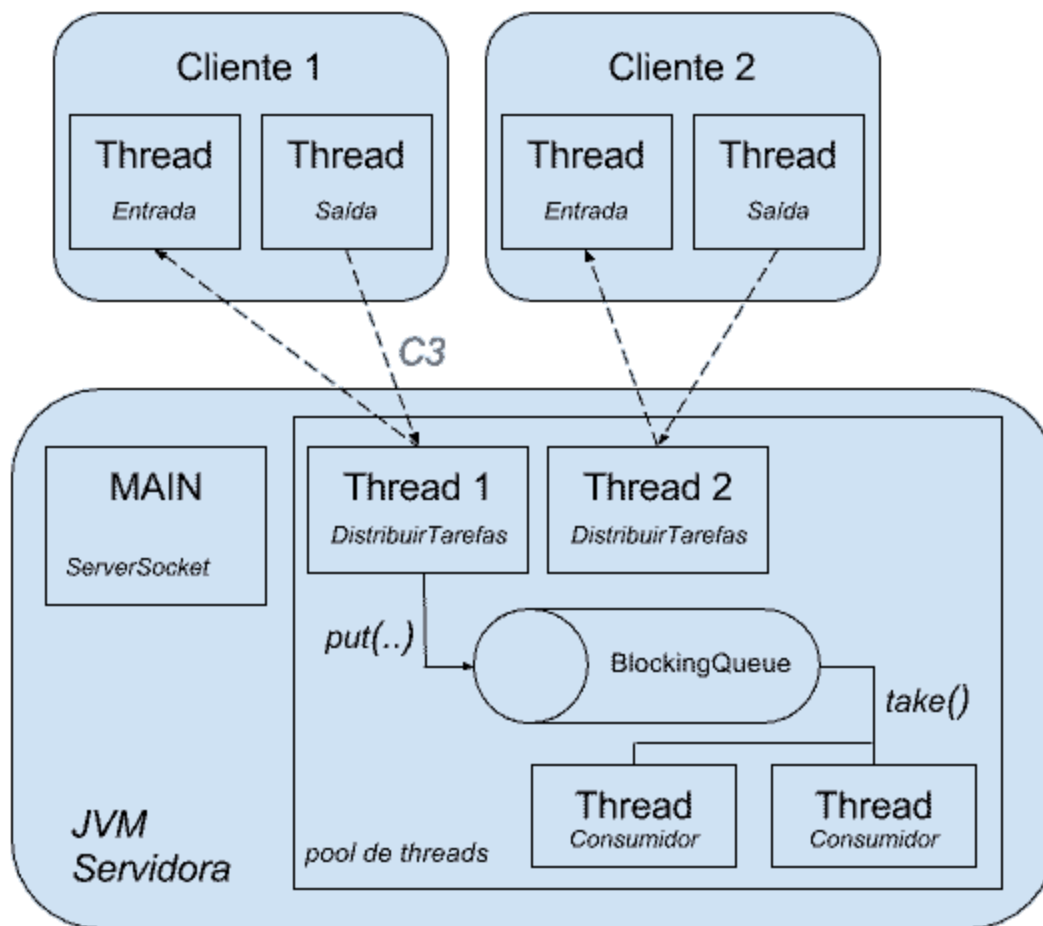
1. Devolvendo
2. Filas
3. Consumindo comandos
4. Conclusão
5. Produzindo e Consumindo com Threads
6. Mãos à obra: Testando BlockingQueue
7. Diferença entre offer() e put()
8. O bloqueio de 3 threads!
9. Mãos à obra: BlockingQueue no Servidor
10. Para saber mais: Processando comandos com ordem natural
11. Download do Projeto Final

Revisão

- Thread e Runnable

- ExecutorService - Pool de Threads
 - CachedThreadPool
 - FixedThreadPool
- thread.join()
- volatile e AtomicBoolean
- UncaughtExceptionHandler
- ThreadFactory
- **Callable e Future**
- BlockingQueue

Produzindo e Consumindo com Threads



O que aprendemos?

- Na API de *Collections* temos a interface `Queue` e suas implementações para representar uma fila.
 - `LinkedList`, além de ser uma lista, também é uma fila.
 - Quando precisamos uma fila que é compartilhada por threads, devemos usar uma implementação da interface `BlockingQueue`.
 - Há várias implementações da interface `BlockingQueue`.
 - `ArrayBlockingQueue` possui um limite de elementos.
 - A interface `BlockingQueue` possui métodos que travam a thread, como `take()` e `put()`.
-

Diferença entre `offer()` e `put()`

Ao tentar adicionar um elemento na fila "sem vaga", o método `put()` bloqueia a execução aguardando por uma vaga dentro da lista. Já o método `offer()` não bloqueia e retorna **false** caso não haja mais espaço na fila. Analogamente há também os métodos `poll()` e `take()` para pegar um elemento da fila. O método `take()` irá esperar a chegada de um novo elemento. Já o método `poll()` permite que seja estabelecido um *Timeout* para a espera de um novo elemento.