07

Mãos à obra: Refatorando o servidor

No projeto `servidor-tarefas` refatore o servidor `ServidorTarefas`, para fechar o recursos corretamente:

1) O servidor deve possuir um atributo `estaRodando`, do tipo `AtomicBoolean`. Além disso, vamos definir o *pool de threads* e o `ServerSocket` como atributos:

```
public class ServidorTarefas {

    //novos atributos
    private ServerSocket servidor;
    private ExecutorService threadPool;
    private AtomicBoolean estaRodando;

    //resto omitido
```

2) Ainda na classe `ServidorTarefas` crie um construtor para inicializar os atributos. Inicialize o `estaRodando` com `true` (`new AtomicBoolean(true)`):

```
    //construtor que inicializa os atributos
    public ServidorTarefas() throws IOException {
        System.out.println("---- Iniciando Servidor ----");
        this.servidor = new ServerSocket(12345);
        this.threadPool = Executors.newCachedThreadPool();
        this.estaRodando = new AtomicBoolean(true);
    }
```

3) Na mesma classe crie um novo método `rodar`, que deve conter todo o código que estava dentro do método `main`, *sem as inicializações* do pool e `ServerSocket`. Ele só deve aceitar novos clientes enquanto `estaRodando` for `true` (`this.estaRodando.get()`):

```
public void rodar() throws IOException {

    while (this.estaRodando.get()) {

        try {
            Socket socket = this.servidor.accept();
            System.out.println("Aceitando novo cliente na porta " + socket.getPort());

            DistribuirTarefas distribuirTarefas = new DistribuirTarefas(socket, this);

            this.threadPool.execute(distribuirTarefas);
        } catch (SocketException e) {
            System.out.println("SocketException, está rodando? " + this.estaRodando);
        }
    }//fim do while
}//fim rodar
```

4) Para fechar os recursos, crie um novo método `parar`, que encerra o `ServerSocket` e o pool de threads, além de atribuir o valor `false` para o atributo `estaRodando` (`this.estaRodando.set(false)`):

```
public void parar() throws IOException {
    System.out.println("Parando servidor");
    this.estaRodando.set(false);
    this.threadPool.shutdown();
    this.servidor.close();
}
```

5) No método `main` crie uma instância de `ServidorTarefas`. Rode o servidor através do método `rodar`:

```
public static void main(String[] args) throws Exception {
    ServidorTarefas servidor = new ServidorTarefas();
    servidor.rodar(); //o servidor roda enquanto estaRodando = true
}
```

6) Para o cliente poder encerrar o servidor crie um novo atributo na classe `DistribuirTarefas`. Adicione um novo parametro no construtor para receber o atributo:

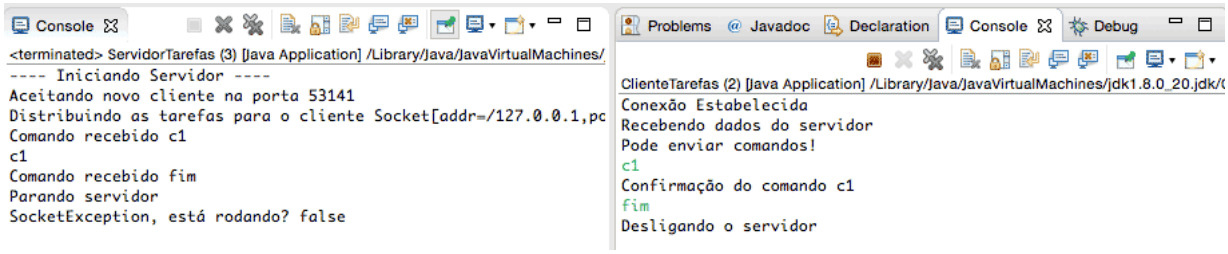
```
public class DistribuirTarefas implements Runnable {

    private Socket socket; //já existe
    private ServidorTarefas servidor; //novo

    public DistribuirTarefas(Socket socket, ServidorTarefas servidor) {
        this.socket = socket;
        this.servidor = servidor; //nova atribuicao
    }
```

7) Para o cliente poder encerrar o servidor, ainda na classe `DistribuirTarefas`, adicione mais um comando no `switch` com o *case* `fim`. Ele imprime uma mensagem para o usuário e chama o método `parar` do `servidor`:

```
//dentro do método run, dentro do switch  (cuidado antes do bloco `default`)
case "fim": {
    saidaCliente.println("Desligando o servidor");
    servidor.parar();
    return; //saindo do do método para liberar a thread
}
```

8) Verifique que nada está rodando ainda e se tudo estiver compilando, teste o código: Rode a classe `ServidorTarefas` e depois a classe `ClienteTarefas`.

9) Tente parar o servidor, enviando o comando `fim` pelo cliente:


Console
<terminated> ServidorTarefas (3) [Java Application] /Library/Java/JavaVirtualMachines/
---- Iniciando Servidor ----
Aceitando novo cliente na porta 53141
Distribuindo as tarefas para o cliente Socket[addr=/127.0.0.1,pc
Comando recebido c1
c1
Comando recebido fim
Parando servidor
SocketException, está rodando? false
Problems
Javadoc
Declaration
Console
Debug
ClienteTarefas (2) [Java Application] /Library/Java/JavaVirtualMachines/jdk1.8.0_20.jdk/
Conexão Estabelecida
Recebendo dados do servidor
Pode enviar comandos!
c1
Confirmação do comando c1
fim
Desligando o servidor