

# Servidor

## Tarefa.java

```
1 package afazeres;
2
3 public class Tarefa {
4     private final int numero;
5     private final String tarefa;
6     public Tarefa(int numero, String tarefa) {
7         this.numero = numero;
8         this.tarefa = tarefa;
9     }
10    public int getNumero() {
11        return numero;
12    }
13    public String getTarefa() {
14        return tarefa;
15    }
16 }
```

## Afazeres.java

```
1 package afazeres;
2
3 import java.net.ServerSocket;
4 import java.net.Socket;
5 import java.util.ArrayList;
6
7 public class Afazeres {
8     private static int contador = 0;
9     private static final ArrayList<Tarefa> tarefas = new ArrayList<>();
10    public static void main(String[] args) {
11        System.out.println("Servidor Afazeres!");
12        try {
13            ServerSocket servidor = new ServerSocket(50000);
14            while (true) {
15                Socket socket = servidor.accept();
16                Thread t = new Thread(new Cliente(socket));
17                t.start();
18            }
19        } catch (Exception e) {
20            System.out.println("Erro: " + e.getMessage());
21        }
22    }
23    public synchronized static void put(String tarefa) {
24        tarefas.add(new Tarefa(++contador, tarefa));
25    }
26    public synchronized static String get(int numero) {
27        for (Tarefa tarefa : tarefas) {
28            if (tarefa.getNumero() == numero) {
29                String afazer = tarefa.getTarefa();
30                tarefas.remove(tarefa);
31                return afazer;
32            }
33        }
34        return null;
35    }
36    public synchronized static String list() {
37        String afazeres = "";
38        for (int i = 0; i < tarefas.size(); i++) {
39            Tarefa tarefa = tarefas.get(i);
40            afazeres += tarefa.getNumero() + ". " + tarefa.getTarefa();
41            if (i < tarefas.size() - 1) afazeres += "\n";
42        }
43        return afazeres.isEmpty() ? null : afazeres;
44    }
45 }
```

## Cliente.java

```
1 package afazeres;
2
3 import java.io.ObjectInputStream;
4 import java.io.ObjectOutputStream;
5 import java.net.Socket;
6
7 public class Cliente implements Runnable {
8     private final Socket socket;
9     public Cliente(Socket socket) {
10         this.socket = socket;
11     }
12    @Override
13    public void run() {
14        try {
15            ObjectOutputStream output = new
16                ObjectOutputStream(socket.getOutputStream());
17            ObjectInputStream input = new
18                ObjectInputStream(socket.getInputStream());
19            String requisicao, resposta;
20            do {
21                requisicao = (String)input.readObject();
22                System.out.println(requisicao);
23                resposta = getRespostaTDP(requisicao);
24                output.writeObject(resposta);
25            } while (!requisicao.equals("EXIT"));
26        } catch (Exception e) {
27            System.out.println("Erro: " + e.getMessage());
28        }
29    }
30    private String getRespostaTDP(String requisicao) { // TDP - ToDo Protocol
31        if (requisicao.startsWith("PUT ") && requisicao.length() > 4) {
32            String tarefa = requisicao.substring(4);
33            Afazeres.put(tarefa);
34            return "OK";
35        } else if (requisicao.startsWith("GET ") && requisicao.length() > 4) {
36            try {
37                int numero = Integer.parseInt(requisicao.substring(4));
38                String tarefa = Afazeres.get(numero);
39                return tarefa == null ? "NOT-FOUND" : numero + ". " + tarefa;
40            } catch (Exception e) {
41                return "ERR";
42            }
43        } else if (requisicao.equals("LIST")) {
44            String afazeres = Afazeres.list();
45            return afazeres == null ? "EMPTY" : afazeres;
46        } else if (requisicao.equals("EXIT")) {
47            return "BYE";
48        } else {
49            return "ERR";
50        }
51    }
52 }
```

# Cliente

## Afazer.java

```
1 package afazer;
2
3 import java.io.ObjectInputStream;
4 import java.io.ObjectOutputStream;
5 import java.net.Socket;
6 import java.util.Scanner;
7
8 public class Afazer {
9     public static void main(String[] args) {
10         Scanner scanner = new Scanner(System.in);
11         try {
12             Socket socket = new Socket(args[0], 50000);
13             System.out.println("Conectado ao servidor Afazer...");
14             ObjectOutputStream output = new
15                 ObjectOutputStream(socket.getOutputStream());
16             ObjectInputStream input = new
17                 ObjectInputStream(socket.getInputStream());
18             while (true) {
19                 System.out.print("> ");
20                 String requisicao = scanner.nextLine();
21                 if (requisicao.trim().isEmpty()) continue;
22                 output.writeObject(requisicao);
23                 String resposta = (String)input.readObject();
24                 if (!resposta.equals("OK")) System.out.println(resposta);
25                 if (requisicao.equals("EXIT")) {
26                     socket.close();
27                     return;
28                 }
29             }
30         } catch (Exception e) {
31             System.out.println("Erro: " + e.getMessage());
32         }
33     }
34 }
```