

LeituraArquivo.java

```
1 package comportamentosComuns;
2
3 import java.io.FileNotFoundException;
4 import java.io.FileReader;
5 import java.io.IOException;
6 import java.util.ArrayList;
7 import java.util.Scanner;
8
9
10 public class LeituraArquivo{
11     static String correlacao;
12     static String tendencia;
13
14     public static String leituraCorrelacao() throws IOException{
15         Scanner scanner = new Scanner(new FileReader("../correlacaoResposta.txt"))
16             .useDelimiter("\\|\\n");
17         while (scanner.hasNext()) {
18             correlacao = scanner.next();
19         }
20         scanner.close();
21         return correlacao;
22     }
23
24     public static String leituraTendencia() throws IOException{
25         //Aqui vou ter que chamar um programa em C, e verificar o arquivo que foi reescrito
26         Scanner scanner = new Scanner(new FileReader("../tendencia.txt"))
27             .useDelimiter("\\|\\n");
28         while (scanner.hasNext()) {
29             tendencia = scanner.next();
30             //System.out.println("MINHA TENDENCIA: "+tendencia);
31         }
32         scanner.close();
33         return tendencia;
34     }
35
36     public static ArrayList<String> leituraFibonacci() throws FileNotFoundException{
37         ArrayList<String> fibonacci = new ArrayList<String>();
```

```

38     Scanner scanner = new Scanner(new FileReader("../FibonacciResposta.txt"));
39
40     fibonacci.add(scanner.next());
41     fibonacci.add(scanner.next());
42     fibonacci.add(scanner.next());
43
44     return fibonacci;
45 }
46
47 public static String leituraMetodo() throws FileNotFoundException {
48     String metodo = new String();
49     Scanner scanner = new Scanner(
49     FileReader("../criterioEntrada.txt")).useDelimiter("\\|\\n");
50
51     while (scanner.hasNext()){
52         metodo = scanner.next();
53     }
54     return metodo;
55 }
56
57 public static double lerAlavancaProlog(){
58     double alavanca;
59     Scanner scanner = null;
60
61     try {
62         scanner = new Scanner(
62     FileReader("../prologResposta.txt")).useDelimiter("\\|\\n");
63     } catch (FileNotFoundException e) {
64         e.printStackTrace();
65     }
66     alavanca = Double.parseDouble(scanner.next());
67     return alavanca;
68 }
69
70 public static long lerTipoGrafico() {
71     long tempoEmMiliSegundos=0;
72     String tipoGrafico = new String();
73     Scanner scanner;
74     try {

```

```

75     scanner = new Scanner(new
FileReader("../criterioEntrada.txt")).useDelimiter("\\|\\n");
76     scanner.nextLine();
77     tipoGrafico = scanner.nextLine();
78     tempoEmMiliSegundos = converteTipoGraficoEmTempo(tipoGrafico);
79 } catch (FileNotFoundException e) {
80     e.printStackTrace();
81 }
82     return tempoEmMiliSegundos;
83 }
84
85     public static long converteTipoGraficoEmTempo(String tipoGrafico) {
86
87         if(tipoGrafico == "M1"){
88             return 60000;
89         }
90         else if (tipoGrafico == "M5") {
91             return 60000*5;
92         }
93         else return 60000*60;
94     }
95
96 }

```

RegistrarNoDF.java

```

1 package comportamentosComuns;
2
3 import jade.core.behaviours.OneShotBehaviour;
4 import jade.domain.DFService;
5 import jade.domain.FIPAException;
6 import jade.domain.FIPAAgentManagement.DFAgentDescription;
7 import jade.domain.FIPAAgentManagement.ServiceDescription;
8
9 public class RegistrarNoDF extends OneShotBehaviour{
10     private static final long serialVersionUID = -5125123631192783579L;
11
12     private String tipo;

```

```

13  private String nome;
14
15  public RegistrarNoDF(String tipo, String nome) {
16      this.tipo = tipo;
17      this.nome = nome;
18  }
19
20  @Override
21  public void action() {
22
23      DFAgentDescription descricaoAgente = new DFAgentDescription();
24      descricaoAgente.setName(myAgent.getAID()); //Registra o nome do agente no DF
25
26      //Criando um servi o
27      ServiceDescription servicoMetodoNumerico = new ServiceDescription();
28      servicoMetodoNumerico.setType(tipo);
29      servicoMetodoNumerico.setName(nome);
30      descricaoAgente.addServices(servicoMetodoNumerico);
31
32      //Registrando o agente no DF
33      try {
34          DFService.register(myAgent, descricaoAgente);
35          System.out.println("Registrado o Agente "+myAgent+" no DF");
36      } catch (FIPAException erro) {
37          erro.printStackTrace();
38      }
39
40
41  }
42
43 }

```

RodarComandos.java

```

1  package comportamentosComuns;
2
3  import java.io.BufferedReader;
4  import java.io.IOException;

```

```

5 import java.io.InputStreamReader;
6
7 public class RodarComandos {
8
9     public static void rodarComandoNoTerminal(String comando) throws IOException{
10         Runtime.getRuntime().exec(comando);
11     }
12 }

```

ConsultorAgente.java

```

1 package metodosNumericos;
2
3 import java.io.FileNotFoundException;
4
5 import comportamentosComuns.LeituraArquivo;
6 import comportamentosComuns.ProcurarCotacoes;
7 import jade.core.Agent;
8
9 public class ConsultorAgente extends Agent{
10     private static final long serialVersionUID = -1528819378039323293L;
11
12     public void setup(){
13         System.out.println("Iniciado o agente consultor");
14         try {
15             // addBehaviour(new ProcurarCotacoes(this,LeituraArquivo.lerTipoGrafico()));
16             addBehaviour(new ProcurarCotacoes(this,6000));
17         } catch (FileNotFoundException e) {
18             e.printStackTrace();
19         }
20
21     }
22 }

```