



Universidade do Minho
Mestrado Integrado em Engenharia Informática
4ºano - 1º Semestre

Inteligência Ambiente : Tecnologias e Aplicações

Questão de Aula nº 1



a83732 – Gonçalo Rodrigues Pinto

Dezembro de 2020

Conteúdo

1	Introdução	3
2	Descrição do Problema	4
3	Descrição do Trabalho	5
3.1	Representação dos Dados	5
3.1.1	Informação	5
3.1.2	Estatísticas	6
3.1.3	Estruturas utilizadas	6
3.2	Algoritmo Implementado	7
3.2.1	1º Etapa - Criação dos ficheiros	7
3.2.2	2º Etapa - Leitura da Informação	7
3.2.3	3º Etapa - Armazenamento e Processamento da informação	8
3.2.4	4º Etapa - Escrita em Ficheiros	9
3.3	Ficheiros obtidos	10
4	Conclusão	11

Lista de Figuras

1	Ficheiro log.html	10
2	Ficheiro logSeason.html	11
3	Ficheiro logEstatisticas.html	11

1 Introdução

No 2º semestre do 3º ano do Curso de Engenharia Informática da Universidade do Minho, existe uma unidade curricular denominada por Inteligência Ambiente Tecnologias e Aplicações, que tem como objectivo introduzir aos estudantes conhecimentos sobre as potencialidades da Inteligência Ambiente, sua caracterização, bem como das tecnologias e metodologias de resolução problemas, mais comuns nesta área, sendo estes capazes de as contextualizar, aplicar e analisar.

O presente trabalho teve como principal objectivo sensibilizar e motivar os alunos para a concepção e aplicação de sistemas baseados em regras (SBR) no contexto de ambientes inteligentes capazes de reconhecer e interpretar dados de sensores.

Este presente trabalho tem como tema **Rule-based Automotive Control System (ACS)** que controla, gere e regula o comportamento de diversos dispositivos e sub-sistemas no veículo. Este assume diversos comportamentos de acordo com dados sensoriais. Deste modo, pretendeu-se desenvolver de um sistema baseado em regras (SBR), tendo como base os dados do dataset fornecido, numa linguagem de programação à escolha.

2 Descrição do Problema

Neste trabalho o *dataset* disponibilizado foi pré-processado, tendo sido substituído todas as vírgulas presentes por um ponto de vírgula de forma a ser possível visualizar os diversos campos no *Excel* como colunas. Este *dataset* tem a seguinte estrutura:

1. Data ISO
2. Nome da Cidade
3. Latitude
4. Longitude
5. Temperatura (em graus Celsius)
6. Humidade (percentagem)

Com base nos dados sensoriais pretende-se controlar o Ar Condicionado, de acordo com valores de conforto de temperatura, nomeadamente 15°C Inverno+Outono e 25°C Primavera+Verão, o SBR construído tem o seguinte comportamento:

- Verifica a estação do ano através da data, de modo a identificar a temperatura de conforto adequada;
- Efectua o output para um ficheiro *log* para posterior interpretação do ACS, no seguinte formato:
 - Se a temperatura for inferior, faz output para o ficheiro do seguinte texto:

*airconditioning+{temperatura °C}**

- Se a temperatura for superior, faz output para o ficheiro do seguinte texto:

*airconditioning-{temperatura °C}**

- Se a temperatura for igual, faz output para o ficheiro do seguinte texto:

*airconditioning={temperatura °C}**

* °C até atingir a temperatura de conforto, quer seja aumentar ou diminuir.

3 Descrição do Trabalho

3.1 Representação dos Dados

O sistema baseado em regras desenvolvido caracteriza valores de conforto de temperatura e como se pretende desenvolver um SBR partindo do dataset fornecido foi necessário representar a informação presente.

Desta forma, criou-se um programa em **Java** disponível que permitiu importar os dados relativos à temperatura. Este sistema efectua a leitura das diversas linhas do *dataset*, processa essa informação e posteriormente escreve nos ficheiros para posterior interpretação.

O universo de discurso considerado para o trabalho prático é o informações meteorológicas (temperatura e humidade). Com intuito de clarificar e restringir alguns significados, considerou-se no âmbito deste trabalho as seguintes definições:

- Informação - possui a informação (apresentada no capítulo anterior) presente no *dataset* disponibilizado;
- Estatísticas - representa as maiores e menores informações meteorológicas como também maior e menores diferenças encontradas dependendo da estação do ano presente no *dataset* que foi facultado;

3.1.1 Informação

Uma informação é caracterizada por um data, nome da cidade, latitude, longitude, temperatura e humidade

```
public class Informacao {  
    private Date date;  
    private String city_name;  
    private double latitude;  
    private double longitude;  
    private double temperature;  
    private int humidity;  
}
```

Assim, podemos constatar que com os métodos apropriados facilmente conseguimos armazenar e obter os vários atributos desta classe.

3.1.2 Estatísticas

Uma estatística é caracterizada por maior/menor temperatura/humidade/diferença para Primavera e Verão como Inverno e Outono além de cada um destes parâmetros existe uma data associada a cada um de forma a registar tal acontecimento:

```
public class Estatisticas {
    private Double maior temperatura Primavera Verao;
    private String data maior temperatura Primavera Verao;
    private Double menor temperatura Primavera Verao;
    private String data menor temperatura Primavera Verao;
    private Double maior temperatura Inverno Outono;
    private String data maior temperatura Inverno Outono;
    private Double menor temperatura Inverno Outono;
    private String data menor temperatura Inverno Outono;
    private Double maior diferenca Primavera Verao;
    private String data maior diferenca Primavera Verao;
    private Double menor diferenca Primavera Verao;
    private String data menor diferenca Primavera Verao;
    private Double maior diferenca Inverno Outono;
    private String data maior diferenca Inverno Outono;
    private Double menor diferenca Inverno Outono;
    private String data menor diferenca Inverno Outono;
    private Integer maior humidade Primavera Verao;
    private String data maior humidade Primavera Verao;
    private Integer menor humidade Primavera Verao;
    private String data menor humidade Primavera Verao;
    private Integer maior humidade Inverno Outono;
    private String data maior humidade Inverno Outono;
    private Integer menor humidade Inverno Outono;
    private String data menor humidade Inverno Outono;
}
```

Assim, podemos constatar que com os métodos apropriados facilmente conseguimos substituir e guardar os vários atributos desta classe .

3.1.3 Estruturas utilizadas

De forma a garantir que não existe informações utilizou-se as estruturas que a linguagem utilizada facilita, a escolha recaiu então guardar num *Set* pois este apenas permite adicionar elementos que sejam diferentes. Para armazenar a estatística de cada cidade criou-se um colecção de cidade com a sua estatística.

```
public class Main {
    private static final Set<Informacao> informacaoSet = new HashSet<>();
    private static final Map<String, Estatisticas> estatisticasMap = new
        HashMap<>();
}
```

Desta forma garante-se que a partir da primeira estrutura não existe repetições e a segunda para cada cidade obtém-se os valores maiores e menores presentes no *dataset*.

3.2 Algoritmo Implementado

De forma a desenvolver o sistema baseado em regras pretendido baseado nas estruturas e classes criadas foi possível desenvolver um algoritmo que efectua a leitura do dataset, guarda em estruturas e posteriormente escreve em ficheiro. Como foi apelado à originalidade criou-se além do que foi pedido mais dois ficheiros, um deles com a informação pedida e escrita no primeiro mas com mais detalhes, por exemplo data e estação do ano, o outro ficheiro criado foi a representação das estatísticas de cada cidade presente no *dataset*.

- **Ficheiro log.html** : Output pretendido separado por parágrafos.
- **Ficheiro logSeason.html** : Tabela em cada linha possui a data de registo, estação do ano e output pretendido.
- **Ficheiro logEstatisticas.html** : É apresentado para cada cidade encontrada as estatísticas previamente apresentadas com o valor registado como também data respectiva.

3.2.1 1ª Etapa - Criação dos ficheiros

Numa primeira fase foi declarado os ficheiros onde pretende-se armazenar a informação.

```
File log = new File("log.html");  
File logSeason = new File("logSeason.html");  
File logEstatisticas = new File("logEstatisticas.html");
```

O tipo de ficheiros escolhido foi HTML pois permite serem executados em qualquer browser de Internet e assim é possível gerar uma página WEB estática o que torna a visualização da informação mais apelativa dados dada a quantidade gigantesca de recursos que fornece, por exemplo tabelas, cabeçalhos, etc. É de salientar que caso os ficheiros já existem, o sistema não gera novos.

3.2.2 2ª Etapa - Leitura da Informação

Após criado os ficheiros implementou-se a leitura e análise de cada linha do *dataset*, ou seja, indicando a localização do ficheiro o sistema vai ler cada linha do ficheiro efectua a separação dos campos pelo ponto e vírgula, se for a primeira linha o sistema não efectua qualquer operação pois esta apenas representa o cabeçalho da informação como também se na separação dos dados não se obter 6 campos (os apresentados previamente) o sistema também não executa qualquer operação.


```

String cvsSplitBy = ";";
BufferedReader br = new BufferedReader(new FileReader(csvFile));
while ((line = br.readLine()) != null) {
    String[] dados = line.split(cvsSplitBy);
    if (i != 0) {
        if (dados.length == 6) {
            ...
        }
    }
    i++;
}

```

3.2.3 3ª Etapa - Armazenamento e Processamento da informação

Posteriormente, após ser lido a informação e esta ser válida é construída uma classe Informação, apresentada previamente, com os dados que foram analisados pois esta possui um método desenvolvido que obtém a estação mediante o atributo "date" obtém o mês e verifica se este encontra-se compreendido entre 3 e 9, o que equivale ao mês de Janeiro a Março(exclusive) e de Setembro(inclusive) a Dezembro, caso esteja devolve a palavra "Inverno+Outono" caso contrário devolve "Primavera+Verão", o que equivale à estação do ano da presente data obtida.

De seguida caso o método tenha sido obtido com sucesso é estabelecido a temperatura de conforto mediante a estação recebida. Finalmente é efectuado o teste se a presente informação existe ou não na estrutura criada para o efeito caso não exista é calculado a diferença após uma verificação se a temperatura de conforto é superior / inferior / igual à temperatura do registo pois pretende-se diferentes tipos de output mediante esta verificação e diferença.

```

Date date = new SimpleDateFormat("dd/MM/yyyy HH:mm", Locale.ENGLISH).parse(
    dados[0]);
Informacao informacao = new Informacao(date, dados[1], Double.parseDouble(
    dados[2]), Double.parseDouble(dados[3]), Double.parseDouble(dados[4]),
    Integer.parseInt(dados[5]));
String estacao = informacao.obterEstacao();
if (estacao.equals("Primavera+Verão")) {
    conforto = 25;
}
if (estacao.equals("Inverno+Outono")) {
    conforto = 15;
}
if (conforto != 0 && !(estacao.equals(""))) {
    if (informacaoSet.add(informacao)) {
        if (informacao.getTemperature() > conforto) {
            diferenca = informacao.getTemperature() - conforto;
            ...
        }
        else if ...
        else ...
    }
}

```

Após o cálculo da diferença, verificou-se se a cidade actual da actual informação existe na colecção criada para armazenar as estatísticas caso não exista mediante a estação do ano presente é criada uma classe Estatistica como foi apresentada previamente com os construtores devidos, caso exista é obtido a estatística dessa cidade e é então verificado um conjunto de condições de forma a averiguar se a temperatura, diferença e humidade encontrada se encontra-se em algum extremo pretendido caso se encontre o valor da estatística dessa cidade como respectiva data são alterados e posteriormente essa estatística é substituída na colecção

```

if (estatisticasMap.containsKey(informacao.getCity_name())) {
    Estatisticas estatisticas = estatisticasMap.get(informacao.getCity_name());
    if (estacao.equals("Inverno+Outono")) {
        if (estatisticas.getMaiortemperaturaInvernoOutono() == null) {
            estatisticas.setMaiortemperaturaInvernoOutono(informacao.getTemperature());
            estatisticas.setDatemaiortemperaturaInvernoOutono(informacao.getDate());
        }
        ...
        if (estatisticas.getMaiortemperaturaInvernoOutono() < informacao.getTemperature()) {
            estatisticas.setMaiortemperaturaInvernoOutono(informacao.getTemperature());
            estatisticas.setDatemaiortemperaturaInvernoOutono(informacao.getDate());
        }
        ...
    }
    ...
    estatisticasMap.replace(informacao.getCity_name(), estatisticas);
}
else {
    if (estacao.equals("Inverno+Outono")) {
        Estatisticas estatisticas = new Estatisticas(null, null, informacao.getTemperature(), informacao.getTemperature(), null, null, diferenca, diferenca, null, null, informacao.getHumidity(), informacao.getHumidity());
        estatisticasMap.put(informacao.getCity_name(), estatisticas);
    }
    ...
}

```

3.2.4 4^o Etapa - Escrita em Ficheiros

Finalmente após a leitura, armazenamento e processamento da informação foi necessário escrever os valores obtidos. Para tal e como foi referido o tipo de ficheiros foi HTML logo foi necessário efectuar algumas declarações iniciais como por exemplo tipo de ficheiro, encoding, estilos, título, cabeçalhos etc mesmo antes da leitura. Contudo apresento apenas a informação relevante que é escrita em ficheiro.

```
// Formatar a diferen a a duas casas decimais
DecimalFormat df = new DecimalFormat("#.00",DecimalFormatSymbols.getInstance
(Locale.US));

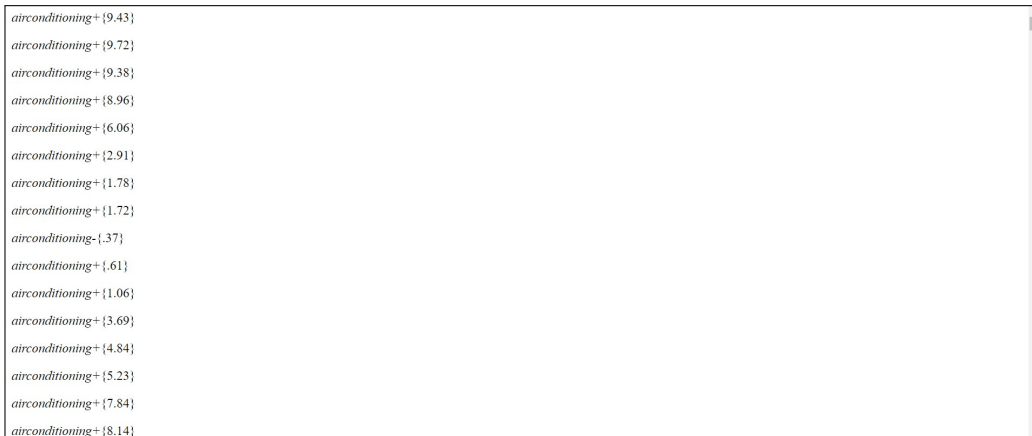
// Escrita no ficheiro log.html
frLog.write("<p> <i>airconditioning-</i>{" + df.format(diferenca) + "}</p>\n
");

// Escrita no ficheiro logSeason.html
frLogSeason.write("<tr>\n" + "      <td>" + informacao.getDate() + "</td>\n" +
"      <td>" + estacao + "</td>\n" + "      <td> <i>airconditioning-</i>{"
+ df.format(diferenca) + "}</td>\n" + "    </tr>");
```

Após a leitura, armazenamento e processamento da informação toda, foi necessário percorrer a colecção que associava a cada cidade uma estatística de forma a guardar essa informação desse ficheiro, para tal necessitou-se de reescrever o método toString da classe estatística de forma a obter uma resposta na sintaxe HTML.

```
frLogEstatisticas.write("<h4 style=\"text-align:center\"> Estat sticas
para as cidades encontradas</h4>\n");
frLogEstatisticas.write("<ul>\n");
for (HashMap.Entry<String, Estatisticas> entry : estatisticasMap.
entrySet()) {
    frLogEstatisticas.write("<li><b>" + entry.getKey() + "</b>");
    frLogEstatisticas.write(entry.getValue().toString());
    frLogEstatisticas.write("</li> \n");
}
frLogEstatisticas.write("</ul>\n");
```

3.3 Ficheiros obtidos



```
airconditioning+{9.43}
airconditioning+{9.72}
airconditioning+{9.38}
airconditioning+{8.96}
airconditioning+{6.06}
airconditioning+{2.91}
airconditioning+{1.78}
airconditioning+{1.72}
airconditioning-{-37}
airconditioning+{.61}
airconditioning+{1.06}
airconditioning+{3.69}
airconditioning+{4.84}
airconditioning+{5.23}
airconditioning+{7.84}
airconditioning+{8.14}
```

Figura 1: Ficheiro log.html

Valores de conforto de temperatura considerados		
<ul style="list-style-type: none"> Inverno+Outono : 15°C Primavera+Verão : 25°C 		
DT_ISO	Estação do Ano	Ação a executar
01/01/2012 00:00	Inverno+Outono	<i>airconditioning+</i> {5.12}
01/01/2012 01:00	Inverno+Outono	<i>airconditioning+</i> {5.14}
01/01/2012 02:00	Inverno+Outono	<i>airconditioning+</i> {5.17}
01/01/2012 03:00	Inverno+Outono	<i>airconditioning+</i> {4.98}
01/01/2012 04:00	Inverno+Outono	<i>airconditioning+</i> {5.17}
01/01/2012 05:00	Inverno+Outono	<i>airconditioning+</i> {5.25}
01/01/2012 06:00	Inverno+Outono	<i>airconditioning+</i> {5.48}
01/01/2012 07:00	Inverno+Outono	<i>airconditioning+</i> {5.41}
01/01/2012 08:00	Inverno+Outono	<i>airconditioning+</i> {5.53}
01/01/2012 09:00	Inverno+Outono	<i>airconditioning+</i> {5.15}
01/01/2012 10:00	Inverno+Outono	<i>airconditioning+</i> {4.47}
01/01/2012 11:00	Inverno+Outono	<i>airconditioning+</i> {3.41}
01/01/2012 12:00	Inverno+Outono	<i>airconditioning+</i> {2.93}
01/01/2012 13:00	Inverno+Outono	<i>airconditioning+</i> {2.68}
01/01/2012 14:00	Inverno+Outono	<i>airconditioning+</i> {2.64}

Figura 2: Ficheiro logSeason.html

Valores de conforto de temperatura considerados	
<ul style="list-style-type: none"> Inverno+Outono : 15°C Primavera+Verão : 25°C 	
Estatísticas para as cidades encontradas	
<ul style="list-style-type: none"> Guimaraes <ul style="list-style-type: none"> maior temperatura Primavera+Verão = 35.72 : 03/08/2018 14:00 menor temperatura Primavera+Verão = 1.77 : 01/04/2016 06:00 maior temperatura Inverno+Outono = 31.91 : 15/10/2017 13:00 menor temperatura Inverno+Outono = -2.12 : 03/02/2012 07:00 maior diferença Primavera+Verão = 23.23 : 01/04/2016 06:00 menor diferença Primavera+Verão = 0.0 : 31/08/2012 16:00 maior diferença Inverno+Outono = 17.12 : 03/02/2012 07:00 menor diferença Inverno+Outono = 0.0 : 15/03/2012 18:00 maior humidade Primavera+Verão = 100 : 17/05/2014 04:00 menor humidade Primavera+Verão = 11 : 17/05/2014 04:00 maior humidade Inverno+Outono = 100 : 02/10/2012 01:00 menor humidade Inverno+Outono = 10 : 07/10/2017 13:00 Vila Real 	

Figura 3: Ficheiro logEstatisticas.html

4 Conclusão

O presente relatório descreveu, de forma sucinta, o desenvolvimento de um sistema baseado em regras (SBR), tendo como base os dados de um dataset fornecido, na linguagem de programação JAVA.

Após a realização deste trabalho, fiquei consciente das potencialidades que um sistema baseado em regras (SBR) pode ter no contexto de ambientes inteligentes capazes de reconhecer e interpretar dados de sensores.

Considero que os principais objectivos foram cumpridos.

Senti também que a realização deste trabalho prático consolidou os meus conhecimentos nas tecnologias e aplicações no âmbito da Inteligência Ambiente.