

Rafael de Souza Trindade - 21350182  
 Thiago Lopes Costa - 21601238  
 Professor Pós-Doc Edjair de Souza Mota  
 Avaliação de desempenho  
 18 de junho de 2019

### Atividade Prática I - Questão 9: Relatório de Horários de Contatos

A atividade consistiu na implementação 2 reports no The ONE. O primeiro devia colocar na saída o número de contatos separados por intervalos de hora. Para isso, considerando que o tempo 0s da simulação corresponde a 00:00 de um dia. Assim, quando o tempo da simulação fosse 86400s, então o tempo seria novamente 00:00 de um outro dia. A saída do report deveria ser semelhante ao exemplo abaixo, com intervalo de horário na primeira coluna e número de contatos na segunda.

```
00:00-00:59 50
01:00-01:59 100
02:00-02:59 85
...
23:00-23:59 90
```

O segundo report deveria fazer o mesmo que o report acima sem agrupar por dias. Assim, cada saída será separada por dia, como nesse exemplo:

```
DIA 1
00:00-00:59 50
01:00-01:59 100
...
23:00-23:59 90
DIA 2
00:00-00:59 50
01:00-01:59 100
...
23:00-23:59 90
...
```

### METODOLOGIA

Para a realização desse trabalho, foi feito um estudo nos arquivos do The One para entender como funciona sua geração de Reports. Para tanto, contamos com o auxílio da geração de documentação Java na pasta Doc usando o arquivo *create\_docs.sh*. Com isso, conseguimos encontrar junto ao diretório **src/report**, a classe *ContactsPerHourReport.java* que gerava um report com duas colunas onde a primeira era um número inteiro referente a quantidade de horas que haviam se passado desde o início da simulação e o segundo a

quantidade de contatos realizados dentro daquela hora. como explicitado na imagem abaixo, retirada da Documentação do The One



Identificado o arquivo, seria necessário entender também onde e como os *Reports* eram criados pelo programa quando executado, para tanto tivemos auxílio do monitor Diogo Soares para a compreensão da configuração do The One, assim, foi visto que no arquivo *default\_settings.txt* no diretório principal era onde se requisitava o geração desses *Reports*, que eram criados no diretório **reports**.

Assim sabendo como realizar a geração dos *Reports* foi feito cópias da classe *ContactsPerHourReport.java* que foram modificadas em *Report1.java* e *Report2.java*, todas no diretório **src/report**, para realizar as tarefas exigidas nesse trabalho.

As modificações foram realizadas no método *done*, que é executado uma vez que a simulação é encerrada, como nos foi explicado pelo monitor. No arquivo *ContactsPerHourReport.java* o método realiza a impressão dos dados que estarão no Report conforme exposto na imagem abaixo:

```

public void done() {
    Iterator<Integer> iterator = contactCounts.iterator();
    int hour = 0;
    while (iterator.hasNext()) {
        Integer count = (Integer)iterator.next();
        write(hour + "\t" + count);
        hour++;
    }
    super.done();
}

```

Então, para o *Report1.java*, era necessário modificar a saída para que a primeira coluna correspondesse às horas, para tanto, foi utilizado a lógica de considerar a quantidade de horas na simulação e seu resto com a divisão por 24 (quantidade de horas num dia), para gerar a saída da maneira desejada. Ainda foi feito um tratamento caso o valor da hora fosse inferior a 10, para obedecer o formato da hora desejado. Como demonstra a imagem abaixo:

```

public void done() {

    Iterator<Integer> iterator = contactCounts.iterator();
    int hour = 0;

    while (iterator.hasNext()) {
        Integer count = (Integer)iterator.next();

        if (hour%24 < 10) {
            write("0" + hour%24 + ":00 - " + "0" + hour%24 + ":59" + "\t" + count);
        }else{
            write(hour%24 + ":00 - " + hour%24 + ":59" + "\t" + count);
        }
        hour++;
    }
    super.done();
}

```

Para o *Report2.java* foi feito a adição da divisão do dia, também considerando o resto da divisão por 24, uma vez que esse valor é zero, é um novo dia. Como demonstra o código a seguir:

```

public void done() {

    Iterator<Integer> iterator = contactCounts.iterator();
    int hour = 0;
    int day = 1;
    while (iterator.hasNext()) {
        Integer count = (Integer)iterator.next();
        if (hour%24 == 0) {
            write("DIA "+day);
            day++;
        }
        if (hour%24 < 10) {
            write("0" + hour%24 + ":00 - " + "0" + hour%24 + ":59" + "\t" + count);
        }else{
            write(hour%24 + ":00 - " + hour%24 + ":59" + "\t" + count);
        }
        hour++;
    }
    super.done();
}

```

Por fim, as alterações foram feitas no arquivo *default\_settings.txt* no diretório principal para fazer a simulação de cinco dias e gerar os *Reports* no diretório **reports**.

```

#
# Default settings for the simulation
#

## Scenario settings
Scenario.name = default_scenario
Scenario.simulateConnections = true
Scenario.updateInterval = 0.1
# 43200s == 12h
Scenario.endTime = 435600

```

...

```
## Reports - all report names have to be valid report classes

# how many reports to load
Report.nrofReports = 2
# length of the warm up period (simulated seconds)
Report.warmup = 0
# default directory of reports (can be overridden per Report with output setting)
Report.reportDir = reports/
# Report classes to load
Report.report1 = Report1
Report.report2 = Report2
```

...

Por fim, foi executado os arquivos *compile.sh* e *one.sh* no diretório principal para compilar nossas mudanças e rodar a simulação, respectivamente. Após o fim da simulação foram gerados os arquivos *default\_scenario\_Report1.txt* e *default\_scenario\_Report2.txt*, no diretório **reports** e em seu conteúdo estava o pedido pela atividade. Assim, a atividade foi concluída.

## CONCLUSÃO

Após o término do trabalho, verificamos que a geração de novos reports é ligeiramente simples, graças a documentação Java e aos próprias classes de Repor já criadas no The One.