

# Problema 4: Análise de ficheiros *log*



Algo correu mal no *datacenter* do DCC e precisamos da vossa ajuda para perceber o que se passou. Felizmente temos os registos dos acontecimentos (*logs*). São ficheiros de texto com uma mensagem por linha; cada linha começa com uma letra indicando o tipo de mensagem:

- **I** para mensagens informativas;
- **W** para mensagens de aviso (*warnings*);
- **E** para mensagens de erro.

As mensagens de erros têm também um nível inteiro entre 1 e 100 que indica a gravidade da ocorrência (1 é o menos grave e 100 o mais grave). Todas as mensagens têm um inteiro que representa o tempo da ocorrência (*timestamp*); o resto da linha é o conteúdo da mensagem.

Eis um extrato de duas linhas com uma mensagem informativa no instance  $t = 147$  seguida de um erro de nível 2 em  $t = 148$ :

```
I 147 mice in the air, I'm afraid, but you might catch a bat, and
E 2 148 #56k istereadeat lo d200ff] BOOTMEM
```

O ficheiro [sample.log](#) contém um pequeno exemplo destas mensagens; o ficheiro [error.log](#) contém todas as mensagens recuperadas do *datacenter*. Como este último ficheiro é longo, vamos escrever um programa para auxiliar a filtrar a informação. Começamos pelas declarações de tipos para estruturar a informação:

```
-- tipos de mensagens
data MessageType = Info
                  | Warning
                  | Error Int    -- argumento: nível do erro
                  deriving (Show, Eq)

type TimeStamp = Int    -- instante de tempo
```

[Início](#) ↩

Thu May 16 11:48:45 2024 Rodrigo Mendes (up202308366)

[Logout](#)

Note que `LogEntry` tem dois construtores: `LogMessage` representa mensagens corretamente formatadas e `Unknown` representa outras linhas de texto que não sigam o formato indicado acima.

## Preparação

Deve descarregar o ficheiro [Log.hs](#) com as declarações acima e colocá-lo no mesmo diretório em que vai desenvolver o seu código (num outro módulo). Para usar as definições do módulo `Log.hs` no seu oprograma deve colocar a seguinte declaração

```
import Log
```

no início do seu módulo. Não coloque as suas definições no módulo `Log.hs` porque não vai submeter esse ficheiro!

## Exercícios

- [Exercício 1: Análise de uma mensagem](#)  
(0 submissões)
- [Exercício 2: Inserir uma entrada numa árvore de pesquisa](#)  
(0 submissões)
- [Exercício 3: Colocar mensagens por ordem](#)  
(0 submissões)

## Conclusão

Será que consegue juntar todas as componentes anteriores para descobrir o que aconteceu no incidente do *datacenter*? Escreva um programa que:

1. leia o ficheiro de texto `error.log` usando `readFile`;
2. parta o texto em linhas usando `lines`;
3. converta a lista de linhas numa lista de `LogEntry` usando `parseLog`;
4. coloque as mensagens por ordem usando `sortMessages`;
5. filtre e imprima as mensagens de erro com gravidade 50 ou superior.

(Este último programa não é para submeter.)

---

Adaptado de um exercício do curso CIS 194, University of Pensilvania.

Pedro Vasconcelos, 2024