

## Teste para Desenvolvedor

Você deve criar um sistema de análise de dados, onde o sistema deve importar lotes de arquivos, ler e analisar os dados e produzir um relatório.

Existem 3 tipos de dados dentro desses arquivos.  
Para cada tipo de dados há um layout diferente.

### Dados do vendedor

Os dados do vendedor têm o formato id **001** e a linha terá o seguinte formato.

001çCPFçNameçSalary

### Dados do cliente

Os dados do cliente têm o formato id **002** e a linha terá o seguinte formato.

002çCNPJçNameçBusiness Area

### Dados de vendas

Os dados de vendas têm o formato id **003**. Dentro da linha de vendas, existe a lista de itens, que é envolto por colchetes []. A linha terá o seguinte formato.

003çSale IDç[Item ID-Item Quantity-Item Price]çSalesman name

### Dados de Exemplo

O seguinte é um exemplo dos dados que o sistema deve ser capaz de ler.

001ç1234567891234çPedroç50000  
001ç3245678865434çPauloç40000.99  
002ç2345675434544345çJose da SilvaçRural  
002ç2345675433444345çEduardo PereiraçRural  
003ç10ç[1-10-100,2-30-2.50,3-40-3.10]çPedro  
003ç08ç[1-34-10,2-33-1.50,3-40-0.10]çPaulo

### Análise de dados

Seu sistema deve ler dados do diretório padrão, localizado em% HOMEPATH% / data / in.

O sistema deve ler somente arquivos .dat.

Depois de processar todos os arquivos dentro do diretório padrão de entrada, o sistema deve criar um arquivo dentro do diretório de saída padrão, localizado em %HOMEPATH% /data/out.

O nome do arquivo deve seguir o padrão, {flat\_file\_name} .done.dat.

O conteúdo do arquivo de saída deve resumir os seguintes dados:

- **Quantidade de clientes no arquivo de entrada**
- **Quantidade de vendedor no arquivo de entrada**
- **ID da venda mais cara**
- **O pior vendedor**

O sistema deve estar funcionando o tempo todo.

Todos os arquivos novos estar disponível, tudo deve ser executado

Seu código deve ser escrito em Java.

Você tem total liberdade para utilizar google com o que você precisa.

Sinta-se à vontade para escolher qualquer biblioteca externa se for necessário.

### **Critérios de Avaliação**

- Clean Code
- Simplicity
- Logic
- SOC (Separation of Concerns)
- Flexibility/Extensibility
- Scalability/Performance

### **OBS de Envio:**

Enviar o projeto com a solução em anexo por email ou link para o github.