Prova Java

# Controle

|  |  |  |  |
| --- | --- | --- | --- |
| Data | Versão | Descrição | Responsável |
| 26/06/2017 | 1.0 | Criação do documento | Saulo Oliveira |
|  |  |  |  |

# Objetivo

Realizar um exercício simples, de maneira a comprovar conhecimentos básicos nas tecnologias utilizadas na Empresa.

A prova deverá ser realizada e entregue segundo as informações abaixo:

1. A prova deverá ser realizada em até 72 horas.
2. Todo o material produzido deverá ser publicado no GitHub para ser consultado pela equipe avaliadora. Fornecer o link junto com a entrega.
3. O projeto será fundamentalmente avaliado quanto às atividades propostas, porém outros aspectos como qualidade do código, aparência, frameworks utilizados e opções extras também serão levados em consideração.
4. Utilizar as tecnologias: Spring MVC, RetFUL, JSON e Spring DB + MySQL
5. Fornecer todo o conteúdo gerado (projetos, fontes, arquivos SQL).
6. O projeto será baixado do GIT fornecido compilado e executado

# O Projeto

O projeto simula as funcionalidades que utilizamos no dia a dia. Não requer uma coerência de negócio específica, mas exemplifica o conhecimento que precisamos. Fique a vontade para propor todo o incremento que desejar, desde que, focando nas premissas estabelecidas anteriormente.

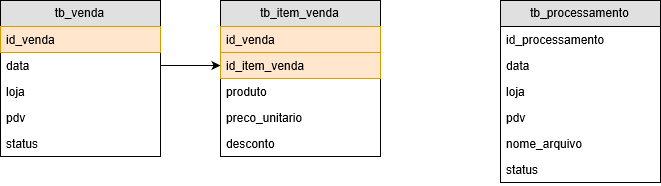
Sendo assim, sua missão será de criar um projeto JAVA para uma ferramenta de integração de dados com o modelo do anexo 1 – Modelo de Dados. Essa integração, ocorrerá através de 2 processos:

1. Consultas em uma tabela de vendas;
   1. Consultas realizadas em **2 threads** (tabela tb\_venda, tb\_item\_venda) onde o resultado dessas consultas devem ser transportado para outra tabela (tb\_processamento) 1 venda por thread, por minuto.
   2. As vendas já capturadas deverão constar como status “OK”
   3. Ao inserir os dados na tabela tb\_processamento, a coluna status deverá constar como “PENDENTE”;
   4. Uma **3a thread** deverá a cada minuto, capturar os registros da tabela tb\_processamento, gerar um arquivo TXT em uma pasta PROCESSADOS e marcar o status desses registros como “OK” na tabela tb\_processamento;
   5. Cada arquivo pode conter apenas 10 vendas, ou seja, serão gerados tantos arquivos quanto vendas até todas as vendas serem processados. O nome do arquivo é livre desde que, não sobreescreva. Gravar na tabela tb\_processamento o nome do arquivoonde a venda foi gerada.
   6. O layout do arquivo deverá ser conforme o ANEXO II – Layout do Arquivo
2. EndPoint com dados de venda
   1. Opcional ao processamento via banco, o endpoitpermitem que outros serviços enviem vendas diretamente para o sistema.
   2. A API RestFUL recebe um JSON contendo a venda
   3. Ao receber os dados (formato JSON) a venda deverá ser gravada na tabela tb\_processamento, a coluna status deverá constar como “PENDENTE”;
   4. O processamento deverá ocorrer de mesma maneira dos passos 1C e 1D
   5. O Layout do JSON deverá ser como o ANEXO III – Layout de venda do JSON
   6. A simulação de uma venda externa será feita, executando um POST no endpoint com o JSON com conteúdo (informar qual endpoint foi criado)
3. Os dados da tabela tb\_vendae tb\_item\_vendadevem ser simulados com o conteúdo do ANEXO IV –dados de venda
4. Para finalizar, uma página com o relatório do processamento deverá ser criada. O objetivo é consultar as vendas processadas, seu conteúdo. Filtros e buscas serão considerados diferenciais (informar a URL da página)

# FLUXO DE PROCESSAMENTO



# ANEXO 1 – Modelo de Dados



# ANEXO II – Layout do Arquivo

Todos os números, datas e valores com zeros a esquerda sem virgulas, pontos e outros caracteres.

ID VENDA(numero) : POS 0 – 10

DATA (data ddmmaaaa): POS 11 – 18

LOJA (numero): POS 19-22

PDV (numero): POS 23-25

PRODUTO (TEXTO): POS 26-36

PRECO UNITARIO (numero 2 casas decimais): POS 37-41

DESCONTO (numero 2 casas decimais): POS 42-46

VALOR TOTAL (numero 2 casas decimais, PRECO UNITARIO – DESCONTO): POS 47-51

*\* processar 1 venda por linha no máximo 10 vendas por arquivo*

# ANEXO III – Layout de venda do JSON

{

"id\_venda": 1,

"data": "2017-06-26T10:17:48-03:00",

"loja": 545,

"pdv": 4,

"itens": [

{

"produto": "0112348576",

"preco\_unitario": 1.99,

"desconto": 1.00

},

{

"produto": "0980987654",

"preco\_unitario": 9.99,

"desconto": 1.99

},

{

"produto": "9182909878",

"preco\_unitario": 1.00,

"desconto": 0.00

},

{

"produto": "098712312",

"preco\_unitario": 6.99,

"desconto": 2.00

}

]

}

# ANEXO IV – CVS de dados de venda

tb\_venda

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **id\_venda** | **data** | **loja** | **pdv** | **status** |
| 1 | 26/06/2017 18:41 | 1 | 1 | NÃO PROCESSADO |
| 2 | 27/06/2017 18:41 | 2 | 1 | NÃO PROCESSADO |
| 3 | 28/06/2017 18:41 | 3 | 1 | NÃO PROCESSADO |
| 4 | 29/06/2017 18:41 | 1 | 2 | NÃO PROCESSADO |
| 5 | 30/06/2017 18:41 | 2 | 2 | NÃO PROCESSADO |
| 6 | 01/07/2017 18:41 | 3 | 2 | NÃO PROCESSADO |
| 7 | 02/07/2017 18:41 | 1 | 3 | NÃO PROCESSADO |
| 8 | 03/07/2017 18:41 | 2 | 3 | NÃO PROCESSADO |
| 9 | 04/07/2017 18:41 | 3 | 3 | NÃO PROCESSADO |
| 10 | 05/07/2017 18:41 | 1 | 4 | NÃO PROCESSADO |
| 11 | 06/07/2017 18:41 | 2 | 4 | NÃO PROCESSADO |
| 12 | 07/07/2017 18:41 | 3 | 4 | NÃO PROCESSADO |
| 13 | 08/07/2017 18:41 | 1 | 5 | NÃO PROCESSADO |
| 14 | 09/07/2017 18:41 | 2 | 5 | NÃO PROCESSADO |
| 15 | 10/07/2017 18:41 | 3 | 5 | NÃO PROCESSADO |
| 16 | 11/07/2017 18:41 | 1 | 6 | NÃO PROCESSADO |
| 17 | 12/07/2017 18:41 | 2 | 6 | NÃO PROCESSADO |
| 18 | 13/07/2017 18:41 | 3 | 6 | NÃO PROCESSADO |
| 19 | 14/07/2017 18:41 | 1 | 7 | NÃO PROCESSADO |
| 20 | 15/07/2017 18:41 | 2 | 7 | NÃO PROCESSADO |
| 21 | 16/07/2017 18:41 | 3 | 7 | NÃO PROCESSADO |
| 22 | 17/07/2017 18:41 | 1 | 8 | NÃO PROCESSADO |
| 22 | 17/07/2017 18:41 | 2 | 8 | NÃO PROCESSADO |
| 22 | 17/07/2017 18:41 | 3 | 8 | NÃO PROCESSADO |

Tb\_item\_venda

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **id\_venda** | **id\_item\_venda** | **produto** | **preco\_unitario** | **desconto** |
| 1 | 1 | 0987654329 | 1.99 | 0.00 |
| 2 | 1 | 0987654329 | 1.99 | 0.00 |
| 3 | 1 | 0987654329 | 1.99 | 0.00 |
| 4 | 1 | 0987654329 | 1.99 | 0.00 |
| 5 | 1 | 0987654329 | 1.99 | 0.00 |
| 6 | 1 | 0987654329 | 1.99 | 0.00 |
| 7 | 1 | 0987654329 | 1.99 | 0.00 |
| 8 | 1 | 0987654329 | 1.99 | 0.00 |
| 9 | 1 | 0987654329 | 1.99 | 0.00 |
| 10 | 1 | 0987654329 | 1.99 | 0.00 |
| 11 | 1 | 0987654329 | 1.99 | 0.00 |
| 12 | 1 | 0987654329 | 1.99 | 0.00 |
| 13 | 1 | 0987654329 | 1.99 | 0.00 |
| 14 | 1 | 0987654329 | 1.99 | 0.00 |
| 15 | 1 | 0987654329 | 1.99 | 0.00 |
| 16 | 1 | 0987654329 | 1.99 | 0.00 |
| 17 | 1 | 0987654329 | 1.99 | 0.00 |
| 18 | 1 | 1234567890 | 2.99 | 1.00 |
| 19 | 1 | 1234567890 | 2.99 | 1.00 |
| 20 | 1 | 1234567890 | 2.99 | 1.00 |
| 21 | 1 | 1234567890 | 2.99 | 1.00 |
| 22 | 1 | 1234567890 | 2.99 | 1.00 |
| 22 | 1 | 1234567890 | 2.99 | 1.00 |
| 22 | 1 | 1234567890 | 2.99 | 1.00 |
| 1 | 2 | 2123456789 | 3.99 | 0.99 |
| 2 | 2 | 2123456789 | 3.99 | 0.99 |
| 3 | 2 | 2123456789 | 3.99 | 0.99 |
| 4 | 2 | 2123456789 | 3.99 | 0.99 |
| 5 | 2 | 2123456789 | 3.99 | 0.99 |
| 6 | 2 | 2123456789 | 3.99 | 0.99 |
| 7 | 2 | 2123456789 | 3.99 | 0.99 |
| 8 | 2 | 2123456789 | 3.99 | 0.99 |
| 9 | 2 | 2123456789 | 3.99 | 0.99 |
| 10 | 2 | 2123456789 | 3.99 | 0.99 |
| 11 | 2 | 2123456789 | 3.99 | 0.99 |
| 12 | 2 | 2123456789 | 3.99 | 0.99 |
| 13 | 2 | 2123456789 | 3.99 | 0.99 |
| 14 | 2 | 2123456789 | 3.99 | 0.99 |
| 15 | 2 | 2123456789 | 3.99 | 0.99 |
| 16 | 2 | 2123456789 | 3.99 | 0.99 |
| 17 | 2 | 2123456789 | 3.99 | 0.99 |
| 18 | 2 | 2123456789 | 3.99 | 0.99 |
| 19 | 2 | 2123456789 | 3.99 | 0.99 |
| 20 | 2 | 2123456789 | 3.99 | 0.99 |
| 21 | 2 | 2123456789 | 3.99 | 0.99 |
| 22 | 2 | 2123456789 | 3.99 | 0.99 |
| 22 | 2 | 2123456789 | 3.99 | 0.99 |
| 22 | 2 | 2123456789 | 3.99 | 0.99 |