

## Capítulo 3 – Mensageria

### Exercícios

1. [Arquitetura] Uma aplicação local produz listas de preços em formato CSV que são depositados em uma pasta. Outra aplicação produz listas em formato XML que são enviadas para uma URL. Uma terceira aplicação produz listas em formato HTML (tabelas) também disponíveis em uma URL. Você precisa elaborar uma solução de integração que construa listas de produtos com preços obtidos das três listas em formato HTML. A solução não precisa fazer nada além de listar produtos e preços. Esboce uma solução usando padrões de integração.
2. [Message, Channel] A classe **MockData** contém um List com 10 documentos XML em formato String. São dados de teste. O método estático **getMockData()** retorna esta lista. Escreva um Produtor JMS que envie esses documentos XML para a fila **"inbound"** do ActiveMQ. Você vai precisar construir uma mensagem. A mensagem construída deve conter um cabeçalho **"Tipo"** com o valor **"xml"**. As configurações do ActiveMQ já estão prontas no arquivo **jndi.properties** (as filas do ActiveMQ têm o mesmo nome JNDI). Configure seu ambiente e verifique que o ActiveMQ recebe as mensagens na fila. Envie seis mensagens com preços diferentes.
3. [Message, Channel] Escreva uma classe Java que consuma as mensagens da fila **"inbound"** e imprima seu conteúdo na saída padrão se elas tiverem um cabeçalho **"Tipo"** com o valor **"xml"**.
4. [Endpoints] A classe **FileInboundAdapter** lê arquivos da pasta **INBOX** e envia para a fila **"inbound"**. A classe **FileOutboundAdapter** consome mensagens de **"inbound"** e grava em arquivo na pasta **OUTBOX**. Configure os valores de **INBOX** e **OUTBOX** na sua máquina. Copie (não mova) os arquivos da pasta **arquivos** para a pasta **INBOX** e execute o **FileInboundAdapter**. Veja que os arquivos sumiram da pasta. Verifique na console do ActiveMQ se as mensagens estão na fila **"inbound"**. Alguns desses são XML, portanto se você rodar o exercício 3 deve conseguir imprimir seu conteúdo.
5. [Pipes and Filters] Copie novamente arquivos para a pasta **INBOX** e rode o **FileInboundAdapter**. Agora deve haver vários arquivos na fila **"inbound"**. Rode o **FileOutboundAdapter**. Ele está esperando mensagens na pasta **"outbound"**, mas eles não estão lá. Elabore uma solução para integrar os dois. Pode ser em Camel, Spring Integration ou em JMS (use a classe **JMSChannelBridge**).
6. [Router] Escreva um roteador que analise o cabeçalho **"Tipo"** de cada mensagem de uma fila e redirecione para um destino diferente. Os tipos possíveis são **"png"**, **"xml"** e **"txt"**. Os destinos podem ser **"png-queue"**, **"xml-queue"** e **"txt-queue"**. Use JMS, Camel ou Spring Integration. Altere a classe que você criou no exercício 3 para que ela consuma das filas **"xml-queue"** e **"txt-queue"** e o **FileOutboundAdapter** para que consuma da fila **"png-queue"**.
7. [Translator] A classe **JMSChannelBridge** simplesmente consome mensagens de um canal e copia em outro. Use-a como template para criar um **UpperCaseTransformer** que transforme mensagens de entrada em caixa-alta na saída. Conecte o componente para receber mensagens do **"txt-queue"**. Se quiser implemente em Camel ou Spring Integration.