



UNIVERSIDADE DO MINHO

DEPARTAMENTO DE INFORMÁTICA

Paradigmans de Sistemas Distribuídos
**NeFIT: Negociação de Fabricação e
Importação de Produtos**

Eduardo Barbosa (a83344) João Vilaça (a82339)

5 de Março de 2021

Conteúdo

1	Introdução	2
2	Análise do Problema	2
2.1	Arquitetura	3
3	Protocol Buffers	4
4	Clientes	4
5	Frontend	4
6	Negociadores	4
7	Conclusão	5

1 Introdução

Este projeto insere-se na Unidade Curricular Paradigmas de Sistemas Distribuídos. O objetivo é então aplicar todos os conhecimentos aprendidos ao longo do semestre, e perceber o lado mais prático dos tópicos lecionados. Assim sendo, neste trabalho iremos aplicar conceitos como serialização, programação por atores, programação orientada a mensagens, e arquiteturas REST.

Para isso, vai desenvolver-se a plataforma Negociação de Fabricação e Importação de Produtos, que entre os vários componentes faz uso deste vários conceitos, metodologias e tecnologias.

2 Análise do Problema

Pretende-se desenvolver um protótipo de uma plataforma de negociação entre fabricantes e importadores de produtos. Esta plataforma deverá ser composta por vários sistemas, nomeadamente por uma aplicação Java para os clientes, um servidor de frontend, desenvolvido em Erlang, e servidores que atuam como negociadores e catálogo, desenvolvidos em Java recorrendo ao uso da dependência Dropwizard.

Cada fabricante indicará a disponibilidade para produzir um determinado artigo, numa quantidade mínima e máxima, a um preço mínimo (unitário), bem como o período de tempo durante o qual os importadores poderão fazer ofertas de encomenda (período de negociação). Por sua vez, cada importador indica a quantidade e valor unitário a que está disposto a pagar por um determinado artigo de um fabricante.

Pretende-se que seja possível a existência de clientes em elevado número, desempenhando funções de fabricante ou importador, e que estes comuniquem exclusivamente com o servidor de frontend.

2.1 Arquitetura

Os clientes assumindo o papel de Fabricante ou Importador comunicam com o Frontend. No caso dos Fabricantes eles enviam pedidos de login e de produção, recebendo, através do uso de ØMQ Pub-Sub, notificações dos resultados das negociações quando terminam. No caso dos Importadores, também eles podem efetuar login através do Frontend, e podem submeter encomendas em negociações. São ainda também capazes de subscrever notificações sobre a criação de negociações de determinada produtor e de resultados de negociações em que participam.

O Frontend por sua vez, faz a comunicação e balanceamento de carga com os negociadores. Este é capaz através de API Rest verificar as credenciais de um utilizador, dado um username e uma password, e através de ØMQ Push/Pull, enviar para os servidores dos negociadores pedidos de criação de negociação e de encomenda, e receber deles notificações sobre a criação das mesmas e, mais tarde, do seu término.

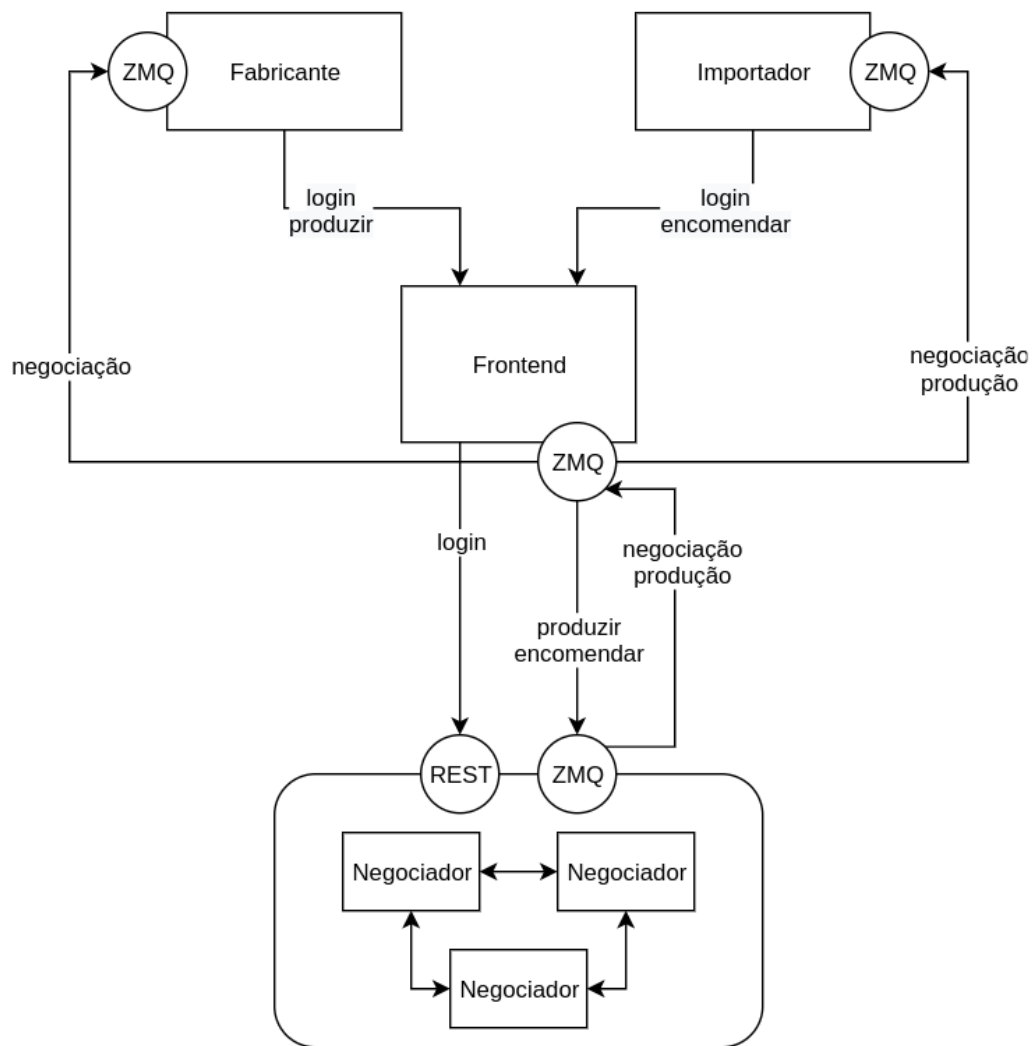


Figura 1: Arquitetura Geral

3 Protocol Buffers

Foram utilizados Protobufs para ajudar a comunicação de dados entre os sistemas homogêneos que compõem esta plataforma. Este é um mecanismo criado e usado pelo Google para serializar dados estruturados, independente de linguagem e de plataforma.

Primeiro definimos os dados estruturados que serão enviados pelos canais de comunicação, num ficheiro de extensão `.proto`. Em seguida, esta definição foi compilada para Java para os clientes e para os negociadores e para Erlang no caso do Frontend.

No ficheiro `.proto`, foram incluídas as definições de Fabricante, Importador, Negociação, Encomenda e Mensagem.

4 Clientes

Os clientes são o elemento mais simples do sistema, este componente é uma interface de texto usada pelos fabricantes e pelos importadores para interagirem com o sistema.

5 Frontend

O Frontend é a camada responsável pela distribuição e balanceamento de carga proveniente dos Clientes, bem como a distribuição dos seus pedidos. Sendo que todos os Clientes se ligam ao Frontend, esta camada necessita de ser programada com um elevado nível de paralelismo em mente, surgindo assim Erlang, e programação por atores, como uma solução natural. Podemos decompor o Frontend em alguns "micro serviços":

- **frontend**: ponto de entrada da aplicação
- **login_manager**: Serviço responsável pela autenticação
- **user_manager**: Serviço responsável por ligar o Cliente ao resto da aplicação
- **nego_manager**: Serviço responsável por fazer toda a comunicação com o Negociadores

O ponto de entrada no Frontend é pelo serviço homónimo, sendo que este quando um cliente se liga é criado um novo ator para o mesmo. A partir daqui o cliente pode-se autenticar, sendo que este pedido fica encarregue ao serviço **login_manager**. Este serviço utiliza a API rest disponibilizada pelos Negociadores para verificar a autenticidade da identidade do cliente. Quando o cliente envia uma mensagem, esta passa pelo **nego_manager**, sendo então encaminhada para os Negociadores. Aquando de uma mensagem de resposta, ou de outro evento, esta é processada pelo **user_manager**, sendo no fim enviada para o Cliente.

6 Negociadores

Os negociadores compõem um sistema distribuído, capaz de se auto-gerir. Estes mantêm uma visão causalmente coerente de todas as operações realizadas no sistema e foram baseados num modelo de distribuição desenvolvido na cadeira de Fundamentos de Sistemas Distribuídos.

Eles são responsáveis por pelo processamento dos anúncios de oferta para a produção de artigos e da colocação das ordens de encomenda que recebe do Frontend.

Recebido e validado um pedido (verificação de autorização e bem estruturação do mesmo) ele é enviado para o middleware de distribuição que através de 2-Phase-Commit submete o pedido. Depois disso o pedido é registado no serviço de negociações e uma notificação é enviada através de ØMQ Pub-Sub.

Os negociadores implementam um Scheduler, que periodicamente verificam quais as negociações cujo tempo de vida terminou, e em seguida, caso a quantidade mínima não tenha sido

atingida a oferta de produção é então cancelada, caso contrário é satisfeito o conjunto de ordens que corresponda ao maior valor (preço) global de encomenda desse artigo. No final, uma notificação é enviado com este resultado.

Os negociadores assumem ainda o papel de Catálogo, disponibilizando uma API Rest, servindo informações de fabricantes, importadores, negociações em curso e histórico.

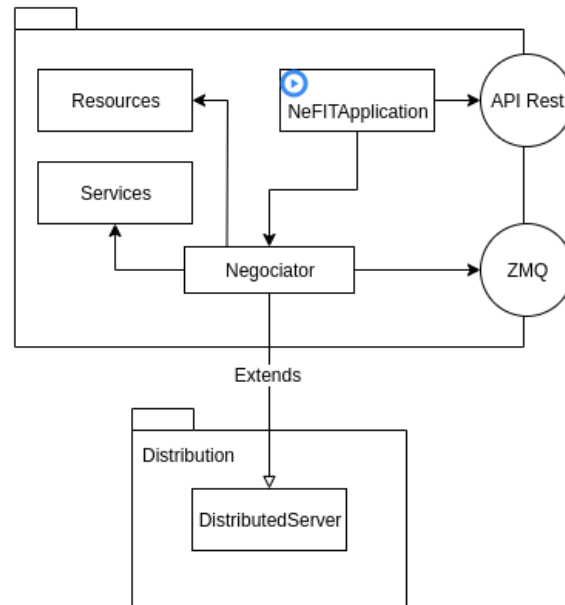


Figura 2: Arquitetura Geral

7 Conclusão

Após terminar o trabalho, percebemos toda a aplicação prática dos conceitos abordados durante todo o semestre, e também ficamos a perceber melhor as vantagens e desvantagens que cada conceito. Apesar de consideramos que o projeto foi bem conseguido, nem todo o processo foi fácil, principalmente devido há falta de tempo. A construção de toda a arquitetura do sistema foi bastante simples, o problema foi mesmo a implementação, ou seja, toda a lógica na Exchange, a definição dos endpoints, e todo o processamento necessário da frontend. A criação dos protocol buffers também variou bastante ao longo do trabalho, consoante as necessidades do grupo.