UFAL – Universidade Federal de Alagoas
IC – Instituto de Computação
Redes de Computadores I – 2014/2

João Paulo Clarindo dos Santos
Karla Elisabeth Cristo dos Santos

Um sistema de geração de produtos de uma fábrica de Caminhões e Ônibus

1. Panorama geral do sistema

Um chassi é uma estrutura de suporte para outros componentes que pode ser feita de aço, alumínio, ou qualquer outro material rígido. Sua aplicação mais conhecida é em veículos para sustentar os sistemas embarcados (carrocerias).

Uma fábrica de Caminhões e Ônibus tem como principal produto os caminhões e chassis de ônibus. Um produto é obtido através peso do chassi e a potência do seu motor em cavalos, já que as combinações destas duas variáveis determinam a aplicação na qual o produto foi designado (exemplo, um caminhão simples para circular na cidade geralmente possui baixa potência e peso).

A fábrica fornece as informações para as concessionárias sobre os seus produtos com as *features* iniciais. Em seguida, é dever das concessionárias armazenar os dados dos produtos com todas as *features* adicionais que é oferecida pelo produto.

As informações gerais dos produtos (tais como peso, potência do motor, altura da cabine de um caminhão ou posição do motor de um chassi de Ônibus) estão armazenadas em um banco de dados da fábrica (servidor), e as concessionárias (os clientes) utilizam-se dessas informações básicas para gerar um novo produto com *features* já pré-estabelecidas (como tipo da direção, freios, etc.), gerando assim um produto final, que é enviado para o email do usuário.

2. Aplicação dos Sockets

Numa situação real de uma indústria, sempre surge novos produtos, com características diferentes. Manter toda a relação de produtos em cada cliente não é vantajoso porque não existem garantias de que todos os clientes estarão atualizados com os novos produtos, e sem os produtos antigos que saíram de linha, por exemplo.

O uso de um servidor para armazenar esses dados é a maneira mais eficaz, para que múltiplos clientes obtenham os dados de uma maneira consistente e atualizada, com isto estamos utilizando a API de sockets em Java para fazer esta comunicação entre os clientes e o servidor de modo a atender as requisições geradas.

3. Problemas enfrentados

Inicialmente, todos os dados eram enviados somente pelo tipo nativo String utilizando os Streams de comunicação, porém as Strings enviadas do servidor para o cliente não possuíam uma organização adequada, e consequentemente gerava muitos erros. Para contornar isso, as listas enviadas do servidor para cliente foram codificadas em Json object e decodificadas pelo cliente.

Além disso, o usuário poderia digitar algo fora do padrão. Para isso, foi feita uma mensagem do servidor para o cliente (seção 4.2) que argumenta erro de sintaxe em algum comando feito pelo cliente.

4. Descrição do protocolo

O protocolo desenvolvido para este projeto é definido pelo seguinte padrão:

1 String: Comando para o servidor (LISTAR, SELECIONAR).
2 String: Escolher os dados a serem manipulados (PRODUTOS , FEATURE).
3 String: Código da mensagem.

4.1 Cliente para Servidor

- LISTAR *X* n: Lista um produto ou *feature* x. n possui um valor fixo determinado pelo programa Cliente, e não pode ser alterado pelo usuário.
- SELECIONAR X codigo: Seleciona um produto ou *feature* x a partir do código do produto ou *feature* que foi listado anteriormente. O usuário pode escolher livremente qual produto ou *feature*, contanto que o código (e o produto/*feature* relacionado) esteja disponível na base de dados.
- ENCERRAR: Envia uma mensagem para o servidor que irá desconectar.

4.2 Servidor para Cliente

- Dados de Produtos/*Features*: Dados no formato json ou String padrão que são recebidos imediatamente após a requisição feita pelo cliente. Os dados são retirados de um banco de dados armazenado no servidor.

- ERRO: Um erro de sintaxe do usuário (como um código de produto/*feature* não encontrado), que faz com que o programa cliente seja encerrado.