

## Gerenciamento distribuído de frota de veículos

A empresa XYZ, atua no ramo da compra e venda de carros. Ela possui diversas filiais espalhadas por todo o Brasil, contando com várias plataformas de processamento. A empresa deseja unificar seu estoque de carros bem como proporcionar o acesso de seus produtos a seus representantes nas filiais bem como divulgar seus carros entre o público em geral. Para resolver esse problema, a empresa XYZ sabe que você estuda Ciência da Computação e possui um grupo de amigos que podem ajudá-lo a desenvolver essa solução.

O trabalho consiste em desenvolver aplicações para manipulação da base de dados de automóveis usando Sistemas Distribuídos. Essas aplicações devem funcionar em várias plataformas e interfaces. O trabalho será desenvolvido pelo seu grupo, usando software de desenvolvimento colaborativo e com controle de versões tais como Git (preferencial), subversion, bitbucket, ....

Deverão ser disponibilizadas aplicações com interface de comunicação usando sockets TCP, sockets UDP, Web Services SOAP e Web Services Rest. Deverá ser implementado um Banco de Dados para o armazenamento de dados dos carros. Cada carro deve possuir os seguintes dados:

- código: Inteiro (chave primária)
- Marca: String
- Modelo: String
- Ano: Integer
- Potencia: Float
- Carga: Float
- Complemento: Varchar

O Banco de Dados pode ser implementado com qualquer tecnologia, uma vez que não é o foco do trabalho.

Deve-se implementar pelo menos os seguintes servidores/serviços (Java):

- Servidor que aceita conexão TCP, porta 2006;
- Servidor que aceita conexão UDP, porta 2010;
- Serviço Web (Web Service) SOAP;
- Serviço Web (Web Service) Rest;

Os servidores/serviços deverão suportar as seguintes operações:

- Adiciona: adiciona carro ao BD. Caso código do carro já exista, não adicionar e informar um erro.
- Altera: altera carros no BD. Deve-se informar o carro, o dado a ser alterado e seu novo valor. Caso o código do carro não exista, retornar essa informação;
- Excluir: apaga um carro no BD. Deve-se informar o código do carro. Caso o carro não exista no BD, retornar essa informação;
- Consulta: retorna os dados de um carro. Caso o carro não exista no BD, retornar essa informação;
- ListaAnoModelo: deve ser recebido um ano e um modelo e retornar uma lista contendo os carros cujo ano de fabricação e modelo conferem com os argumentos recebidos.

A cada conexão TCP recebida pelo servidor, deve ser criada uma *thread* independente para atender o processamento da conexão. O servidor deve receber do cliente as operações e os argumentos, processar e retornar o resultado ao cliente pela mesma conexão, atualizando um log (tela ou arquivo). Deve-se elaborar um formato para ser usado nas comunicações TCP. A conexão deve ser mantida até que o cliente resolva concluir a conexão.

Cada mensagem UDP recebida deve ser colocada em uma fila de mensagens a serem processadas. Essa operação de recebimento deve ser realizada em uma *thread*. Uma outra *thread* é responsável por retirar mensagens da fila, processar e devolver ao cliente os resultados. Um log de processamento das mensagens (tela ou arquivo) deve ser criado.

Os serviços Web vão disponibilizar o acesso as mesmas operações que os servidores. No caso de serviços Rest, devem ser implementadas operações em pelo menos quatro métodos HTTP: GET, PUT, POST e DELETE. Pode-se escolher a codificação dos dados entre XML ou JSON.

O desenvolvimento deverá contar com os seguintes clientes:

- Cliente com interface texto para servidor TCP (java);
- Cliente com interface texto para servidor UDP (java);
- Cliente com interface gráfica para interação com Serviço Web SOAP;
- Cliente Web para interação com o Serviço Web Rest.

O cliente texto para o servidor TCP deve estabelecer a conexão e solicitar ao usuário a ação (operação) a ser realizada e seus argumentos. Ele deve mostrar o resultado do servidor e ficar disponível para outras operações. Um opção para encerrar o cliente deve ser fornecida.

O cliente texto para o servidor UDP deve receber a ação (operação) a ser realizada e os argumentos necessários em linha de comando. Deve mostrar os usuário os dados recebidos e encerrar sua execução.

O cliente com interface gráfica para interação com o Web Service SOAP deve proporcionar acesso a todas as operações disponibilizadas pelo Serviço SOAP. O formato da interface gráfica é de livre escolha.

O cliente web para interação com o Web Service Rest deve proporcionar acesso a todas as operações disponibilizadas pelo serviço com uma interface Web. O formato da interface Web bem como a tecnologia usada para o desenvolvimento é de livre escolha.

Serão realizadas duas avaliações: uma referente ao desenvolvimento do grupo e outra referente ao desenvolvimento individual de cada componente do grupo. A relação de funcionalidades descritas até aqui equivalem 70% da nota do trabalho. Os demais 30% serão computados se o grupo tiver funcionalidades extra, como por exemplo:

- modelo de falhas (geral) e tratamento de falhas para o serviço e consumidor SOAP;
- uma interface de acesso para dispositivos móveis;
- replicação do BD com uma cópia nas nuvens;
- integração com uma rede social;
- exploração da tecnologia PUSH HTTP;
- outros (consultar professor)...

O líder da equipe deverá realizar uma apresentação técnica (mostrar os modelos usados, as aplicações desenvolvidas e as tecnologias usadas; não usar código-fonte) do que foi construído (10 minutos). Em seguida, deve-se proceder a demonstração das implementações. Uma máquina ficará com os servidores/serviços, mostrando os logs de processamento enquanto outra ficará com os clientes/consumidores usados. Serão reservados dois projetores para a apresentação (40 min).

Em seguida, cada integrante será entrevistado individualmente para informar o que implementou e como foi realizada a implementação.

## Grupos Trabalho SD

Grupo 1	Grupo 2
Cristiano	Andrei
Djesica	Eduardo
Júlio	Fabricio
Leandro	Felipe (Lider)
Leonardo	João
Mariana	Luciano
Mateus (Lider)	Tiago
Ricardo	Victor
Sanderson	