Universidade Federal de Ouro Preto

Instituto de Ciências Exatas e Aplicadas

Departamento de Computação e Sistemas

Sistemas Distribuídos

Relatório Final:

Sistema de Gerenciamento de uma Concessionária de Veículos

Guilherme Marx Ferreira Tavares - 14.1.8006

Ícaro Bicalho Quintão - 14.1.8083

Leonardo Sartori de Andrade - 15.1.8061

Professora - Carla Rodrigues Figueiredo Lara

João Monlevade

Dezembro de 2018

Sumário

Tema

Replicação de dados e sincronização de Banco de Dados localizados em computadores diferentes.

Introdução

O Sistema de Gerenciamento de uma Concessionária de Veículos utiliza replicação de dados e sincronização de Banco de Dados localizados em computadores diferentes. Em tempos antigos, os dados eram armazenados em fichas de papel e pastas. A partir da década de 70 surgiram os primeiros Bancos de Dados relacionais da forma que são utilizados atualmente, os quais foram criados na intenção de facilitar a organização física e lógica dos dados com base em entidades e relacionamentos. Nos dias atuais, a organização dos dados em Banco de Dados se tornou pré-requisito para qualquer empresa de sucesso. A utilização de consultas nesses bancos facilitou infinitamente a compreensão de dados armazenados e seus usos na administração de empresas.

Justificativa

A utilização de Banco de Dados em sistemas maiores, entretanto, nos trás diversos problemas. O primeiro e mais marcante deles é manter a consistência dos dados. Em sistemas grandes, temos muitos dispositivos fazendo consultas e escritas nesse Banco de Dados e são necessários mecanismos para manter os dados existentes nele corretos e atualizados. Outro problema marcante em Bancos de Dados utilizados em sistemas maiores é o gargalo: Muitos dispositivos tentando acessar o mesmo banco geram lentidões e travamentos, o que significa prejuízo quando se trata no contexto de empresas. Para contornar esses problemas, podemos nos utilizar de Banco de Dados distribuídos em arquitetura Peer-to-peer. Dessa forma, cada computador tem seu próprio banco de dados e o sistema se responsabiliza de sincronizá-los de forma transparente ao usuário do sistema. Durante todo o projeto nos deparamos com conceitos vistos em sala de aula como a replicação de dados, RMI e o controle de concorrência.

Arquitetura Cliente Servidor

Clientes “invocando” serviços em um servidor

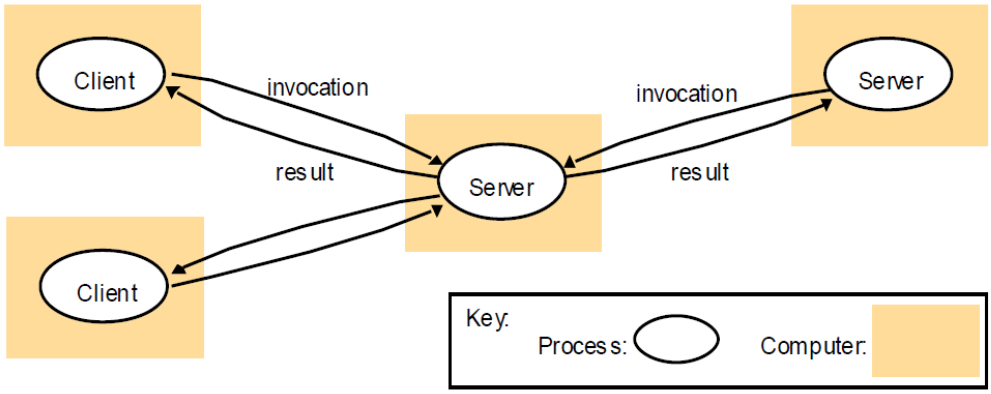

Figura 1: Slide 41 da aula 3 de Sistemas Distribuídos. Arquitetura Cliente-Servidor utilizada em nosso trabalho.

Implementação de RMI

- Vários objetos e módulos separados estão envolvidos na realização de uma invocação a método remoto.
- A invoca um método em um objeto remoto B, para o qual mantém uma referência de objeto remoto.

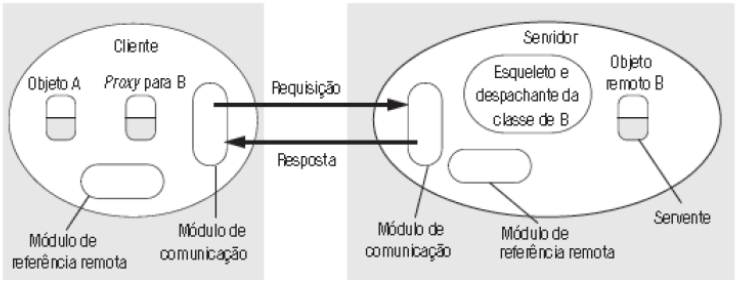

Figura 5.15 A função do *proxy* e do esqueleto na invocação a método remoto.

Figura2: Slide 20 da aula 9 de Sistemas Distribuídos. O modo de comunicação RMI foi utilizado no trabalho.

O cronograma

O cronograma apresentado no dia 22 de outubro de 2018, na apresentação de andamento do projeto, foi mantido até a conclusão do mesmo. Ele ficou da seguinte forma:

O que foi feito	Data
O Repositório do projeto no GitHub. O Diagrama Entidade-Relacionamento do Banco de Dados. Esquema Relacional (projeto lógico) do Banco de Dados com definição das tabelas. O Script de criação do Banco. A Interface RMI da rede, que será utilizada tanto na comunicação entre ClienteBanco de IDs quanto entre Cliente-Cliente. Os Objetos Model.	Apresentado no dia 22/10/2018
Prototipação das telas do projeto e definição das consultas possíveis ao Banco.	Versão final pronta no dia 28/10/2018
Complementação dos Objetos Model e DAO para suportar todas essas possíveis consultas.	Finalizado no dia 12/11/2018
Desenvolvimento dos pacotes View e Controller.	Finalizado no dia 08/12/2018

Desenvolvimento e alterações

O instalador do sistema deverá ser único. Os objetos compartilháveis representarão objetos DAO que foram inseridos, removidos ou atualizados no estado mais atual dos bancos de dados. Será utilizada Comunicação indireta e síncrona, uma vez que, a fim de se manter transparente ao usuário do sistema, os computadores da rede receberão as atualizações que forem feitas em todos os Bancos em uma porta específica para comunicação desse sistema. Como arquitetura do Software do sistema, será utilizado o padrão ModelView-Controller com um pacote DAO explicitamente separado a fim de ser utilizado como objetos compartilháveis.

A primeira e mais importante decisão de projeto tomada foi o abandono da arquitetura peer-to-peer e uso de arquitetura Cliente-Servidor. Como o fato de não se ter um

servidor central é pré-requisito do projeto, nosso trabalho fica concentrado em duas entidades:

• Banco de IDs: Um mini-servidor, serve para armazenar IDs das operações feitas no Banco de Dados e o nome do respectivo cliente que fez essa operação.

• Cliente: Apesar do nome, esse cliente é ao mesmo tempo Cliente (na relação com o Banco de IDs) e pode ser Cliente ou Servidor na relação com outros Clientes.

Essa comunicação funciona da seguinte forma:

1. Cliente que deseja fazer uma alteração no banco de dados local se comunica com o Banco de IDs para que este atualize sua tabela de IDs colocando o nome deste Cliente no respectivo local de sua alteração (O Banco de IDs não armazenará nenhum script de alteração no banco de dados, somente o ID da operação e o nome do cliente que a fez).

2. Todos os Clientes possuem uma thread que monitora essa tabela no Banco de IDs via RMI, assim, no momento em que essa thread detecta uma alteração na tabela, os Clientes pegam o nome de quem fez a alteração através de um método remoto.

3. Uma vez em posse do nome de quem fez a alteração, os Clientes desatualizados se comunicam via outra interface RMI com o Cliente que está atualizado e pegam o script da alteração que foi feita no banco.

4. Ao aplicar o mesmo script em seus respectivos Bancos de Dados Locais, os bancos se mantêm atualizados, a replicação foi feita com sucesso.

Outra decisão importante tomada foi com relação à arquitetura Model-View Controller, foi decidido que seria mais organizado manter toda interação de rede dentro do pacote Model de modo que este teria, então, as regras de negócio, toda a estrutura de controle do banco de dados e ainda toda a interface de comunicação em Rede. Dessa forma, fica claro que o pacote Model concentra a maior parte do trabalho, deixando o cronograma inicial (com 1 mês de desenvolvimento de cada pacote) obsoleto, uma vez que o Model é a parte mais trabalhosa desse projeto.

Durante o desenvolvimento do trabalho, utilizamos conceitos aprendidos não somente em Sistemas Distribuídos, como foi dito, utilizamos também, conceitos vistos em Banco de Dados e Engenharia de Software. Para auxiliar na criação do banco de dados e no

entendimento geral do projeto, foi criado a Entidade-Relacionamento, esquema relacional, assim como o protótipo de telas.

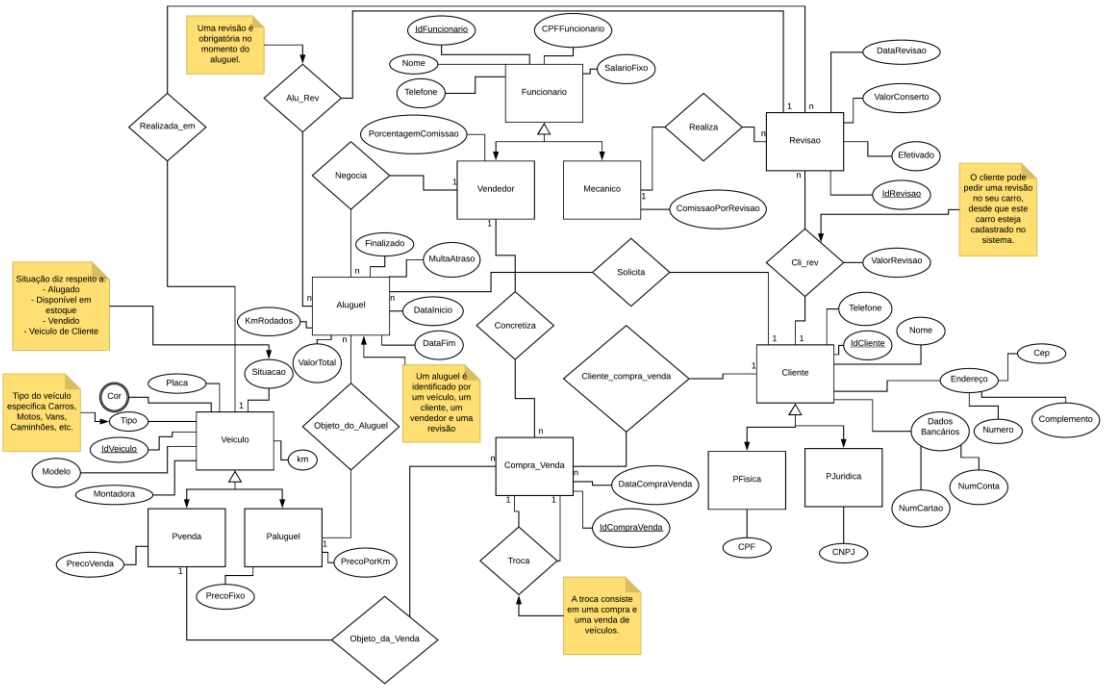

Figura 3: Modelo de Entidade-Relacionamento

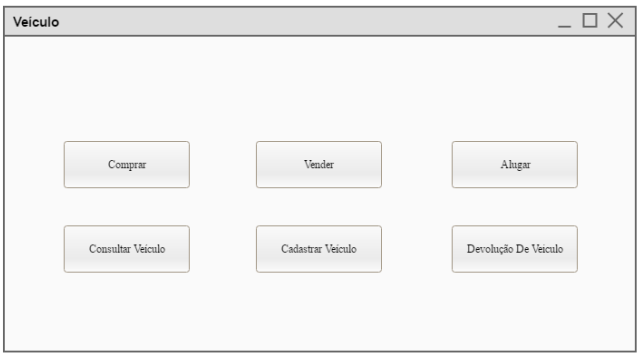

Figura 4: Protótipo de tela elaborado com o programa Pencil, para a interface de Veículo do programa.

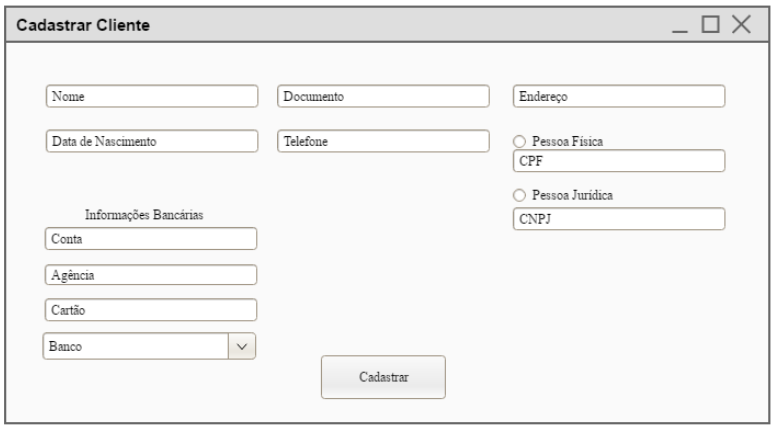

Figura 5: Protótipo de tela elaborado com o programa Pencil, para a interface de Cadastro de cliente do nosso programa.

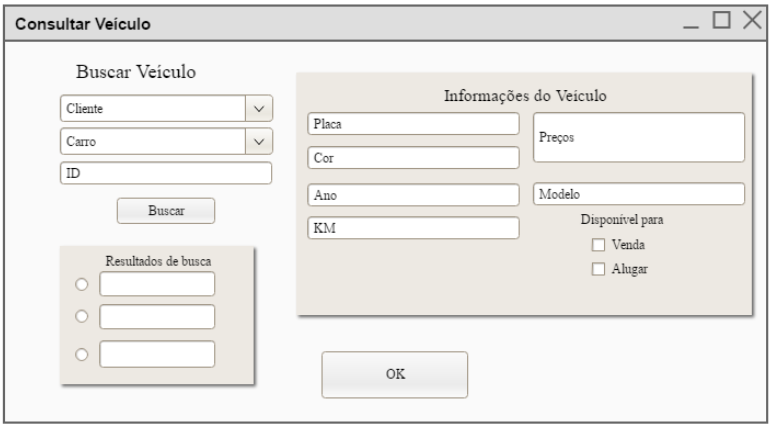

Figura 6: Protótipo de tela elaborado com o programa Pencil, para a interface de fazer consulta de um Veículo do programa.

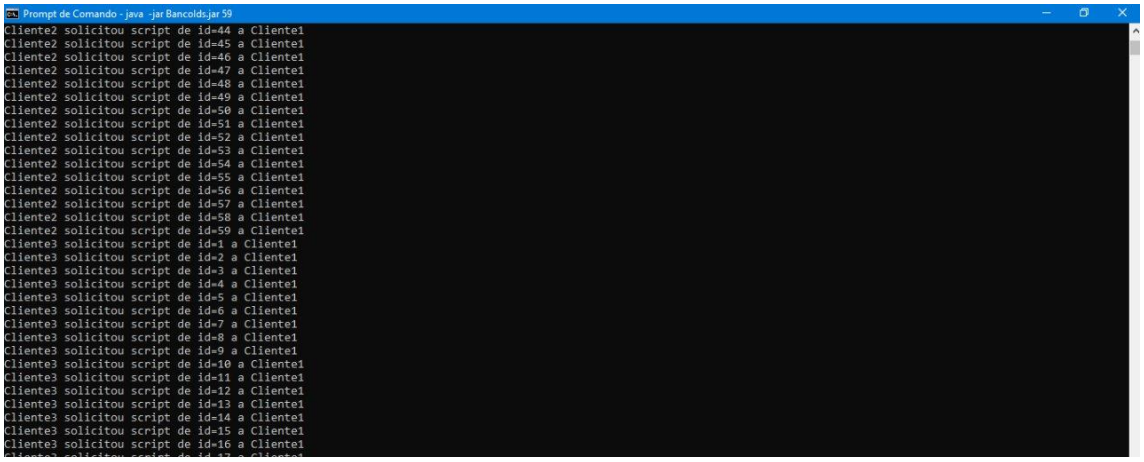

Figura 7: Atualizações de Cliente2 e Cliente3.

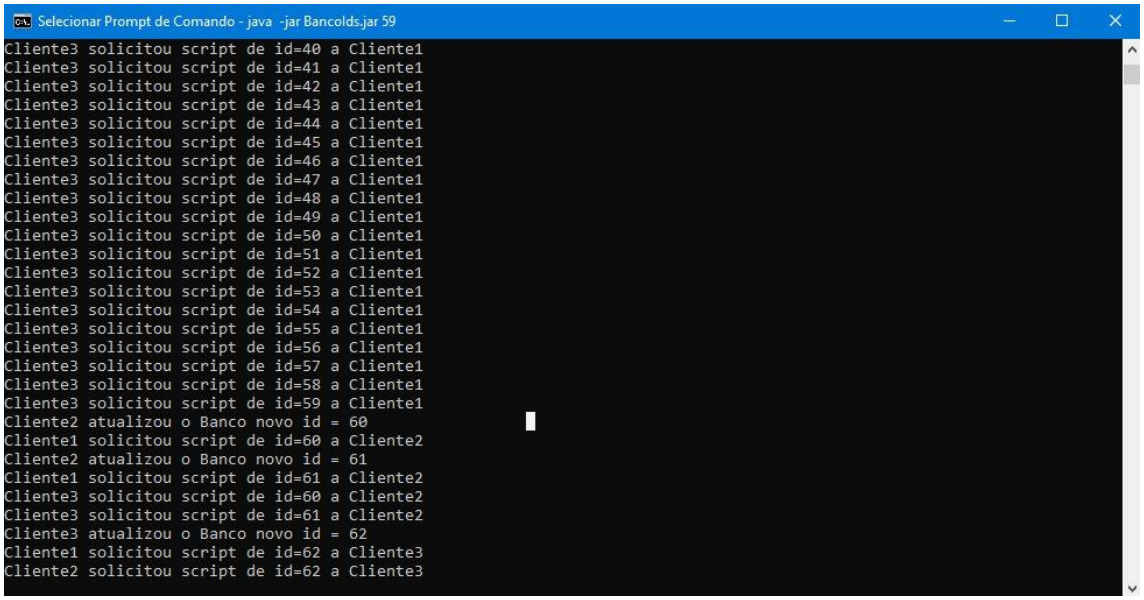

Figura 8: Cliente2 atualizou o banco e Cliente1 e Cliente3 receberam a atualização.

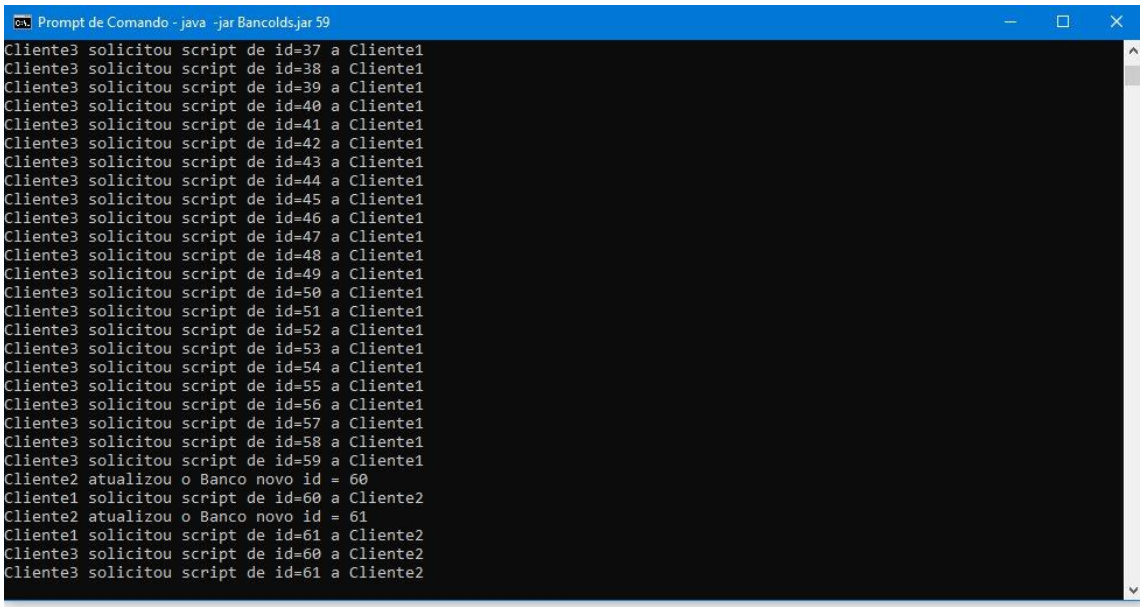

Figura 9: Cliente3 atualizou o banco e Cliente1 e Cliente2 receberam a atualização.

Todos os protótipos de telas, o esquema relacional, assim como todo o código feito por nós para o trabalho está disponível e publicado no GitHub, no link: *https://github.com/guilhermemarx14/Concessionaria*.

Conclusão

A experiência que o trabalho proporcionou a todos os membros do grupo, de podermos por em prática o que vemos em sala de aula, não só na disciplina de Sistemas Distribuídos, como também de Banco de Dados e Engenharia de Software, foi de muita importância para a bagagem de todos. Alguns problemas com que deparamos, foram resolvidos de prontidão e com trabalho em equipe, como, por exemplo, a substituição do modelo peer-to-peer para o modelo Cliente-Servidor.

Por todos esses aspectos, podemos concluir que o trabalho foi um sucesso, visto que o programa proposto pelo projeto funcionou de maneira satisfatória e atendeu aos requisitos propostos inicialmente pelos membros do grupo, mesmo com as alterações feitas com o seu desenvolvimento. Acreditamos também que esse projeto gerou uma grande bagagem de conhecimento, por tudo o que foi feito, para todos que participaram do mesmo.

Referências

- ELMASRI, R.; NAVATHE, S. B. Sistemas de Banco de Dados. 6a ed. São Paulo: Pearson, 2011.
- TANENBAUM, Andrew; STEEN, M. Van. Sistemas Distribuídos: Princípios e Paradigmas. 2a Ed, Prentice-Hall, 2008
- Slides da Professora Carla Rodrigues Figueiredo Lara, disponibilizados no Moodle UFOP no segundo semestre de 2018.