

LOCADORA DE VEÍCULOS

1 ANÁLISE

Suponha o caso de uso apresentado em seguida resultante da fase de análise de requisitos. Na fase de análise (realização dos casos de uso) sua descrição pode ser detalhada sem se preocupar profundamente com a implementação. Descreva a realização do mesmo fazendo os itens abaixo:

- a. Detalhe a descrição do caso de uso considerando que ele será voltado a funcionar na web. Os clientes devem ter meios de realizar reservas on-line:
 - i) Podem escolher a categoria de veículo ao invés de visualizar todos (passo 5).
 - ii) Podem participar de um programa de fidelidade e, neste caso, informações de cadastro (ou perfil) são utilizadas para pré-popular os formulários.
 - iii) Ao confirmar a reserva, o sistema deve mostrar os dados da mesma.
 - iv) Se o usuário informou um endereço de e-mail válido, o sistema envia-lhe a confirmação por email.
- b. Identifique as classes de análise candidatas.
- c. Filtre as classes de análise reduzindo (possivelmente) o número de classes candidatas
- d. Identifique e descreva as responsabilidades de cada classe e possíveis relacionamentos entre elas
- e. Estude a interação entre os objetos das classes
- f. Identifique possíveis atributos para as classes

Sistema para locação de carros on-line (web)

NOME: Reservar veículo

DESCRIÇÃO: Este caso descreve como um cliente usa o sistema para reservar um veículo.

PRÉ-CONDIÇÕES: o cliente está conectado ao sistema.

PÓS-CONDIÇÕES: o cliente realiza uma reserva ou não.

REQUISITOS NÃO-FUNCIONAIS: serviço acessível pela WEB.

FLUXO BÁSICO DE EVENTOS

1. O caso de uso inicia quando o ator *cliente* escolhe a opção de reservar veículo.
2. O sistema solicita ao usuário os locais, datas e horários de retirada e de devolução do veículo.
3. O cliente informa os locais, datas e horários de retirada e de devolução do veículo.

4. O sistema pergunta ao usuário qual o tipo de veículo desejado.
5. O sistema apresenta todas as opções de tipos de veículo disponíveis no local de retirada para a data e horários escolhidos.
6. O cliente escolhe um veículo para locar.
7. O sistema solicita informações de identificação ao cliente (nome completo, telefone, e-mail, bandeira do cartão de crédito, data de expiração, etc.).
8. O cliente fornece as informações de identificação solicitadas.
9. O sistema apresenta informações sobre seguros e proteções e pergunta ao cliente para aceitar ou rejeitar cada oferta.
10. O cliente informa suas escolhas.
11. O sistema solicita confirmação da reserva.
12. O cliente aceita a reserva.

Solução:

a) Pontos importantes:

- ◇ O fluxo é extremamente genérico
- ◇ Completamente independente do projeto e de tecnologias. Por exemplo, no passo 5, não há nenhuma preocupação sobre a origem das informações dos veículos que poderiam estar num sistema centralizado de banco de dados.
- ◇ Esta descrição serve tanto para um sistema que funcionará na web como numa aplicação desktop cliente-servidor.

Detalhamento (voltado ao funcionamento Web)

NOME: Reservar veículo

DESCRIÇÃO: Este caso descreve como um cliente usa o sistema para reservar um veículo por meio da Web.

PRÉ-CONDIÇÕES: o cliente está conectado ao sistema.

PÓS-CONDIÇÕES: o cliente realiza uma reserva.

FLUXO BÁSICO DE EVENTOS

13. O caso de uso inicia quando o ator *cliente* entra [na página de reserva de veículos](#).
 14. O sistema [apresenta campos](#) ao cliente para preenchimento dos locais, datas e horários de retirada e de devolução do veículo e um número de fidelidade.
 15. O cliente informa os locais, datas e horários de retirada e de devolução do veículo [e seu número de fidelidade, caso possua um](#).
- {cliente participa do programa fidelidade}**
- {solicita categoria de veículo}**
16. O sistema pergunta ao usuário qual o tipo de veículo desejado.

17.O sistema apresenta todas as opções de tipos de veículo disponíveis no local de retirada para a data e horários escolhidos. Junto a cada veículo disponível, é apresentado o valor base de locação que pode sofrer desconto segundo o perfil de fidelidade do cliente.

{visualizar detalhes do veículo}

{escolha do veículo}

18.O cliente escolhe um veículo para locar.

19.O sistema solicita informações de identificação ao cliente (nome completo, telefone, e-mail, bandeira do cartão de crédito, data de expiração, etc.). Informações existentes no perfil podem ser utilizadas para popular os campos do formulário. Alguns campos são obrigatórios, outros, opcionais.

20.O cliente fornece as informações de identificação solicitadas.

21.O sistema apresenta informações sobre seguros e proteções com os respectivos valores diários e solicita ao cliente que aceite ou rejeite cada oferta.

22.O cliente informa suas escolhas.

23.O sistema solicita confirmação da reserva.

24.O cliente aceita a reserva.

25.O sistema apresenta uma página com os dados da reserva (categoria de veículo, datas, horários e locais, proteções, custo total da locação) e uma confirmação da mesma.

{enviar confirmação}

26.O caso de uso termina.

A1: no ponto de extensão {cliente participa do programa de fidelidade}

Se o cliente participa do programa de fidelidade, recuperar perfil do mesmo e reutilizá-los nos formulários sempre que possível.

A2: no ponto de extensão {visualizar detalhes do veículo}

Se cliente desejar, escolhe a opção de visualizar os detalhes de um veículo disponível. O sistema busca as informações no inventário de veículos: cor, marca, modelo, motorização, combustível e acessórios.

A3: entre os pontos de extensão {solicita categoria de veículo} e {escolha do veículo}

Se cliente desejar restringir a busca de veículos, pode escolher uma ou mais categorias.

27.O sistema apresenta quais as categorias de veículo existentes.

28.O cliente informa a(s) categoria(s) desejada(s).

29.O sistema apresenta as categorias escolhidas disponíveis na disponíveis no local de retirada para a data e horários escolhidos.

30.O caso de uso retorna em {escolha do veículo}

A4: no ponto de extensão {enviar confirmação}

Se cliente informar um endereço de e-mail válido, o sistema envia um e-mail com os dados da reserva e a confirmação da mesma.

b, c, d) Identificar, filtrar as classes de análise e descrever as responsabilidades

Se seguir estritamente o RUP, então defina classes de fronteira, controle e entidade. Há inúmeras formas de se descobrir classes de análise:

- ◇ Conhecimento de sistemas que são similares
- ◇ Modelo do domínio
- ◇ Glossário de termos
- ◇ Sessões de CRC (Classe/Responsabilidade/Colaborador)

É possível também seguir uma abordagem “gramatical” onde substantivos (simples ou compostos) são candidatos a classes ou a atributos de classes. Abaixo, representa-se num único passo a lista de substantivos, a escolha das classes e a descrição de suas responsabilidades.

Lista de substantivos (em azul classes selecionadas)

Respondem a questão: é classe?

Acessórios: não, propriedades específicas dos veículos.

Bandeira do cartão de crédito: não, não apresenta comportamento, pode ser um atributo.

Categoria de veículo: não, critério de organização dos veículos (atributo)

Cliente: sim, armazena e trata informações sobre a pessoa que faz a locação (ex. endereço, email, telefone, etc.)

Confirmação de reserva: não, operação que pode ser efetuada sobre a reserva.

Data de devolução: não, atributo da reserva.

Data de retirada: não, atributo reserva.

Desconto de locação: não, atributo do perfil do usuário.

Endereço: endereço do cliente ou da filial da locadora, é um atributo.

Endereço email: não, atributo de cliente.

Filial (adicionado): sim, descreve de forma mais acurada local de devolução e retirada de veículos. Conhece seu endereço e possui um identificar único junto à companhia de locação. Gerencia seu inventário de veículos para determinar quais veículos estão disponíveis numa busca prospectiva por veículos.

Impostos: não, é um atributo de reserva.

Inventário de veículos: sim, controla os veículos de numa filial. Gerencia os veículos fisicamente disponíveis e disponíveis (em atraso) em função de datas e horários, locados, em manutenção, etc.

Local de devolução: não, ver filial.

Local de retirada: não ver filial.

Motorização: não, é um atributo de veículo.

Nome: não, nome do cliente que faz a locação, é um atributo do cliente.

Número de fidelidade: não, é um atributo de cliente para obter informações armazenadas.

Número de telefone: não, atributo de filial ou de cliente.

Perfil do cliente: sim (possivelmente), armazena informações específicas sobre preferências de locação do cliente. Conhece o cliente para o qual administra estas preferências.

Programa de fidelidade: sim (possivelmente), programa para contar pontos de fidelidade, contém regras para atribuir pontos para cada reserva e sabe como transformar pontos em descontos em reservas futuras.

Reserva: sim, é o contrato entre cliente e locadora. Tem várias relações com outras classes candidatas, faz a ligação entre um cliente e um veículo, com seguros e proteções, deve saber calcular seu custo total. Uma reserva identifica de forma única a associação entre cliente e veículo para uma data e hora específicas.

Valor base de locação: não, é um atributo de reserva.

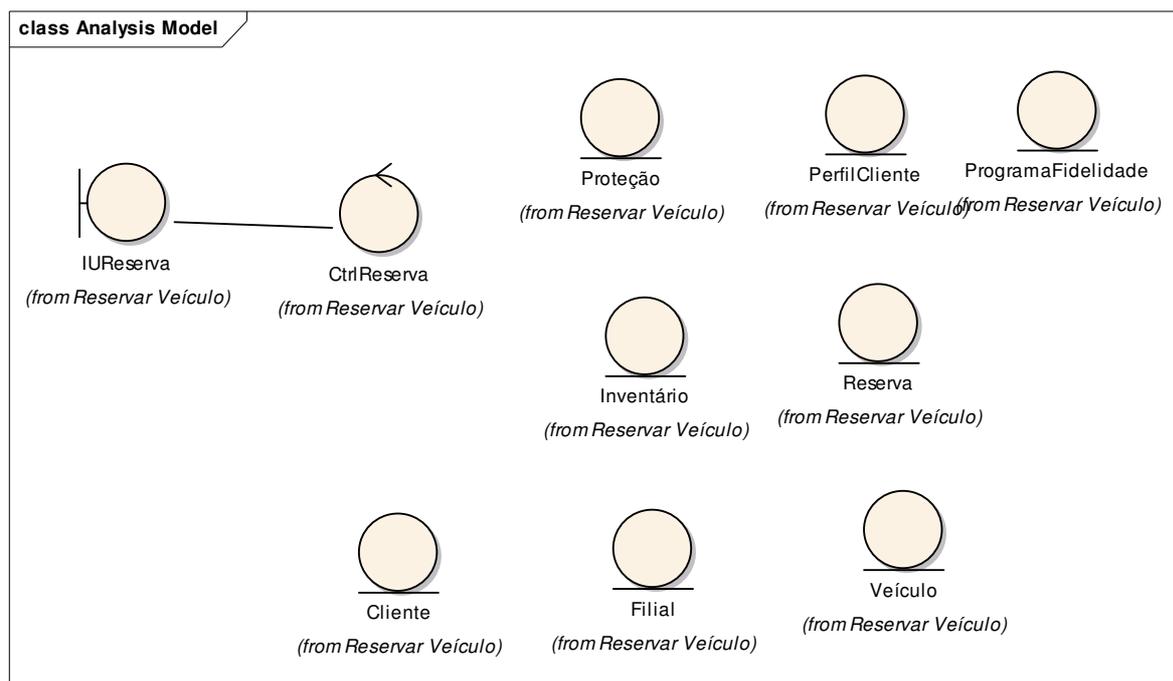
Proteção: sim, apresenta relação com outras classes (reserva e veículos) e deve saber como calcular seu preço, quais as condições aplicáveis de cauções e responsabilidades em cada estado da federação.

Veículo: sim, contém informações de um veículo que pode estar em vários estados (locado, em manutenção, disponível). Um veículo é parte de um inventário de veículos. É cliente de qual inventário faz parte e conhece sua agenda de locação.

Como saber quais substantivos indicam classes?

- ◇ É interno ao sistema?
- ◇ Apresenta comportamento?
- ◇ Apresenta estrutura?
- ◇ Tem relações com outros elementos?
- ◇ Caso responda sim a todas as perguntas, então a considere com uma classe de análise.

Aqui poderíamos montar um diagrama de classes de análise com as relações já identificadas. Retornaremos ao diagrama de classes após apresentação das relações possíveis entre classes.

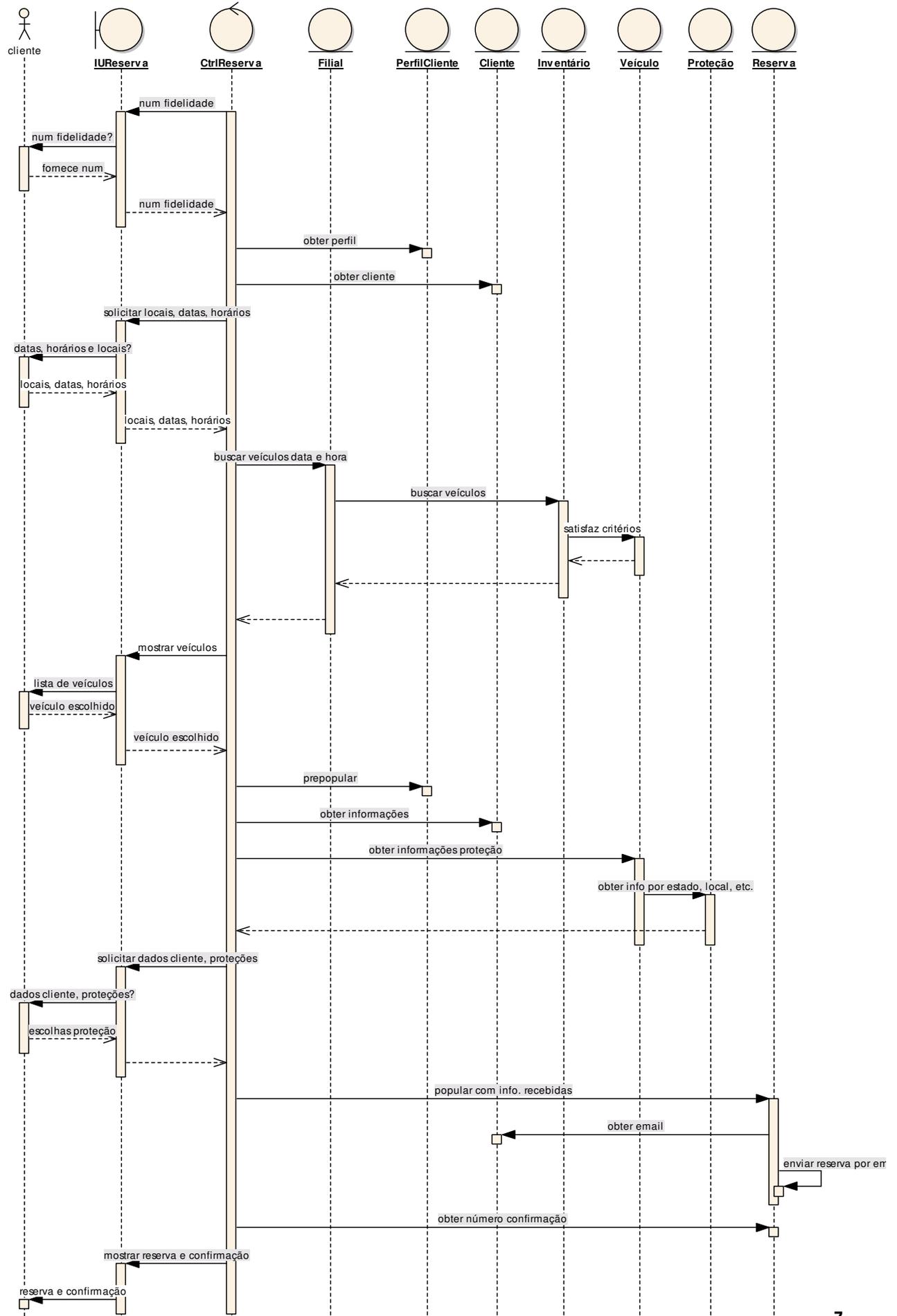


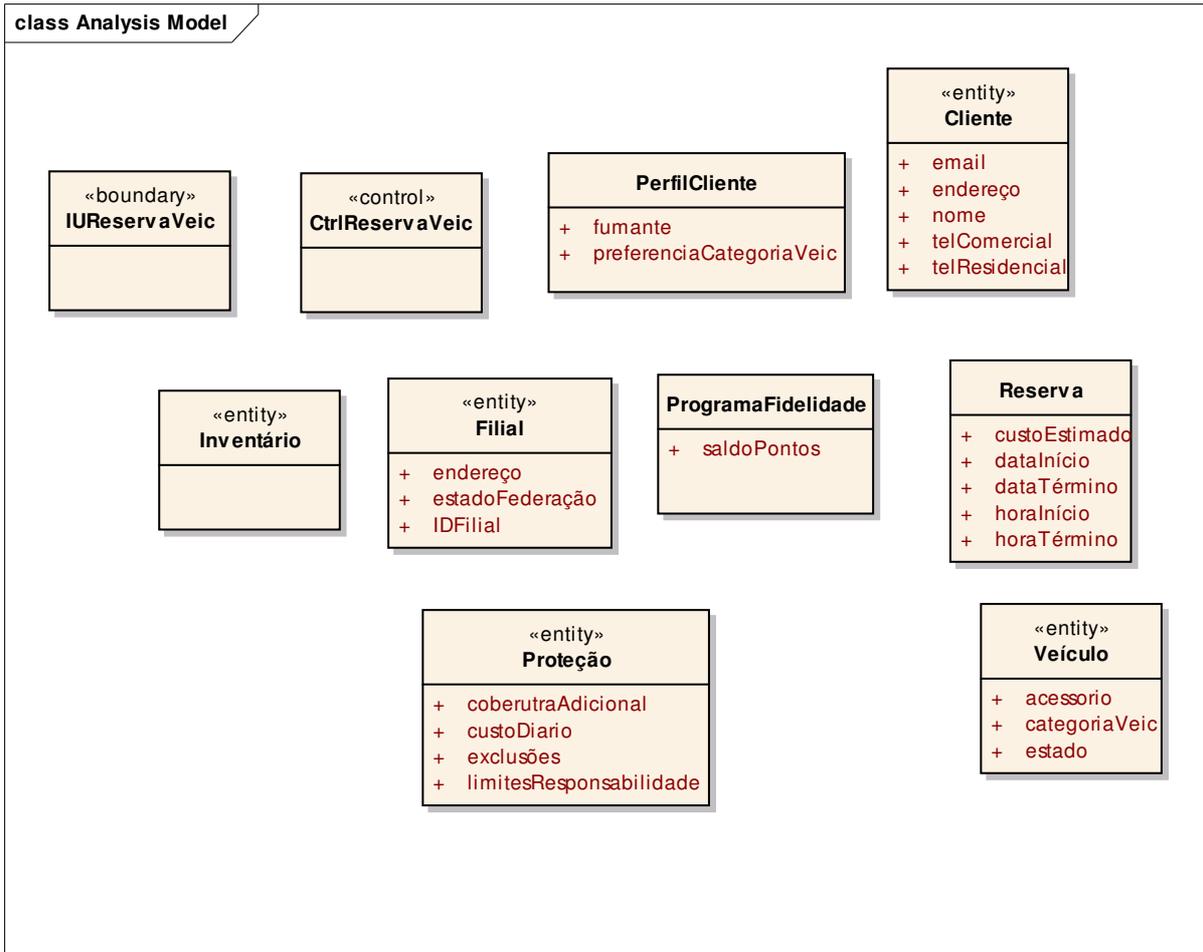
Pontos importantes:

- ◇ Alguns autores dizem que na análise é muito cedo para decidir se uma classe é de fronteira, controle e entidade. A de controle e as de entidade, a meu ver, não poluem o diagrama. A classe de interface com o usuário, no entanto poderia ser suprimida sem maiores prejuízos ao entendimento do diagrama.
- ◇ Na análise, mensagens mostram intenções e não implementação. Elas indicam o que os objetos receptores fazem, não interessa a assinatura nem se é chamada ou sinal.
- ◇ As responsabilidades definidas anteriormente (textualmente) ajudam-nos a definir quais objetos recebem quais mensagens. Por que enviar à Filial os veículos disponíveis em inventário? Por que cada filial tem um inventário associado.

f) Atributos iniciais das classes

Reservar Veículo

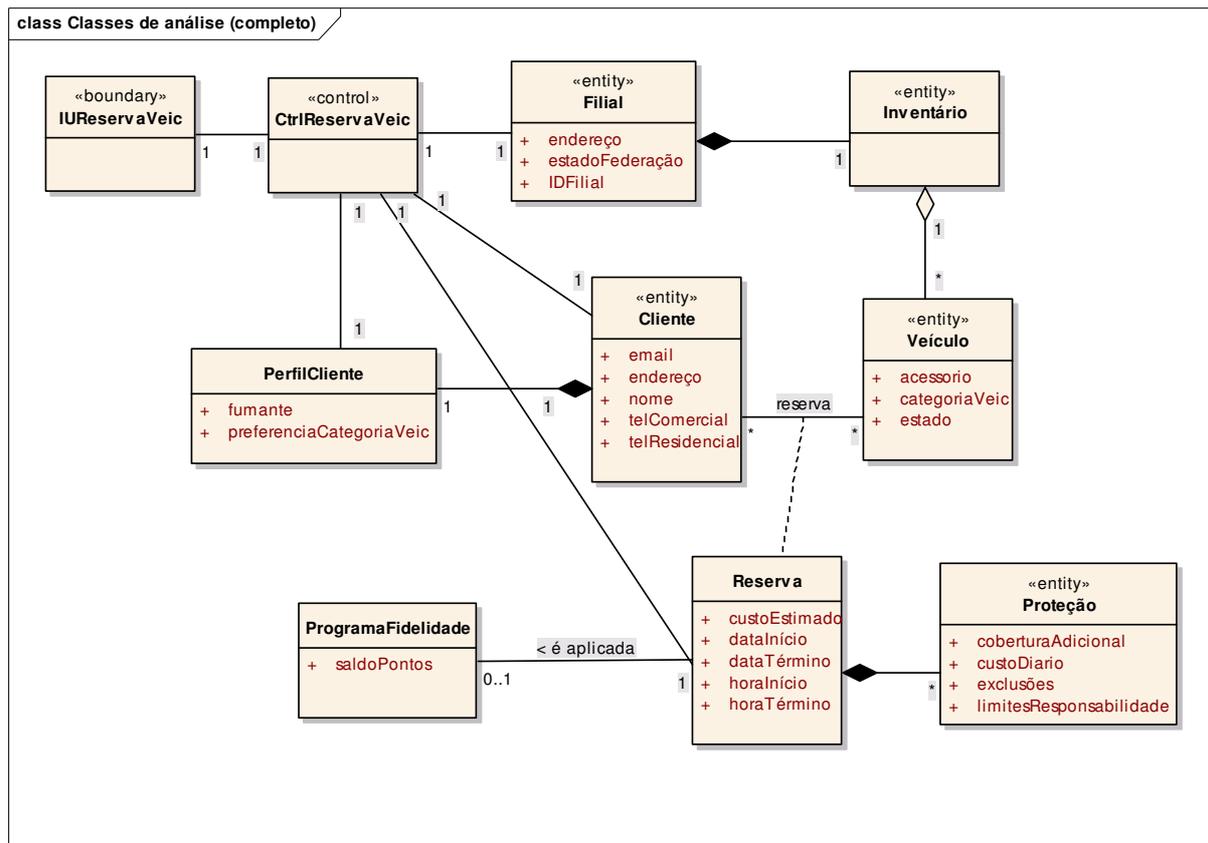




Pontos importantes

- ◇ Não é certo que todos os atributos permaneçam no projeto
- ◇ Não é certo que os atributos permaneçam como tal ou se serão classes
- ◇ Atributos da classe Inventário ainda não foram determinados (incertezas)
- ◇ Observar que não tem tipo de dado
- ◇ Observar que não tem estrutura (por exemplo, endereço)

Relações entre classes de análise (capítulo VI – exercício 8)



Para concluir: embora não tenha sido pedido no exercício é importante observar que alguns mecanismos são necessários para satisfazer as necessidades do negócio (ainda no domínio do problema e não da solução técnica).

- ◇ Por exemplo, no caso da reserva é preciso salvar (**persistir**) o objeto **Reservar** ao longo de várias sessões que podem modificá-lo, cancelá-lo, etc.
- ◇ É preciso também **segurança** na execução das transações, ou seja, somente o cliente que realizou uma reserva pode visualizá-la. Então, é preciso um mecanismo de autenticação.
- ◇ Outra necessidade é **interfacear** o sistema em desenvolvimento com o sistema legado que mantém os dados de todos os veículos da companhia.

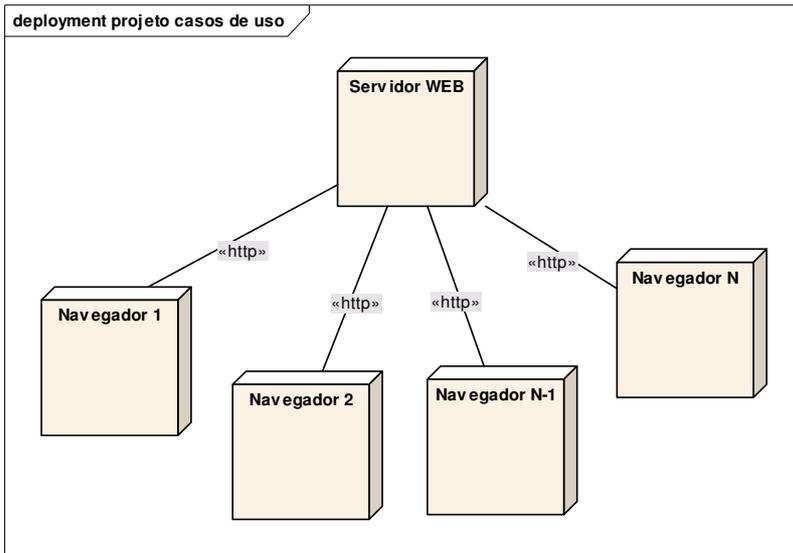
Mecanismos de análise definem o que é preciso e não como fazê-lo. Por exemplo:

Mecanismo de Análise	Mecanismo de Projeto	Mecanismo de Implementação
Persistência	RDBMS OODBMS	JDBC para SQL Server Ontos, Versant, ObjectStore
Distribuição de objetos	RMI Web Services	Java 1.4 da Sun Microsoft .Net
Interface com sistema legado	CICS (customer information control system from IBM)	SNA 3270, LU 6.2 APPC

2 PROJETO

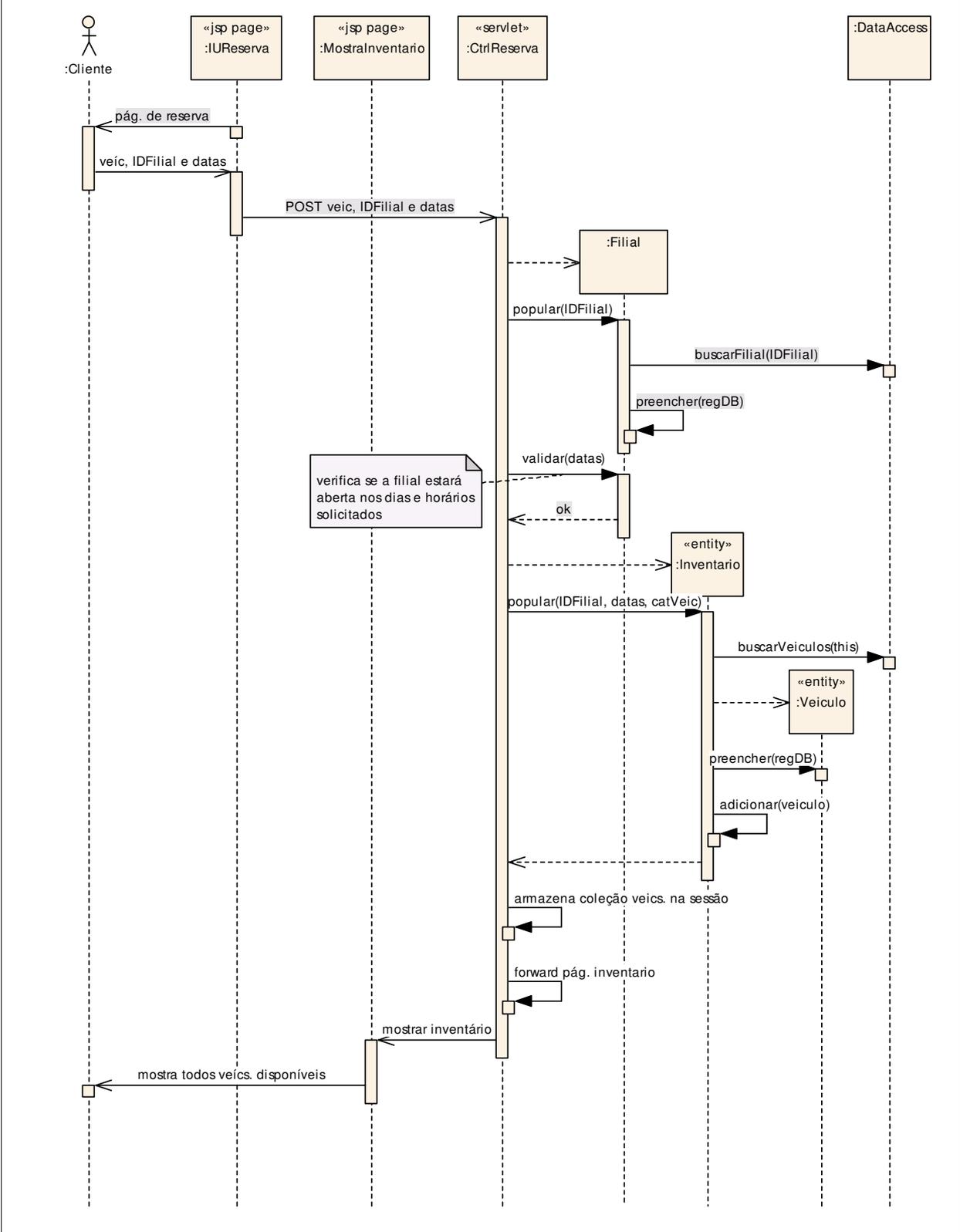
CAPÍTULO VII – Exercício 2

Para o sistema de locação de veículos, caso de uso Reservar Veículo, supor que na fase de projeto o arquiteto de software decidiu por uma arquitetura Web como mostra o diagrama de implantação. Além disso, o arquiteto de software optou por utilizar o conhecimento da equipe desenvolvedora em Java, Java ServerPages, servlets e Java Script.

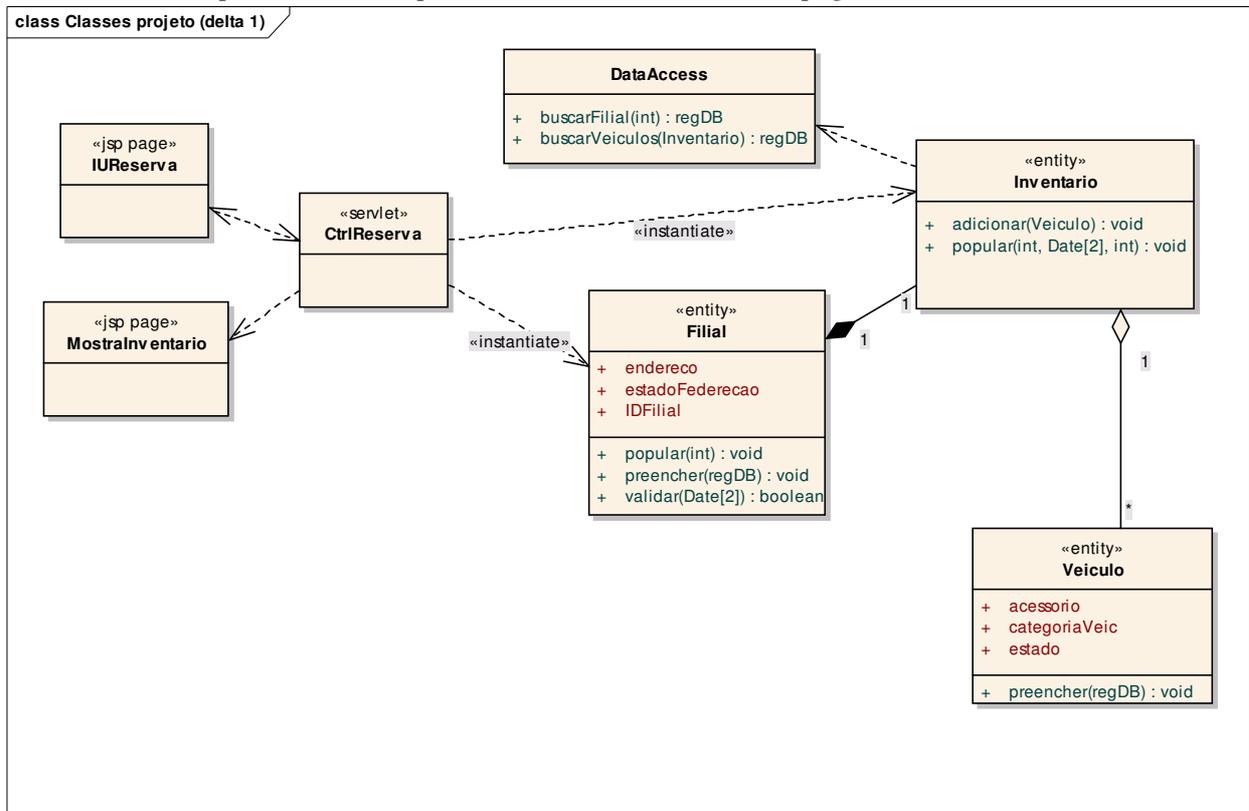


Com estas decisões, o diagrama de seqüência “reservar veículo” foi parcialmente refinado, ou seja, somente a parte de seleção de veículo por parte do cliente em um cenário onde tudo funciona como mostra a figura seguinte. Não está incluído o tratamento do perfil do cliente.

sd Reservar Veículo (prj)



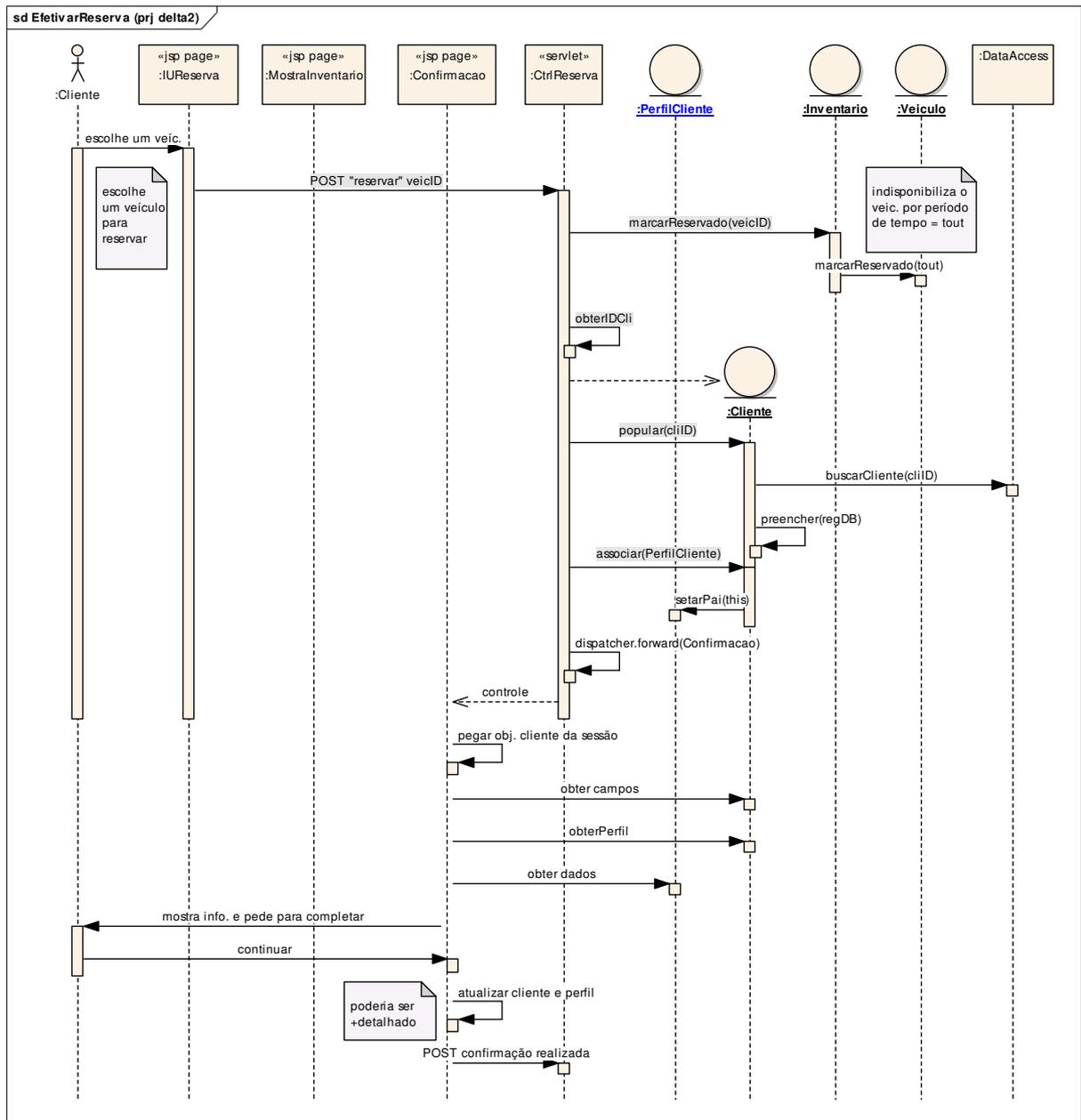
A partir do diagrama de seqüência, refinou-se o diagrama de classes. A figura seguinte mostra apenas as modificações em relação ao diagrama de classes de análise. Note que operações foram adicionadas e outras classes dependentes da arquitetura escolhida, tais como as páginas JSP.



Pede-se:

Faça o diagrama de seqüência que mostre a efetivação da reserva por parte do usuário

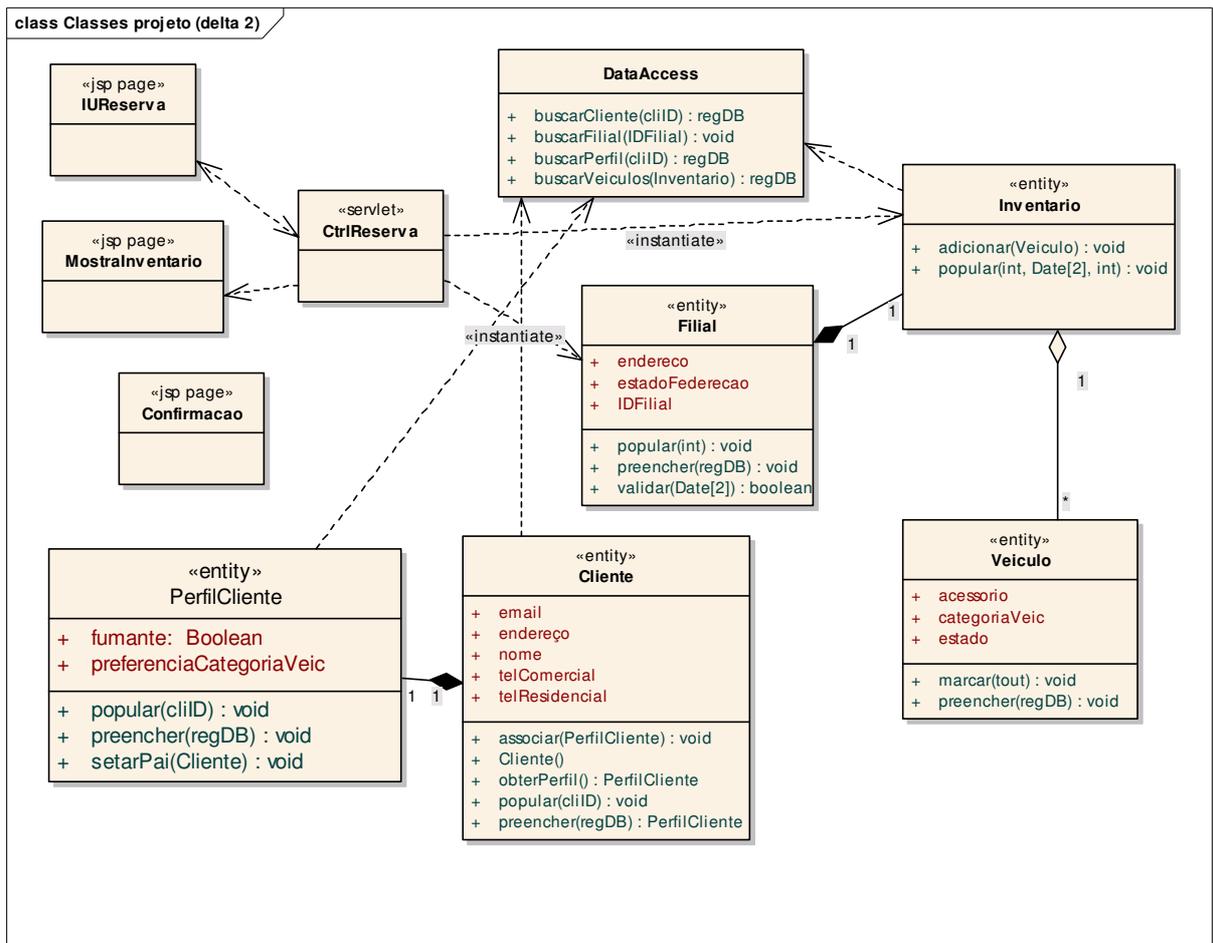
Faça as mudanças necessárias no diagrama de classes.



O diagrama de seqüência Efetivar Reserva (prj Delta 2) ainda não chega até a efetivação da reserva. É preciso detalhá-lo ainda mais.

O diagrama de classes recebeu novas operações nas seguintes classes a partir do estudo da interação entre objetos:

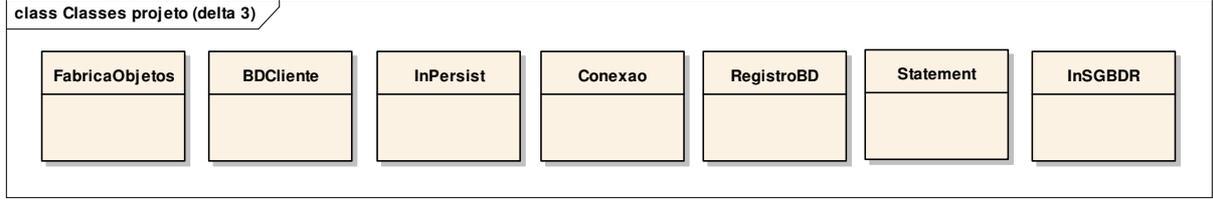
- ◇ *DataAccess*: buscarPerfil e buscarCliente
- ◇ *Veículo*: marcarReservado
- ◇ *Cliente*: popular, preencher, associar (ao perfil), obterPerfil, construtora
- ◇ A classe *Confirmação* também foi criada para mostrar os dados do cliente e do perfil e solicitar confirmação/alteração.



Outros refinamentos no diagrama de seqüência podem ser feitos. Por exemplo, alguns detalhes de acesso ao BD foram encapsulados na classe DataAccess, porém os objetos das classes PerfilCliente, Cliente e Veículo (e todas aquelas que possuem uma operação preencher(regDB)) devem conhecer a estrutura das tabelas do banco de dados para poder preencher seus atributos. Também há vários objetos que acessa o banco por meio de uma chave primária (cliID, ID Filial, etc.). Isto também não é bom, pois se a estrutura do banco muda, afetará os objetos do domínio. Para tornar os objetos do modelo independentes da camada de persistência, é possível utilizar novos objetos:

Fábrica de objetos (object factory): responsável pela instanciação dos objetos já populados com dados do banco;

Objeto proxy: um irmão gêmeo do objeto do domínio que conhece o banco de dados, ou seja, sabe como se conectar, invocar a execução queries SQL, interpretar as colunas das tabelas e mapeá-las para os atributos do objeto. Mudanças no esquema do banco de dados ou no conjunto de dados resultantes de uma query ficam escondidas atrás da fábrica de objetos e não afetam os objetos do modelo.



sd Reservar Veículo (com camada de persistência)

