Primeiro Trabalho de Técnicas de Programação 1
Locadora de Automóveis

Alberto Tavares Duarte Neto, 18/0011707
Felipe Xavier Barbosa da Silva, 18/0016326
Guilherme Chagas Suzuki, 18/0032518

[1]Dep. de Ciência da Computação – Universidade de Brasília (UnB)
CiC 117889 - Técnicas de Programação 1

1. Introdução

Dentre os 4 problemas possíveis para se implementar, foi escolhido pelo grupo o primeiro: um sistema para gerenciar locações e devoluções de carros feitas por clientes de uma determinada empresa.

As principais entidades obtidas do problema foram as classes Empresa, Agência, Cliente, Categoria, Marca, Modelo, Automóvel, Fluxo(Locação e Devolução), Locação (com prazo e sem prazo) e Devolução.

Uma importante relação do sistema é a do cliente com suas locações. Um cliente pode ter muitas locações mas cada locação terá um cliente associado. Esta locação é uma classe abstrata, e a implementação se dará pelas suas duas subclasses: locação com prazo e locação sem prazo.

Outra relação relevante considera os automóveis e as locações. Cada carro pode ter muitas relações com locações, mas cada locação terá apenas um carro associado. Tais carros relacionam-se a modelos únicos, que por sua vez relacionam-se com marcas singularmente. Além disso, esses carros podem compartilhar de categorias, que definem a qualidade do carro e o seu preço.

Essas locações têm suas devoluções registradas na forma de classes de devolução, sendo desvinculada do cliente no sentido cliente para locação. Assim, a locação e a devolução são armazenadas na forma de classes de fluxo, compondo um histórico de locações dos carros de uma agência.

Para que seja possível realizar buscas específicas, foi implementado métodos para que se busque todos os modelos de uma marca e todos os carros de um modelo. Deste modo seria possível, em outro momento, implementar um sistema para que o usuário busca e alugue seu carro de forma simples.

2. Implementação

Os casos de uso foram definidos em um diagrama apropriado, em formado UML.

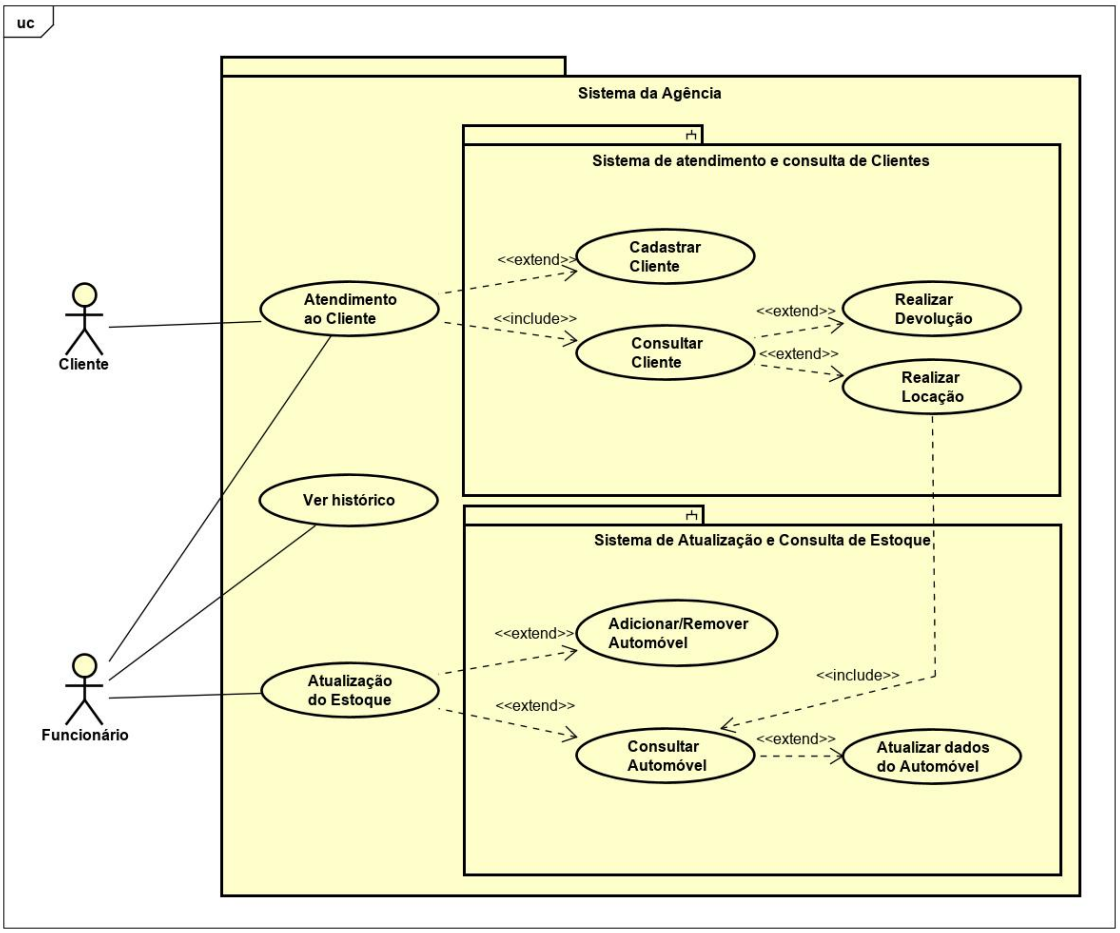

Figure 1. Diagrama de Casos e de Uso para o problema

A partir dele, foi formulada uma implementação. Essa implementação pode ser dividida em 3 partes principais:

- Sistema de registro de automóveis (Carros, Modelos, Marcas e Categorias)
- Sistema de aluguel (Clientes e Fluxos)
- Sistema geral (Empresa e Agências)

Cada uma será explicada em mais detalhes a seguir. Mas antes, foi preciso implementar uma classe de listagem geral para representar relações de um objeto com múltiplos objetos de outra classe. Nessa classe, é possível adicionar, remover e procurar por id e também é possível escolher um elemento manualmente.

Com tal classe, foi possível prosseguir na elaboração do sistema.

2.1. Sistema de registro de automóveis

Para a representação de um automóvel, são utilizadas 4 classes: Carro, Modelo, Marca e Categoria. A relação entre eles pode ser conferida no modelo abaixo:

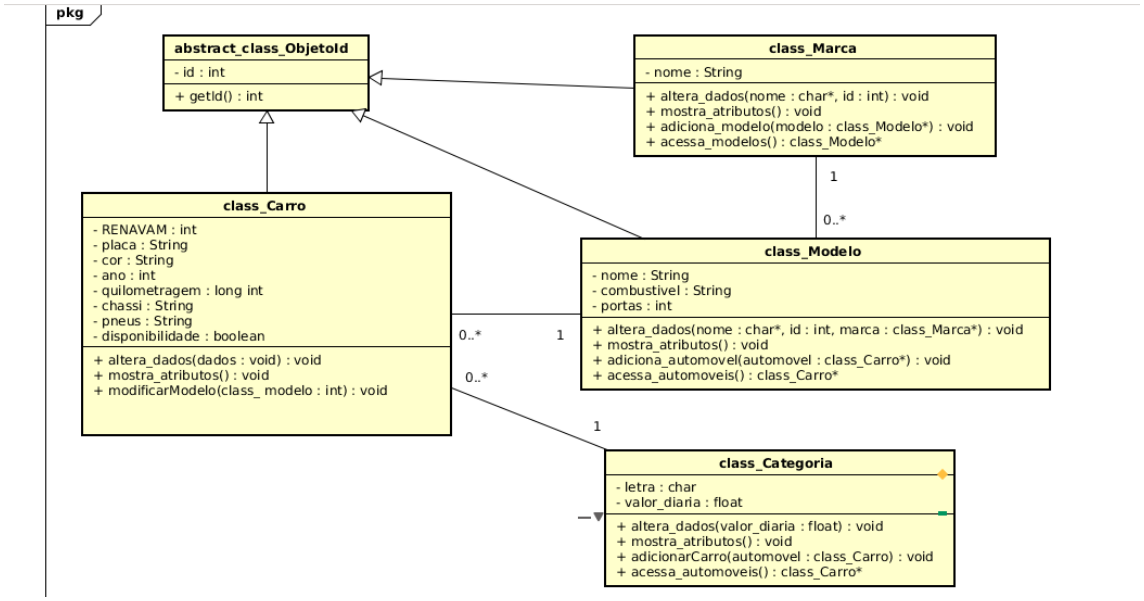

Figure 2. Diagrama de Classes

A primeira coisa a se notar é que as classes de Carro, Modelo e Marca são subclasses de ObjetoId. Como as três primeiras todas usam o id para identificação única (o que facilitaria na implementação de buscas), foi decidido, para que houvesse reutilização de código, criar essa classe abstrata ObjetoId que implementasse o get_id() e o atributo id.

Os atributos que se referem ao numero de portas e combustível foram colocados no modelo, já isso é uma característica de um modelo de um carro (diferente da placa, por exemplo, que muda pra todo carro mesmo em modelos idênticos).

A Categoria, utilizada para se definir o preço de aluguel, possui uma relação de um para muitos com Carro. Uma outra opção era associar Categoria com Modelo, porém a quilometragem pode interferir no preço do aluguel, portanto foi decidido fazer a relação com o Carro.

Na multiplicidade da relação de um para muitos entre Modelo e Carro, e entre Marca e Modelo, diz que o muitos pode ser de 0 ou mais. Isso foi feito dessa forma por simplicidade.

2.2. Sistema de Aluguel

No sistema de aluguel, o funcionário é capaz de procurar por um automóvel usando do sistema de automóveis, pela Marca, Modelo e Categoria. Além disso, pode escolher o tipo de locação. Dessa forma, com todos os dados necessários em mãos, é gerada a locação do cliente.

As possíveis locações geradas são as com prazo e as sem prazo. Locações com prazo possuem limite de tempo, então guarda-se nela o prazo, que é usado para verificar se o cliente não passou do prazo. Caso ocorra isso, ele é marcado por um booleano. As locações sem prazo tem um valor de desconto, o qual é usado para reduzir o valor final da locação quando o cliente devolver o veículo, com base na data de início e de entrega.

Evidentemente, foi criada uma superclasse Locação para associar esses dois tipos,

usada para executar um cálculo da locação de modo polimórfico. Assim, o cliente agrupa superclasses de Locação, na verdade. Quando uma locação é criada, ela também é transferida para o histórico da agência.

No caso de devoluções, uma locação a ser devolvida é selecionada e a devolução é criada a partir dos dados da locação e dos dados atuais. As devoluções possuem um valor associado, que foi ou será pago pelo cliente. Elas são registradas no histórico também.

Conforme dito, as locações e devoluções são registradas no histórico da empresa. Isso é feito de modo polimórfico, usando a superclasse comum a ambas chamada de Fluxo. Os fluxos associam horário,clientes, veículos e agência às locações/devoluções. Assim, o histórico é exibido com locações e devoluções misturadas, embora buscar por tipos específicos seja possível com ”instanceof”.

Segue um diagrama parcial do que foi falado:

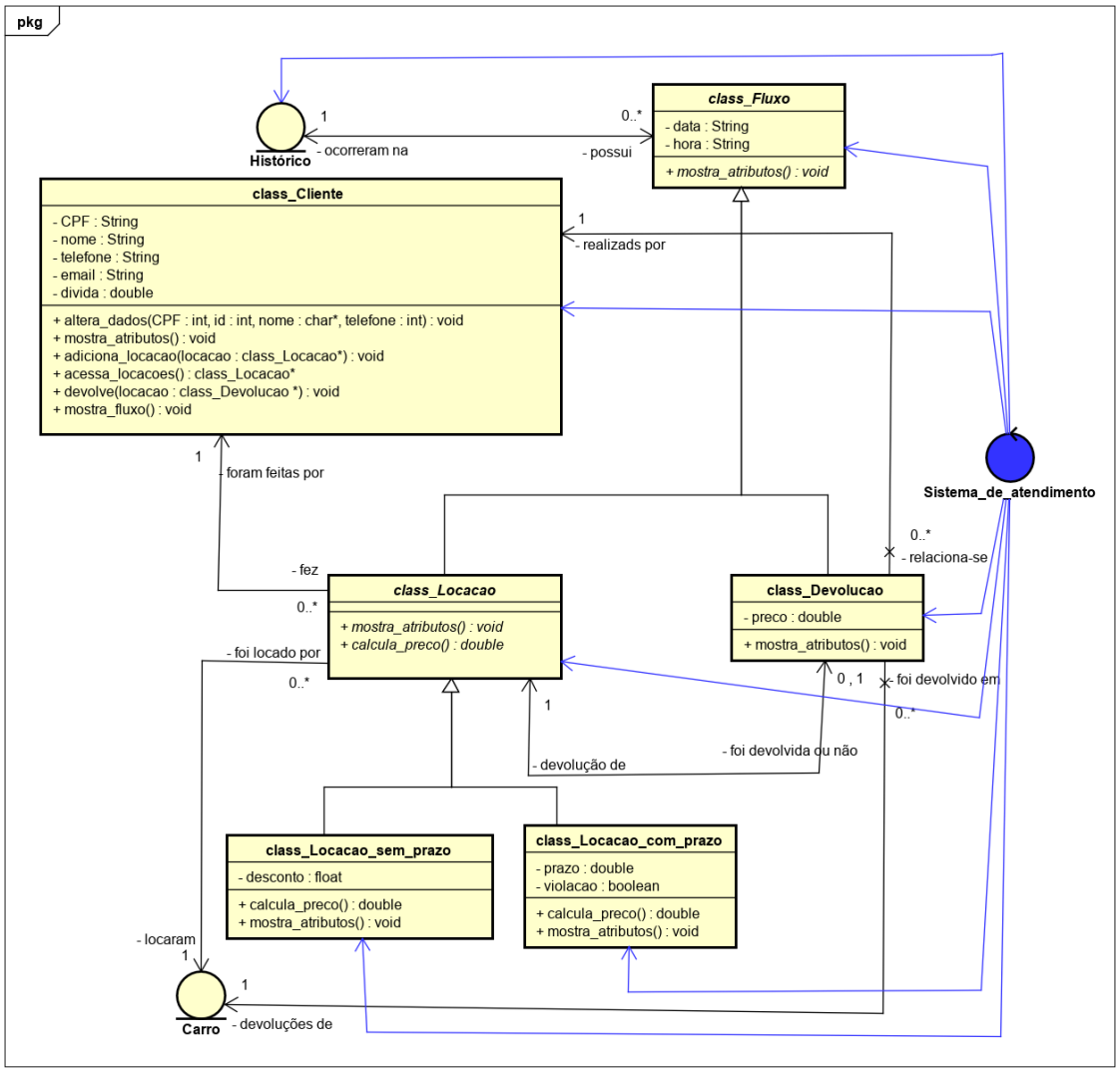

Figure 3. Diagrama de Classes com as entidades que participam do sistema de atendimento ao cliente.

2.3. Sistema geral

O sistema geral coordena os dois anteriores, sendo possível ver e alterar a lista de carros que cada agência tem disponível. Ainda, é possível ver o histórico de transações realizadas por cada agência, salvos na memória com a classe Fluxo.

Além disso, é possível manusear sistemas globais acessíveis para toda a empresa, como informações de clientes e categorias. Outra funcionalidade do sistema geral é adicionar novas marcas de carro, o que é essencial para a adição de carros nos sistemas locais das agências.

Também é possível adicionar novas agências. No momento de criação, cada agência recebe um id único, que é usado pelo sistema na hora de selecionar uma certa agência para alterar ou buscar informações.

O uso de um sistema global garante uma maior flexibilidade na locação e devolução de carros ao permitir que dados de todas as agências estejam acessíveis a cada funcionário. Com isso, um cliente pode, por exemplo, pegar um carro em uma agência e efetuar a devolução em outra.

A imagem a seguir mostra o sistema geral, representado pela classse Empresa, que administra, entre outras coisas, as agências. As agências são responsáveis pelos sistemas locais.

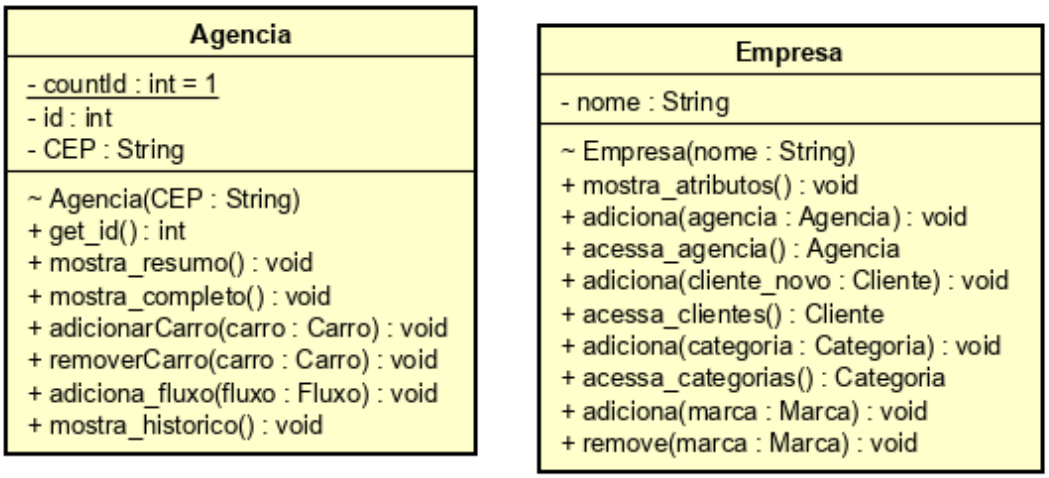

Figure 4. Classes Empresa e Agencia.

3. Conclusão

Com esta formulação do problema, foi possível elaborar um protótipo simples de um sistema de locação de veículos, que pode ser melhorado com a implementação de conhecimentos posteriores. No decorrer do trabalho, algumas dificuldades ocorreram.

Uma das maiores dificuldades enfrentadas foi definir de modo consistente as ideias e os conceitos do problema na forma de UML, para que fosse mais fácil transformá-lo

em código. Para isso, foi necessário revisar os modelos conceituais de modo que estes refletissem melhor o que se desejava fazer.

Por fim, houveram dificuldades em estabelecer métodos colaborativos eficientes, o que resultou na redução da produtividade do grupo, pois foi preciso ajustar alguns códigos devido à propagação de alterações em códigos externos.