

MODELAGEM INICIAL DA ATIVIDADE DE ANÁLISE

Iolanda Cláudia Sanches Catarino

o

Ver anotações

MODELAGEM DO ESTUDO DE CASO

Especificação dos casos de uso e das classes e, da documentação dos principais casos de uso.



Fonte: Shutterstock.

Deseja ouvir este material?

Áudio disponível no material digital.

CONVITE AO ESTUDO

O lá! Seja bem-vindo à última unidade do livro de Análise Orientada a Objetos.

Ao longo da história da engenharia de software, os modelos de processo de desenvolvimento de software evoluíram para atender satisfatoriamente aos diferentes domínios de sistemas computacionais, em consonância com as características e aplicabilidade das tecnologias de desenvolvimento de softwares.

As características do modelo de processo de software denominado de Processo Unificado (PU – *Unified Process*) foram apresentadas na primeira unidade deste livro, e nesta seção o tema será retomado, aplicado em um estudo de caso. PU surgiu para apoiar a Linguagem de Modelagem Unificada (UML – *Unified Modeling Language*), e nas demais unidades você compreendeu o objetivo e representação

das principais técnicas de modelagem estrutural e comportamental da UML, que enfatizam as características de desenvolvimento dirigido a casos de uso, de forma iterativa e incremental, conduzindo, assim, de forma sistemática e evolutiva, a modelagem de sistemas orientados a objetos com a UML. Lembre-se: a UML não faz grande distinção entre a modelagem das atividades de análise e projeto de um modelo de processo de desenvolvimento de software.

Agora, nesta unidade, para consolidar o seu aprendizado acerca da análise orientada a objetos com a UML e reforçar a aplicação e integração das principais técnicas da UML na modelagem de um sistema de informação, demonstramos a modelagem das atividades de análise e projeto correspondente à fase de elaboração do PU, referente ao estudo de caso Locadora de Veículos. Considerou-se que na fase de concepção foram definidos a ideia inicial do negócio, o domínio do problema, a delimitação do escopo e planejamento do projeto e a identificação dos atores do sistema, estabelecendo a natureza dessa interação em uma perspectiva de alto nível, e foram elencados os principais casos de uso do sistema, compreendendo assim, o entendimento dos requisitos funcionais do sistema. Para especificação das técnicas de modelagem foi adotada a ferramenta *CASE Visual Paradigm Community Edition*.

Na primeira seção desta unidade, iniciamos a modelagem do estudo de caso, apresentando a descrição e regras de negócio do estudo de caso, a especificação dos casos de uso e das classes e, ainda, a documentação dos principais casos de uso, no formato de descrição numerada, a partir da descrição dos cenários principal e alternativos.

Na segunda seção, continuamos com a modelagem do estudo de caso a partir do Diagrama de Estrutura Composta das principais colaborações do sistema e, posteriormente enfatizamos a modelagem comportamental e de interação do sistema com a especificação dos principais Diagramas de Atividades, Diagramas de Máquina de Estados e Diagramas de Sequência.

Na terceira seção da unidade, apresentamos o refinamento dos aspectos estáticos e estruturais do Modelo de Classes, complementando as classes com os tipos de dados dos atributos com as demais operações que foram identificadas na especificação dos diagramas comportamentais e de interação, além da revisão dos relacionamentos das classes, apresentando, assim, uma nova versão do Modelo de Classes. Na sequência, abordamos a persistência dos objetos, demonstrando o mapeamento das classes em tabelas de banco de dados relacional, conforme os tipos de relacionamentos estabelecidos entre as classes de objetos. Por fim, apresentamos o refinamento dos principais aspectos comportamentais da modelagem, que reflete na consistência e interação dos diagramas da UML. Assim, tratamos a modelagem da atividade de Projeto como um refinamento da modelagem de Análise, ou seja, uma extensão detalhada do Modelo de Casos de Uso e do Modelo de Classes.

PRATICAR PARA APRENDER

Caro aluno, seja bem-vindo à seção sobre a modelagem inicial da atividade de Análise de um estudo de caso!

Nesta seção vamos adotar a UML para modelagem da atividade de Análise e Projeto do estudo de caso Locadora de Veículos, e diante do conjunto de diagramas classificados em estruturais e comportamentais da UML que enfatizam a modelagem da perspectiva estática e dinâmica do sistema, iniciamos a documentação do sistema com a especificação do Modelo de Casos de Uso e do Modelo de Classes.

Você se recorda do motivo pelo qual chamamos de Modelo de Casos de Uso e não Diagrama de Casos de Uso? No contexto de modelagem de um sistema, referente a uma atividade ou fase de um processo de desenvolvimento de software, o termo modelo refere-se exatamente a um conjunto de diagramas, independentemente da técnica de modelagem, ou seja, casos de uso ou classes, por exemplo. Isso quer dizer que, dependendo do tamanho do sistema, não é possível representar todos os casos de uso ou classes em um único diagrama. Assim, o recomendado é categorizar os casos de uso ou classes, agrupando-os por assunto e elaborar mais de um Diagrama de Casos de Uso ou Diagrama de Classes. Dessa forma, por exemplo, para a modelagem dos casos de uso de um sistema ou módulo de um sistema é necessário ter mais de dois Diagramas de Casos de Uso, ou seja, os vários Diagramas de Casos de Uso compõem o Modelo de Casos de Uso.

De acordo com Booch, Rumbaugh e Jacobson (2006), um modelo é uma simplificação da realidade e pode ser estrutural, com ênfase à organização do sistema, ou comportamental, com ênfase à dinâmica do sistema, consistindo nos objetivos de ajudar a visualizar o sistema como ele é ou como desejamos que seja, proporcionar um guia para a construção do sistema, permitir especificar a estrutura ou o comportamento de um sistema, além de formalizar as decisões tomadas para o sistema. Assim, não confunda o conceito de modelo, que remete à modelagem de um sistema, com o conceito de modelo de processo.

Segundo Pressman e Maxim (2016, p. 40) “um modelo de processo fornece um guia específico para o trabalho de engenharia de software. Ele define o fluxo de todas as atividades, ações e tarefas, o grau de iteração, os artefatos e a organização do trabalho a ser feito”.

Diante da quantidade de casos de usos que foram identificados para o sistema da locadora de veículos, decidiu-se representar os casos de uso em dois diagramas, um concentrando os casos de uso correspondentes aos cadastros e movimentações de negócio, e outro diagrama mantendo os casos de uso referentes às consultas e relatórios.

Avançando a modelagem do sistema da locadora de veículos a partir do entendimento da funcionalidade de cada caso de uso, as classes de objetos do sistema foram abstraídas e elaboramos a primeira versão do Diagrama de Classes, priorizando a estrutura dos atributos e a listagem das operações básicas dos objetos de cada classe.

É importante reforçar que a especificação do Modelo de Casos de Uso e do Modelo de Classes contempla as regras de negócio definidas na descrição do estudo de caso e demais definições que foram consideradas relevantes para complementar as funcionalidades, seguindo o ponto de vista de um analista de sistemas. Assim, fique à vontade para enriquecer a solução proposta com demais funcionalidades ou adaptações que considere pertinentes ao domínio do sistema; contudo, não esqueça de respeitar as boas práticas da engenharia de software.

Para reforçar a sua aprendizagem, considere que você faz parte da equipe de desenvolvedores da empresa que está responsável pelo desenvolvimento do sistema da locadora de veículos, e você é o analista de sistema responsável pela modelagem do módulo “Pagamento” que está integrado com o módulo “Locação de Veículos”.

Considere a seguinte complementação da descrição do estudo de caso, referente ao módulo “Locação de Veículos”.

O pagamento de um aluguel é efetuado no ato da devolução do carro. A forma de pagamento de um aluguel, pode ser à vista ou a prazo, dependendo do valor total do aluguel. Para pagamento à vista em dinheiro, o valor total da devolução deve ser lançado automaticamente no caixa da empresa, assim que uma devolução é concluída. Para pagamento a prazo em cartão de crédito, deve ser gerado um controle de contas a receber (crédito a receber), correspondente ao número de parcelas, sendo necessário posteriormente tratar a baixa de uma conta a receber. Mesmo para pagamento com cartão de crédito do tipo débito é lançado como contas a receber, devido o prazo mínimo de 24 horas para ser creditado na conta da locadora

Considerando o complemento da descrição do estudo de caso e o Diagrama de Classes apresentado nesta seção para o módulo de “Locação de Veículo”, faça: elabore o Diagrama de Classes, referente ao módulo “Pagamento”, representando a integração com as classes do módulo “Locação de Veículo”.

Bom trabalho e ótimo projeto!

CONCEITO-CHAVE

A importância em documentar todo o processo de desenvolvimento de software, evidenciando principalmente as etapas iniciais do processo, foi destacada a partir da “Crise do Software”, no final da década de 1960. Independentemente da natureza de um projeto, a modelagem é essencial para delimitamos o problema em estudo e desenvolvimento, restringindo o foco a um único aspecto por vez.

A modelagem não faz parte apenas da indústria da construção civil. Para as diferentes áreas da engenharia, construímos modelos para demonstrar a definição e detalhes do projeto, auxiliando os usuários envolvidos a visualizarem qual será o produto final. Assim, na engenharia de sistemas de software a modelagem não se restringe apenas a grandes sistemas, no entanto, quanto maior e mais complexo for o sistema, maior será a importância da modelagem para diminuir a probabilidade de ocorrência de erros ou da construção dos requisitos errados.

Para compreender a arquitetura de um sistema de software, é necessário integrar várias visões de modelos que se inter-relacionam ou se complementam. Em conjunto, essas visões representam a base do projeto do software. Segundo Booch, Rumbaugh e Jacobson (2006), dependendo da natureza do sistema, alguns modelos podem ser mais importantes e adequados do que outros e podem ser especificados em diferentes níveis de precisão. Por exemplo, em sistemas que utilizam muitos dados predominarão modelos voltados para a visão estática do projeto. Em sistemas centrados no uso de *Graphical User Interface* (GUI), são importantes as visões estáticas e dinâmicas dos casos de uso, e em sistemas de execução crítica em tempo real, a visão dinâmica do processo tende a ser mais importante.

A Unidade 1 contextualizou os fundamentos e princípios da *Unified Modeling Language* (UML) e você estudou que a UML privilegia a descrição da modelagem de sistemas de software em três perspectivas principais de visões, que são: a **estrutural**, que enfatiza a visão estática do sistema, ou seja, os dados; a **funcional**, que prioriza as funcionalidades do sistema, enfatizando os requisitos funcionais; e a **temporal**, que prioriza a especificação dos eventos, representando o comportamento dos objetos em tempo de execução.

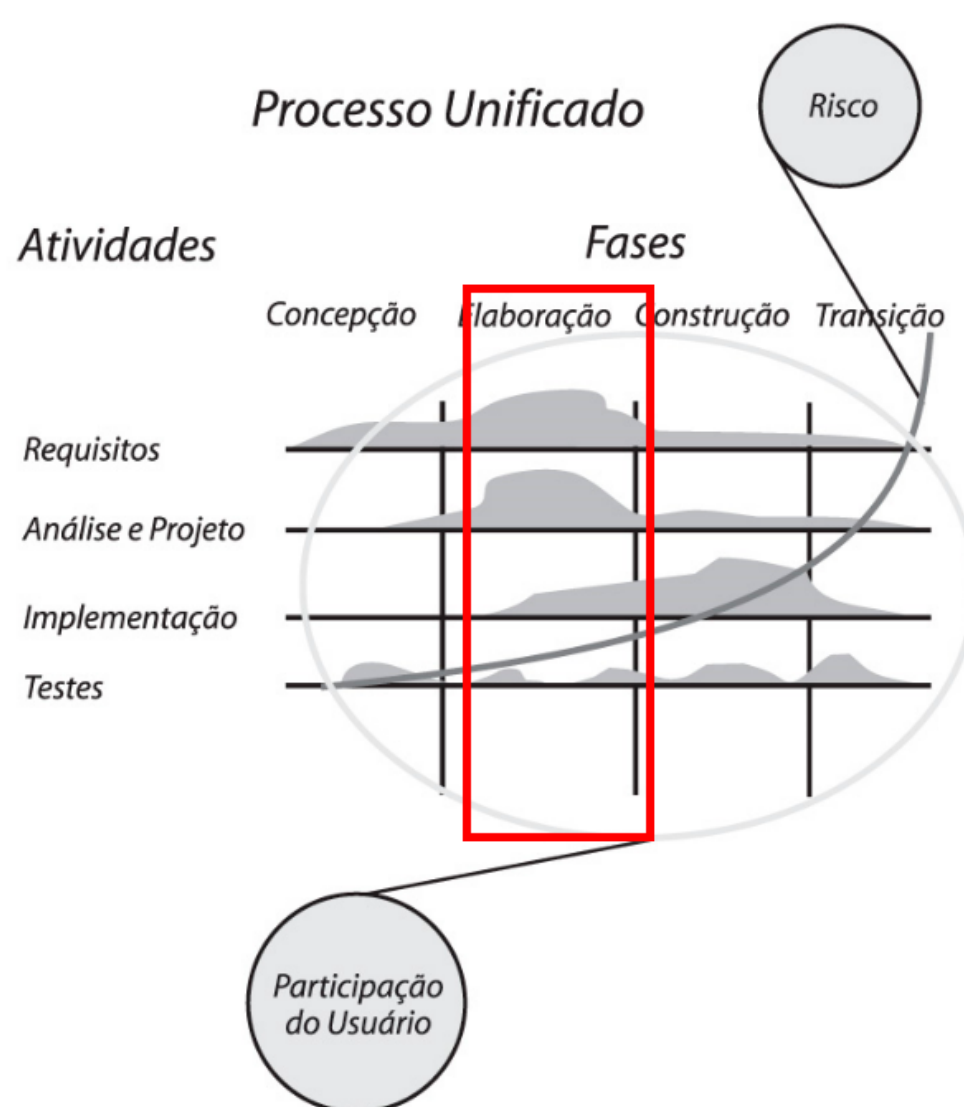
MODELAGEM DE SISTEMA COM PROCESSO UNIFICADO

Esta unidade consiste na demonstração da modelagem de um estudo de caso para o domínio de locadora de veículos, a partir da especificação das principais técnicas de modelagem da UML, para documentar as atividades de requisitos e análise e projeto, correspondentes à fase de elaboração do processo unificado.

Entre os vários modelos de processo da engenharia de software, o modelo de processo escolhido foi o Processo Unificado, considerando a sua relevante característica de manter um ciclo de vida incremental e iterativo, ou seja, o Processo Unificado apoia o desenvolvimento incremental, a partir de modelos que podem evoluir com a inclusão de novos detalhes durante a execução das atividades (*Workflow*) e, a cada ciclo do Processo Unificado, uma nova versão do produto é entregue aos usuários, conhecida como uma iteração.

A Figura 4.1 representa o ciclo de vida do Processo Unificado, indicando no eixo X a dimensão temporal distribuída nas fases de Concepção, Elaboração, Construção e Transição. Cada fase integra um conjunto de atividades – Requisitos, Análise e Projeto, Implementação e Testes, indicadas no eixo Y da figura, que são executadas de forma incremental e evolutiva, representando a entrega dos artefatos de software. Observe na figura a fase de Elaboração destacada, na qual prevalecem as atividades de Requisitos e Análise e Projeto. Assim, a documentação a seguir refere-se à especificação das principais técnicas de modelagem estruturais e comportamentais da UML, que foram adotadas na modelagem do estudo de caso proposto.

Figura 4.1 | Processo Unificado



Fonte: Medeiros (2004, p. 16).

FASE DE CONCEPÇÃO

Iniciando a modelagem do sistema denominado "Locação de Veículos", na primeira fase do Processo Unificado – Concepção, considerou-se:

1. O planejamento do projeto em consonância com a metodologia definida para o desenvolvimento do sistema.
2. A definição da ideia inicial do negócio, a partir da compreensão do domínio do problema ou domínio do sistema, ou seja, a compreensão do contexto para o qual foi provida a solução.

3. A delimitação do escopo do sistema, no qual identificam-se os processos de negócio aplicados aos requisitos do sistema.
4. O entendimento do contexto do sistema a partir da identificação dos atores do sistema e a natureza da interação entre os atores e o sistema, em uma perspectiva de alto nível.
5. A definição dos principais casos de uso do sistema, consistindo no entendimento dos requisitos funcionais do sistema.

ASSIMILE

Um processo de negócio é um conjunto de atividades ou tarefas estruturadas relacionadas que produzem um serviço ou produto específico para seus clientes. No processo de negócio, os insumos (materiais, conhecimento e outros) são transformados em resultados (produtos e serviços). Segundo Kirchoff (2015), um processo de negócio é o conjunto de atividades ou tarefas que são estruturadas e giram em torno da produção de um resultado de valor para o cliente, por meio da entrega de um serviço ou produto. Ele mostra o que deve ser realizado, como deve ser realizado e quem é o responsável.

FASE DE ELABORAÇÃO

É na fase de Elaboração que se define como o sistema será construído, a partir da especificação dos requisitos funcionais do sistema, estabelecendo a arquitetura e mecanismos para o desenvolvimento do sistema. Nessa fase prevalece a execução das atividades de Requisitos e Análise e Projeto, conforme foi destacado na Figura 4.1.

No item a seguir, apresentamos a descrição do estudo de caso e uma modelagem mínima da atividade de Análise de Requisitos, utilizando algumas técnicas de modelagem da UML.

DESCRIÇÃO DO ESTUDO DE CASO – LOCAÇÃO DE VEÍCULOS

Para complementar o seu aprendizado, utilizaremos o estudo de caso Locação de Veículos para orientar na exemplificação da modelagem dos principais diagramas da UML.

LOCAÇÃO DE VEÍCULOS

Deseja-se desenvolver um novo sistema para uma locadora de veículos que expandiu e está atuando em várias cidades do país. A locadora de veículos aluga carros aos clientes previamente cadastrados. Um cliente será descrito por: nome, data de nascimento, sexo, nacionalidade, CPF, RG, endereço completo, telefone residencial, celular, endereço eletrônico, dados da CNH (número, registro, data validade, UF), profissão e situação (ativo, preferencial, inadimplente ou encerrado). Sendo um cliente estrangeiro, além dos dados já descritos, deve-se manter também um número de documento estrangeiro e o número do passaporte. Um cliente vinculado a uma empresa conveniada (empresa com cadastro aprovado e com desconto concedido aos seus colaboradores) da locadora deve indicar o nome da empresa no seu cadastro pessoal.

Os carros devem ser cadastrados, mantendo: placa, ano, marca, modelo, características, quilometragem, valor da diária, situação (disponível, alugado, manutenção interna, manutenção externa ou inativo) e observações. Os carros devem ser classificados por categoria ou grupo, conforme o seu modelo e características, por exemplo: econômico manual hatch, econômico manual sedan, econômico automático hatch, intermediário manual sedan, executivo, SUV etc.). O valor da diária e demais valores (cobertura do veículo, cobertura de terceiro, taxa de aluguel e quilometragem excedente) variam de acordo com a categoria do carro.

Um cliente poderá realizar o seu próprio cadastro pela web ou aplicativo a partir da criação de uma conta de usuário com login (endereço eletrônico ou CPF) e senha. Um cliente ativo ou preferencial poderá alugar vários carros em datas e horários distintos, com reserva prévia.

Um cliente previamente cadastrado poderá realizar reservas por telefone, web ou aplicativo. Uma reserva deve manter os dados do cliente, categoria do carro a ser reservada, opcionais do carro (bebê conforto, cadeira de bebê e/ou assento de elevação), período da reserva, loja e hora de retirada e de devolução do carro. Cada reserva corresponde a um carro. Toda reserva realizada por um cliente deverá enviar um comprovante da reserva para o cliente via e-mail e/ou SMS.

Um cliente poderá alugar um carro e retirá-lo em uma loja e posteriormente devolvê-lo em outra loja. Para cada aluguel deverá ser registrado um código do aluguel, dados do cliente, dados da categoria do carro alugado (categoria, valores, opcionais, quilometragem atual do carro), a data e hora do aluguel e a data e hora prevista de devolução. Cada aluguel corresponde a um carro. Após a efetivação do aluguel do veículo, deverá ser emitido o contrato de aluguel e o termo de vistoria do carro, sendo que uma cópia será entregue ao cliente e uma cópia o cliente assinará e ficará de posse da locadora.

Na devolução do carro, o cliente informará a forma de pagamento, será conferida a quilometragem percorrida durante o período do aluguel, registrará a quilometragem atual do carro, além da data e hora de devolução, será calculado o valor total do aluguel e a nota fiscal de serviço será emitida e/ou enviada para o e-mail do cliente. Com a confirmação da devolução do carro, a quilometragem e situação do carro serão atualizadas, sendo a situação alterada para “manutenção interna”. Posteriormente, realiza-se a conferência e lavagem do carro e sua disponibilização para um novo aluguel.

A forma de pagamento poderá ser à vista ou a prazo, dependendo do valor total do aluguel. A transação referente ao pagamento de um aluguel será realizada pelo sistema de pagamento que será integrado ao sistema de locação.

O sistema deve disponibilizar consultas e relatórios aos funcionários e gerentes, referentes aos clientes, carros, reservas, aluguéis e devoluções.

que estão explícitos na descrição, ou seja, as funcionalidades que o sistema deve prover para os diferentes atores que interagem com o sistema. Alguns requisitos funcionais já descrevem as suas regras de negócio (políticas, condições ou restrições que devem ser consideradas na execução dos processos), como para o requisito funcional “Cadastro de Cliente” estão evidentes algumas regras, sendo “a locadora de veículos aluga carros aos clientes previamente cadastrados”; “um cliente ativo ou preferencial poderá alugar vários carros em datas e horários distintos, com reserva prévia ou não”.

■ ATIVIDADE REQUISITOS

No modelo de processo de engenharia de software – Processo Unificado, a atividade de Requisitos é a primeira atividade do ciclo de cada fase do processo. Abstrair, entender e definir os requisitos do domínio do problema é uma das

tarefas mais difíceis da engenharia de software, pois é a etapa que fundamenta e sustenta todo o processo de desenvolvimento do software.

Segundo Pressman e Maxim (2016, p. 32)

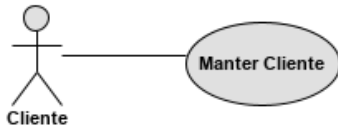
“ [...] a engenharia de requisitos é uma ação de engenharia de software importante que se inicia durante a atividade de comunicação e continua na de modelagem. Ela deve ser adaptada às necessidades do processo, do projeto, do produto e das pessoas que estão realizando o trabalho.

Para Sommerville (2018), a engenharia de requisitos preocupa-se com o que deve ser feito, ou seja, a compreensão do problema, e não em como fazer, considerando o domínio do sistema. A Engenharia de Requisitos é o processo de descobrir, analisar, documentar e verificar os serviços e restrições.

Uma primeira classificação dos requisitos de um sistema é tratada como requisitos de usuário e *requisitos de sistema*, conforme exemplifica a Figura 4.2 (SOMMERVILLE, 2018):

- **Requisitos de usuário:** são declarações em uma linguagem natural, com diagramas ou não, de quais serviços o sistema deverá fornecer a seus usuários e as restrições com as quais este deve operar.
- **Requisitos de sistema:** são descrições mais detalhadas das funções, serviços e restrições operacionais do sistema de software. O documento de requisito de sistema, denominado de especificação funcional, deve definir exatamente o que a funcionalidade deve implementar.

Figura 4.2 | Exemplo de Requisito de Usuário e Requisito de Sistema

Requisito de Usuário	Requisito de Sistema
	<ol style="list-style-type: none">1. Cliente solicita seu cadastro.2. Sistema exibe o cadastro de clientes.3. Cliente informa seus dados pessoais.4. Sistema verifica que não existe cliente cadastrado.5. Cliente informa demais dados pessoais.6. Cliente confirma o seu cadastro.7. Sistema registra o cliente.

Fonte: elaborada pela autora.

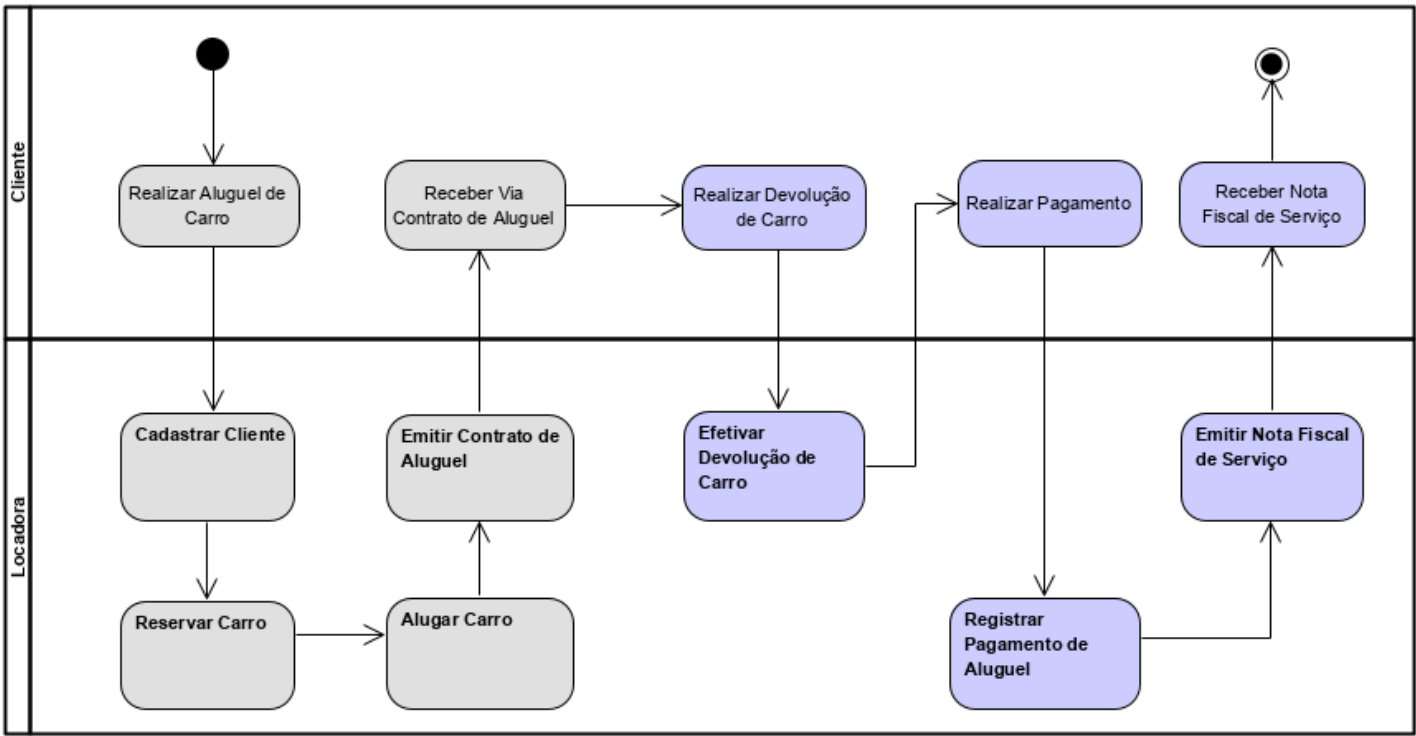
Por sua vez, os *requisitos de sistema* são classificados em dois tipos:

- **Requisitos Funcionais (RF):** são declarações de serviços que o sistema deve fornecer, de como o sistema deve reagir a entradas específicas e de como o sistema deve se comportar em determinadas situações, ou seja, representa uma funcionalidade que o sistema deve fornecer para atender a uma necessidade do usuário.
- **Requisitos Não Funcionais (RNF):** expressam restrições aos serviços ou funções, ou qualidades específicas às quais o software deve atender. As restrições referem-se a tempo, ao processo de desenvolvimento e muitas vezes são impostas pelas normas organizacionais, surgindo por meio das necessidades dos usuários, devido a restrições de orçamentos, políticas organizacionais e outros motivos. Um requisito não funcional pode estar vinculado a um requisito funcional ou pode-se aplicar ao sistema com o um todo.

Das técnicas de modelagem da UML, pode-se adotar o **Diagrama de Casos de Uso** para modelar os requisitos funcionais, que posteriormente, guiará o processo de desenvolvimento; o **Diagrama de Atividades** para representar o comportamento de cada requisito funcional do sistema, subsistemas ou de um ou mais processos de negócio do domínio do sistema; e pode-se adotar o **Diagrama de Sequência** para especificar o cenário de cada funcionalidade identificada como requisito funcional.

Para iniciar a modelagem dos requisitos do sistema “Locação de Veículos”, adotamos o Diagrama de Atividades em compartimentos, no formato de *swimlanes* (raias de natação), para facilitar a compreensão e entendimento do domínio do sistema, representando o fluxo de atividades dos principais processos de negócio que demonstram o comportamento do sistema. A Figura 4.3 ilustra o Diagrama de Atividades, demonstrando os atores “Locadora” e “Cliente” que realizam as atividades de locação e devolução de um carro de uma forma geral, sem detalhar as ações específicas que envolvem cada atividade e, ainda, considerando que o aluguel de um carro é realizado a partir de uma reserva.

Figura 4.3 | Diagrama de Atividades – Locação e Devolução de Carro



Fonte: elaborada pela autora.

A partir da descrição do estudo de caso e da representação das atividades gerais que consistem o processo geral de locação e devolução de um carro a partir de uma reserva prévia, conforme ilustrou a Figura 4.3, foram identificados os principais **requisitos funcionais** relacionados no Quadro 4.1 a seguir. No quadro foram listados apenas os requisitos funcionais referentes aos cadastros e controles dos processos de locação e devolução. As consultas e relatórios não foram listados, mas serão representados diretamente no Diagrama de Casos de Uso, na modelagem da próxima atividade.

Quadro 4.1 | Listagem dos Requisitos Funcionais

Nº	Requisito	Descrição
RF1	Cadastro de filial	O sistema deve prover um cadastro de filiais da locadora de veículos. O cadastro será mantido pelo administrador do sistema.
RF2	Cadastro de cliente	O sistema deve prover um cadastro de clientes. Um cliente pode se cadastrar via web ou aplicativo. Cada cliente deve ser gerenciado pela sua situação (ativo, preferencial, inadimplente ou encerrado).
RF3	Cadastro de empresa	O sistema deve prover um cadastro das empresas conveniadas com a locadora de veículos.
RF4	Cadastro de grupo de carro	O sistema deve prover um cadastro de grupos de carros para categorizar os carros por características.

Nº	Requisito	Descrição
RF5	Cadastro de carro	O sistema deve prover um cadastro dos carros disponíveis para o aluguel. Cada carro deve ser gerenciado pela sua situação (disponível, alugado, manutenção interna, manutenção externa ou encerrado).
RF6	Controle de reserva de carro	O sistema deve prover um controle de reserva de carros para clientes da locadora. Uma reserva pode ser realizada por telefone, web ou aplicativo. Cada reserva deve ser gerenciada pela situação (agendada, cancelada, realizada, aprovado ou reprovado). Toda reserva efetivada deve enviar um comprovante da reserva para o cliente via e-mail e/ou SMS.
RF7	Controle de aluguel de carro	O sistema deve prover um controle de aluguel de carros para clientes da locadora.
RF8	Emissão de contrato de aluguel	O sistema deve emitir cópias do contrato de aluguel.
RF9	Controle de devolução de carro	O sistema deve prover um controle de devolução de carros. Uma devolução refere-se à baixa de um aluguel. O controle de devolução estará integrado com o módulo de pagamento.
RF10	Emissão de nota fiscal de aluguel	O sistema deve emitir a nota fiscal de serviço, referente ao aluguel de um carro. A nota fiscal pode ser impressa ou enviada por e-mail para o cliente.

Fonte: elaborado pela autora.

ATIVIDADE ANÁLISE E PROJETO

Na segunda atividade do modelo Processo Unificado – Análise e Projeto, a modelagem da atividade de **Análise** consiste em identificar e definir o que o sistema de software deve fazer em uma visão lógica do negócio e a atividade de

Projeto consiste em definir como será o desenvolvimento do software, em consonância com as tecnologias de linguagens de programação e banco de dados que serão adotados para implementação do software.

Em um processo de desenvolvimento iterativo a atividade de Análise precede o a atividade de Projeto, entretanto, a modelagem da análise e o projeto podem acontecer simultaneamente. Segundo Bezerra (2007, p. 176)

“

[...] os modelos especificados na análise esclarecem o problema a ser resolvido. No entanto, as perspectivas do sistema fornecidas por esses modelos não são suficientes para se ter uma visão completa do sistema para que a implementação comece.

Na atividade de análise, inicia-se a modelagem com a definição dos casos de uso, a partir dos requisitos funcionais identificados na atividade de Requisitos, especificando o Modelo de Casos de Uso. Posteriormente, considerando que o Modelo de Casos de Uso está pronto, a próxima etapa é analisar cada caso de uso e iniciar a identificação das classes de objetos, compreendendo qual classe ou quais classes participam da realização de um caso de uso e como o sistema será estruturado internamente, especificando o Modelo de Classes geralmente em várias perspectivas de visão. A partir da primeira versão do Modelo de Classes, é mais fácil complementar a modelagem dos casos de uso com a documentação descritiva de cada caso de uso, pois nesse momento já se identificou os objetos que colaboram e devem participar da execução de cada caso de uso.

MODELO DE CASOS DE USO

A partir da definição dos requisitos funcionais analisamos os requisitos identificados e abstraímos as regras de negócio do sistema, e assim foi decidido estruturar os casos de uso em dois pacotes.

Na UML, os agrupamentos que organizam um modelo (conjunto de diagramas) são chamados de pacotes.

DIAGRAMA DE PACOTES

A técnica de modelagem estrutural – **Diagrama de Pacotes** demonstra os elementos do sistema agrupados e organizados em pacotes lógicos ou físicos, com o objetivo de representar os componentes ou módulos que integram um sistema e suas dependências. Para Bezerra (2014), um pacote é um mecanismo utilizado

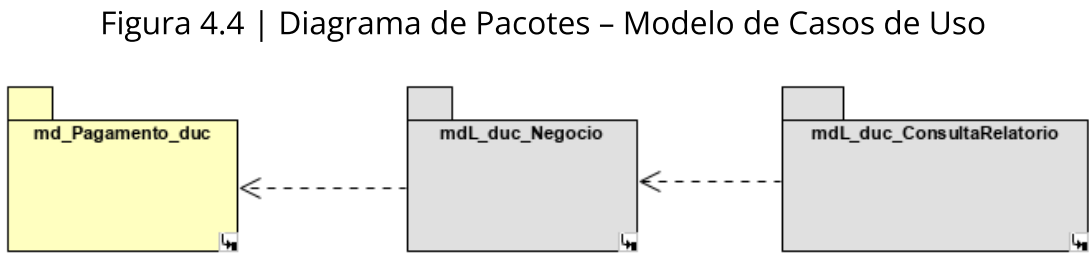
para agrupar elementos semanticamente relacionados, inclusive outros pacotes, sendo que as ligações entre pacotes são indicadas pelo relacionamento de dependência entre pacotes.

ASSIMILE

O Diagrama de Pacotes tem uma notação gráfica simplificada que demonstra como os elementos do sistema estão organizados em pacotes e suas dependências, a partir da categorização dos elementos em grupos que representam as partes do sistema, sendo que um pacote pode se subdividir em outros pacotes. O diagrama pode ser especificado de maneira independente ou associado a outros diagramas da UML principalmente para representar o Modelo de Casos de Uso e o Modelo de Classes.

Antes de iniciar a especificação dos casos de uso, é importante listar todos os casos de uso identificados, para tomar a decisão de agrupá-los ou não por categoria ou assunto e, com isso, desenhar o Diagrama de Pacotes e o Diagrama de Casos de Uso correspondentes a cada pacote, compondo o Modelo de Casos de Uso.

A Figura 4.4 exemplifica o Diagrama de Pacotes dos casos de uso. Foram definidos dois pacotes – “mdL_duc_Negocio” e “mdL_duc_ConsultaRelatorio”, correspondente ao módulo locação, e o pacote “md_Pagamento_duc”, correspondente ao módulo pagamento, que foi integrado ao modulo locação que está sendo especificado. Assim, os pacotes representados com suas dependências correspondem aos Diagramas de Casos de Uso que agruparam os casos de uso por assunto, com o objetivo de organizarem os casos de uso identificados, constituindo um Modelo de Casos de Uso.



Fonte: elaborada pela autora.

A Figura 4.5 ilustra o Diagrama de Casos de Uso correspondente ao pacote “mdL_duc_Negocio”, ilustrando os casos de uso correspondentes aos requisitos funcionais e outros casos de uso que foram identificados a partir da abstração desses requisitos, considerando também o contexto do domínio do sistema. Foram

definidos os atores primários “Funcionário Adm”, “Cliente”, “Empresa” e “Gerente”, que interagem com os casos de uso, ou seja, o analista de sistemas deve identificar a pessoa, setor da empresa, um disposto ou outro sistema de software responsável em fornecer os dados para que ocorra a execução do caso de uso, caracterizando a interação com o sistema. É importante também estudar cada requisito funcional definido na atividade anterior e, assim, tomar a decisão de estabelecer os casos de uso e seus relacionamentos, pois um requisito funcional pode ser representado por um ou mais casos de uso. Recomendamos pensar no objetivo de cada caso de uso, considerando a usabilidade do sistema para os usuários.

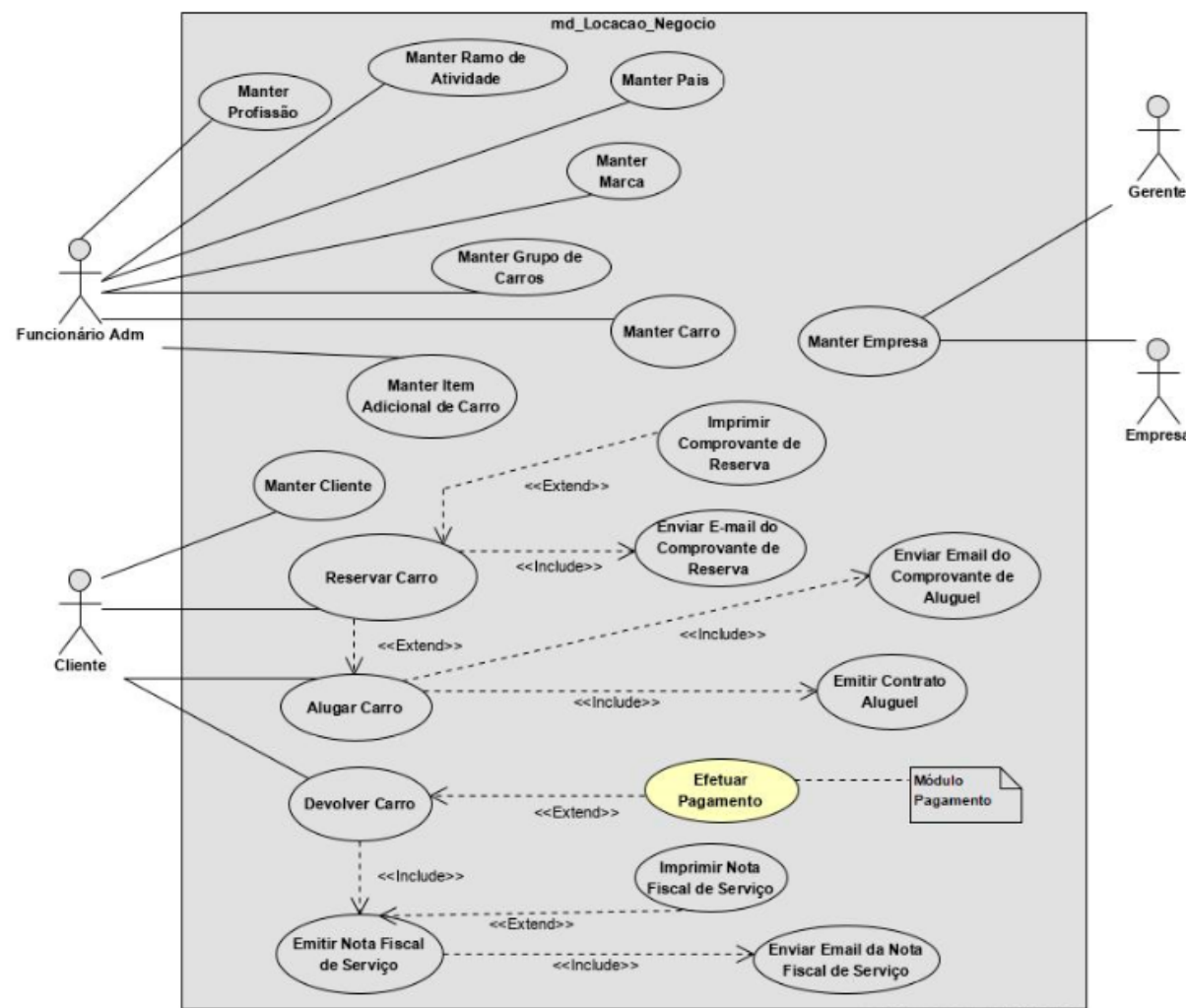
A Figura 4.5 consiste na representação dos casos de uso base (casos de uso associados a atores) e alguns relacionamentos entre os casos de uso já foram definidos porque completam o papel da funcionalidade, considerando a sua execução. Observa-se essa situação no caso de uso “Reservar Carro”, pelo qual, a partir de uma reserva que um cliente realiza, o cliente receberá um comprovante por e-mail da reserva efetuada automaticamente pelo sistema, sendo representado no diagrama pelo relacionamento de inclusão <<Include>>, e se for de interesse do cliente, ele poderá imprimir o comprovante da reserva efetuada após finalizar a confirmação de reserva, sendo representado no diagrama pelo relacionamento de extensão <<Extend>>.

Lembre-se de que o relacionamento de extensão é um tipo de relacionamento de dependência existente somente entre casos de uso. O caso de uso estendido é uma descrição completa de uma sequência de interações, com significado completo de uma outra funcionalidade do sistema. O relacionamento de dependência, representado pelo estereótipo <<Extend>>, é indicado por uma seta que parte do caso de uso “estendido” para o caso de uso “base”, que é o caso de uso que tem interação com o ator. E o relacionamento de inclusão é um tipo de relacionamento existente somente entre casos de uso para indicar a continuidade de execução obrigatória entre os casos de uso. O relacionamento de dependência, representado pelo estereótipo <<Include>>, é demonstrado por uma seta direcionada do caso de uso “base” para o caso de uso “incluído”.

REFLITA

Se os tipos de relacionamento de extensão e inclusão são estabelecidos apenas entre os casos de uso, como podemos identificar os tipos de relacionamentos de extensão e inclusão entre os casos de uso?

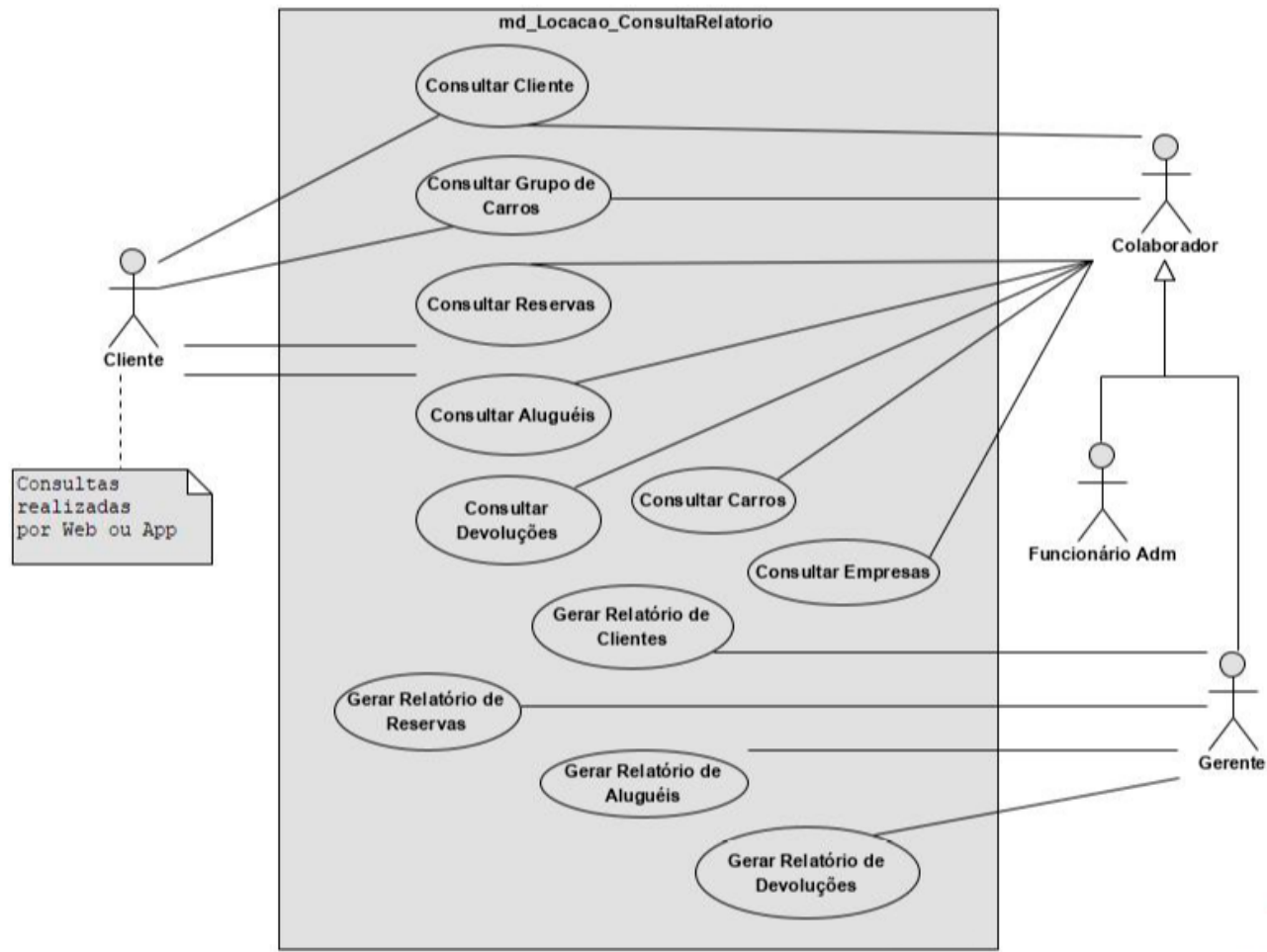
Figura 4.5 | Diagrama de Casos de Uso – Pacote mdL_duc_Negocio



Fonte: captura de tela do software Visual Paradigm Community Edition elaborada pela autora.

A Figuras 4.6 representa o Diagrama de Casos de Uso correspondente ao pacote “mdL_duc_ConsultaRelatorio”, que concentra os casos de uso definidos para as funcionalidades das consultas e relatórios operacionais do sistema. Perceba que alguns casos de uso estão associados ao ator genérico “Colaborador”, o qual indica que os atores especializados “Funcionário Adm” e “Gerente” têm acesso a todos os casos de uso associados ao ator genérico.

Figura 4.6 | Diagrama de Casos de Uso – Pacote mdL_duc_ConsultaRelatorio



Fonte: captura de tela do software Visual Paradigm Community Edition elaborada pela autora.

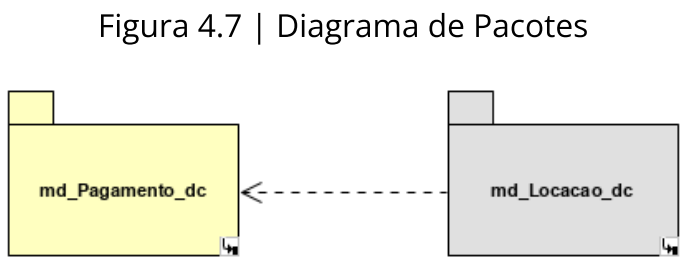
DIAGRAMA DE CLASSES

A partir da abstração dos casos de uso, iniciou-se a identificação das classes de objetos e a elaboração do Diagrama de Classes, que é considerada a principal técnica de modelagem estrutural da UML, representando a modelagem da parte estática do sistema e simbolizando um conjunto de classes com seus atributos, operações e relacionamentos.

Entre as diferentes técnicas de identificação de classes, como a da Análise Textual de Abbott; a da Análise dos Casos de Uso; a de Cartões Classes, Responsabilidades e Colaboradores (CRC); a da Taxonomia de Conceitos e outras; adotamos a técnica da Análise dos Casos de Uso. Assim, para cada caso de uso definido no Modelo de Casos de Uso, analisa-se qual ou quais objetos é preciso criar ou acessar durante a execução do caso de uso, ou seja, em tempo de execução do sistema. Por exemplo, se foi definido o caso de uso “Manter Cliente” é porque o sistema deve permitir a manipulação dos objetos do tipo cliente com as operações básicas de inclusão, alteração, exclusão (ou controlar o status do objeto, ativo ou não) ou pesquisa, assim é necessário definir uma classe para esses objetos com os atributos e operações pertinentes ao contexto do domínio do sistema. Entretanto, pode acontecer de um ou mais casos de uso manipularem objetos da mesma classe ou vice-versa.

Após a identificação das classes e atributos deve-se verificar a consistência entre as classes e os casos de usos já definidos previamente e, assim, observar a necessidade de novas classes ou eliminar incoerências e redundâncias. A partir da elaboração de uma primeira visão do Diagrama de Classes, deve-se refiná-lo e incrementá-lo com novos detalhes correspondentes às tecnologias de implementação que serão adotadas, especificando o modelo ideal do Diagrama de Classes da atividade de Projeto.

Considerando o contexto do sistema de “Locação de Veículos”, decidimos organizar as classes em um único pacote “md_Locacao_dc” com dependência com o pacote “md_Pagamento_dc”, referente ao módulo de pagamentos, conforme ilustra o Diagrama de Pacotes da Figura 4.7.



Fonte: elaborada pela autora.

A Figura 4.8 representa o Diagrama de Classes de análise correspondente ao pacote “md_Locacao_dc”, respeitando as regras básicas de modelagem do diagrama e as regras de negócio do sistema.

No Diagrama de Classes, além da representação das classes, forma estabelecidos os relacionamentos entre as classes, os quais indicam o compartilhamento de informações entre os objetos das classes por meio da troca de mensagens entre os objetos, em tempo de execução do sistema. Observa-se no diagrama da Figura 4.8 o relacionamento do tipo generalização, estabelecido entre a superclasse “Pessoa” e as subclasses “PessoaFisica” e “PessoaJuridica”, e todos os demais relacionamentos estabelecidos entre as classes são do tipo associação binária.

A classe “FilialLoja” está destacada em outra cor e não foram listados os atributos e operações porque é uma classe que existe no sistema, mas seus objetos são criados e manipulados diretamente no banco de dados pelo administrador do sistema. Há necessidade, contudo, de representar essa classe para estabelecer a associação com as classes “Reserva” e “AluguelDevolucao”.

Figura 4.8 | Diagrama de Classes (Análise) – md_Locacao_dc

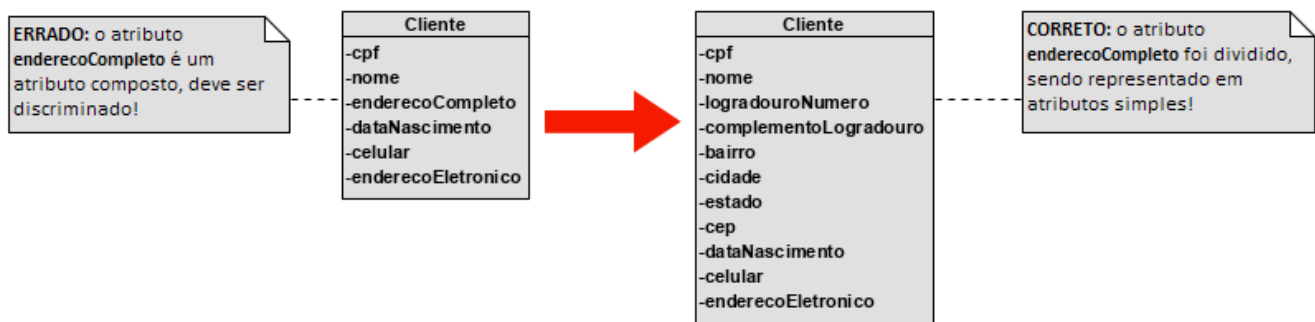
REFLITA

Na definição dos relacionamentos do Diagrama de Classes, existe uma regra ou padrão para definir a multiplicidade do relacionamento do tipo associação?

Lembre-se da nomenclatura recomendada para definir o nome das classes, de seus atributos e operações. Na primeira parte exibe-se o nome da classe, geralmente um substantivo ou expressões breves, devendo ser único no modelo de classes. Por convenção, o nome é apresentado no singular e quando há palavras compostas usa-se concatená-las, começando cada palavra por letra maiúscula. Na segunda parte, são declarados os atributos que representam os dados do objeto. São nomeados por substantivos ou expressões que representam alguma propriedade da classe, tipicamente em letra minúscula. Quando há palavras compostas usa-se concatená-las, sendo que a partir da segunda palavra o termo é iniciado com letra maiúscula, por exemplo: *dataNascimento* e *endereçoEletrônico*. Na terceira parte são declaradas as operações que correspondem às ações dos objetos, nomeadas por um verbo ou uma locução verbal breve, usando a mesma convenção de letras minúsculas e maiúsculas dos atributos, como *criarCliente*, *alterarCliente*, *validarDataNascimento* e *calcularIdade*.

Um ponto de atenção que vale a pena reforçar é a definição e listagem dos atributos de uma classe. Lembre-se de que todo atributo definido para uma classe tem que ser um atributo atômico – também denominado de simples –, ou seja, não é possível dividir o atributo em outros campos, como CPF, estado civil, placa, ano do modelo etc. Diferentemente de um atributo composto, por exemplo, filiação, que é composto por nome da mãe e nome do pai; endereço, que é composto de nome do logradouro, número do logradouro, complemento, bairro, cidade, estado e CEP; período da inscrição de um evento, composto de data de início e data de término. Então, em uma classe, está errado listar um atributo composto, conforme mostra a Figura 4.9.

Figura 4.9 | Classes com Atributo Composto e Atributo Simples



Alguns atributos compostos podem ser definidos como um atributo simples na representação de uma classe, como no atributo nome de uma pessoa. Nesse caso, é necessário também se atentar ao contexto do domínio do sistema, ou seja, às vezes, para o domínio de um sistema, é possível considerar o atributo nome de uma pessoa como sendo um atributo simples, mantendo em um único atributo o nome e sobrenome da pessoa; e para outro contexto, é importante definir o atributo nome de uma pessoa separado do sobrenome da pessoa.

EXEMPLIFICANDO

Um tipo de associação importante na definição dos relacionamentos entre as classes do Diagrama de Classes é o tipo chamado Agregação, conhecido como associação “Todo-Parte”. Demonstra que as informações de um objeto (objeto-todo) precisam ser complementadas pelas informações contidas nos objetos da outra classe (objetos-partes) relacionada, representando que ambos os objetos das classes podem “viver” de forma independente, ou seja, não existe uma ligação forte entre eles; contudo são partes de um único todo.

A notação da agregação é representada por um losango na extremidade da classe que contém o “objeto-todo”, conforme mostra o exemplo da Figura 4.10 a seguir, demonstrando que um objeto “Pais” é composto por nenhum ou vários objetos “Estado” e um estado é composto por nenhum ou vários objetos “Cidade”, ou seja, eu posso criar um objeto “Pais” e posteriormente, em outro momento, criar o objeto que representa as suas partes, o objeto “Estado”.

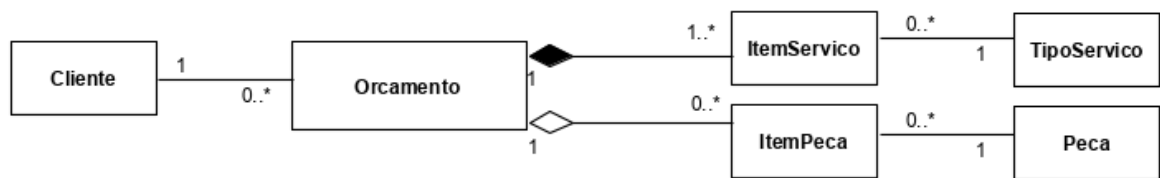


Fonte: elaborada pela autora.

E um tipo especial, uma variação da associação agregação, é a chamada Composição. Na associação desse tipo estabelece-se um vínculo mais forte entre o “objeto-todo” e os “objetos-partes”, demonstrando que os objetos-partes integram um único “objeto-todo”. Assim, os “objetos-partes” não podem viver quando o “objeto-todo” for destruído. Utiliza-se um losango sólido para indicar a composição, conforme representa o exemplo a seguir.

No exemplo da Figura 4.11, um objeto “Orcamento” é composto de no mínimo um objeto “ItemServico” ou de muitos, mas obrigatoriamente todo orçamento deve ter um item de serviço, senão não é um orçamento. E, no caso da composição, o objeto que representa o todo “Orcamento” deve ser criado, e os objetos-partes “ItemServico” devem ser criados na sequência, não podendo ser criados em outro momento de execução do sistema. Essa característica diferencia a Composição da Agregação.

Figura 4.11 | Relacionamento tipo composição



Fonte: elaborada pela autora.

DOCUMENTAÇÃO DE CADA CASO DE USO

Na sequência, agora que já temos a primeira versão do Diagrama de Classes, fica mais fácil elaborar a documentação de cada caso de uso. Não existe um formato específico de documentação para os casos de uso definido pela UML. O formato de documentação de um caso de uso é flexível, permitindo que se documente o caso de uso da forma que se considerar melhor, até mesmo com o uso de pseudocódigo ou de código de uma linguagem de programação. Outra opção é utilizar a prototipação da interface gráfica para facilitar a compreensão da execução do caso de uso.

O formato sugerido para documentação dos casos de uso é o formato numerado de descrição dos cenários – principal e alternativos. O cenário principal relata a descrição de uma tarefa que represente o mundo perfeito, sem exceções, e a descrição dos cenários alternativos relata qualquer situação que represente uma exceção (condição) do cenário principal. A seguir é demonstrada a documentação do caso de uso “Manter Empresa”.

Documentação do Caso de Uso Manter Empresa:

Número: 04

Use Case: Manter Empresa

Descrição: Empresa informa os seus dados para realização do seu cadastro como empresa conveniada da locadora de veículos

Ator Principal: Empresa

Cenário Principal:

1. Empresa informa seus dados para realização do seu cadastro.
2. Sistema exibe o cadastro de empresas.
3. Empresa informa seus dados.
4. Sistema verifica se empresa não está cadastrada.
5. Sistema verifica que ramo de atividade que referencia a empresa está cadastrado.
6. Sistema recupera dados do ramo de atividade vinculado à empresa.
7. Sistema verifica que país que referencia a empresa está cadastrado.
8. Sistema recupera dados do país vinculado à empresa.

9. Empresa confirma o seu cadastro.
10. Sistema registra a empresa.
11. Sistema emite Msg, informando que a empresa foi cadastrada com sucesso.
12. Sistema encerra o caso de uso.

Cenário Alternativo 4:

4. Sistema verifica que existe empresa cadastrada.
 - 4.1 Sistema recupera dados da empresa cadastrada.
 - 4.2 Sistema permite alteração ou exclusão da empresa.
 - 4.3 Empresa escolhe a opção de alteração.
 - 4.4 Empresa informa dados a serem alterados.
 - 4.5 Empresa confirma alteração dos seus dados.
 - 4.6 Sistema atualiza dados da empresa.
 - 4.7 Sistema encerra o caso de uso.

Cenário Alternativo 4.3:

- 4.3. Empresa escolhe a opção “a empresa não está associada a outro objeto”.
 - 4.3.2. Sistema emite Msg, confirmando a exclusão da empresa.
 - 4.3.3. Empresa confirma a exclusão da empresa.
 - 4.3.4. Sistema exclui a empresa.
 - 4.3.5. Sistema encerra o caso de uso.

Cenário Alternativo 4.3.1:

- 4.3.1 Sistema valida que a empresa está associada a outro objeto.
 - 4.3.1.1 Sistema emite Msg, informando que a empresa não pode ser excluída.
 - 4.3.1.2 Sistema encerra o caso de uso.

Cenário Alternativo 6:

6. Sistema verifica que o ramo de atividade não está cadastrado.
 - 6.1 Sistema permite a execução do use case “Manter Ramo de Atividade”.

Cenário Alternativo 8:

8. Sistema verifica que país não está cadastrado.

8.1 Sistema permite a execução do use case “Manter País”.

Cenário Alternativo 9:

9. Empresa não confirma o seu cadastro.

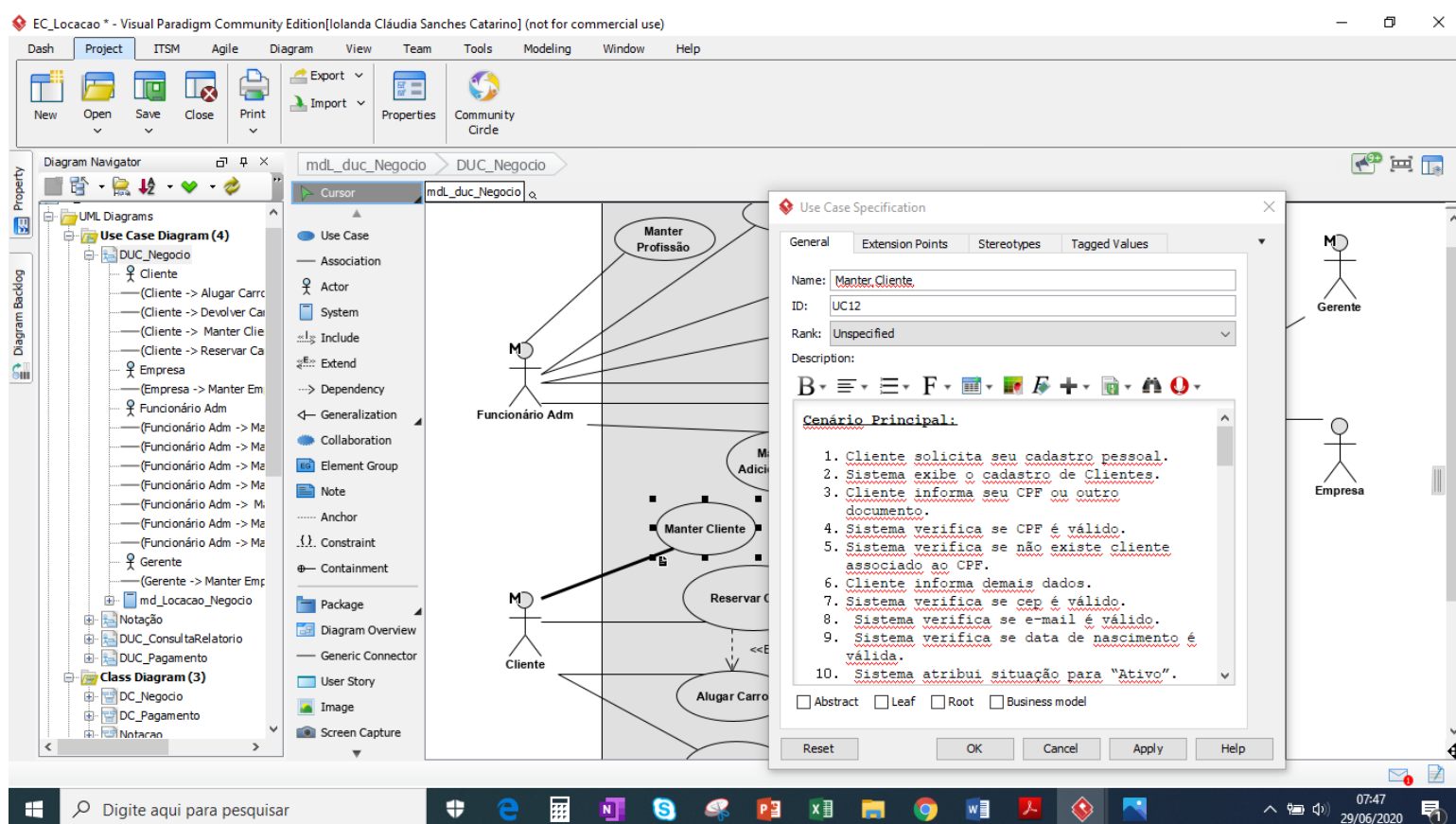
9.1 Empresa cancela o cadastro.

9.2 Sistema emite Msg, informado que o cadastro será cancelado.

9.3 Sistema encerra o caso de uso.

A documentação do caso de uso também pode ser feita diretamente na ferramenta CASE de modelagem, vinculada à representação do caso de uso, conforme mostra a Figura 4.12, que ilustra a documentação do caso de uso “Manter Cliente”. Na ferramenta CASE adotada – Visual Paradigm, deve-se selecionar o caso de uso, abrir suas propriedades na opção “*Use Case Specification*”, acessando o menu com o botão direito do mouse e no item “*Description*” descrever o relato do funcionamento do caso de uso.

Figura 4.12 | Documentação do Caso de Uso na Ferramenta CASE



Fonte: captura de tela do software Visual Paradigm Community Edition elaborada pela autora.

Posteriormente, deve-se continuar a modelagem comportamental e de interação do sistema, a partir do detalhamento do funcionamento dos casos de uso, adotando o Diagrama de Atividades, Diagrama de Sequência ou o Diagrama de

Visão Geral de Interação, bem como especificar os estados e suas transições dos objetos das classes que apresentam estados relevantes, como é o caso das classes “Reserva”, “Carro” e “Pessoa”.

REFERÊNCIAS

BEZERRA, E. **Princípios de análise e projeto de sistemas com UML**. 3. ed. Rio de Janeiro: Elsevier, 2014.

BOOCH, G.; RUMBAUGH, J.; JACOBSON, I. **UML: guia do usuário**. 2. ed. Rio de Janeiro: Campus, 2006.

GUERRINI, F. M. *et al.* **Modelagem da Organização: uma visão integrada**. Porto Alegre: Bookman, 2014.

KIRCHOFF, E. **BPMN em exemplos: aprenda como modelar processos de negócio**. Kirchoff, 2015.

MEDEIROS, E. S. **Desenvolvendo software com UML 2.0: definitivo**. São Paulo: Pearson, 2004.

PRESSMAN, R.; MAXIM, B. **Engenharia de software: uma abordagem profissional**. 8 ed. Porto Alegre: AMGH, 2016.

SOMMERVILLE, I. **Engenharia de software**. 10. ed. São Paulo: Pearson Education do Brasil, 2018.