

MODELAGEM DE INTERAÇÕES – DIAGRAMA DE SEQUÊNCIA

Iolanda Cláudia Sanches Catarino

0

Ver anotações

DIAGRAMA DE SEQUÊNCIA

O diagrama de sequência representa a ordem temporal em que as mensagens são trocadas entre os objetos envolvidos na execução de um processo.



Fonte: Shutterstock.

Deseja ouvir este material?

Áudio disponível no material digital.

PRATICAR PARA APRENDER

Caro aluno, seja bem-vindo à seção sobre modelagem de interações com o diagrama de sequência!

Nesta seção vamos continuar a modelagem dinâmica do sistema para detalhar a comunicação entre os objetos do sistema a partir do diagrama de sequência, um dos diagramas de interação da UML. É preciso lembrar que os diagramas de interação representam um subgrupo dos diagramas comportamentais e visam mostrar uma interação, formada por um conjunto de objetos e seus relacionamentos, detalhando as mensagens que são enviadas entre eles. Em geral,

a modelagem de um sistema tem vários diagramas de interação, e esse conjunto, com todos os diagramas de interação especificados, constitui o seu modelo de interações.

O diagrama de sequência tem o objetivo de representar a ordem temporal em que as mensagens são trocadas entre os objetos envolvidos na execução de um processo, que foi especificado como um caso de uso.

Assim, você vai conhecer e compreender os conceitos, os componentes, a notação gráfica de cada elemento do diagrama e as orientações para proceder com a construção do diagrama de sequência.

Na especificação dos requisitos funcionais do sistema, cada um deles foi representado no diagrama de casos de uso como um caso de uso e, para documentá-lo com a descrição de seu funcionamento interno, recomenda-se elaborar a descrição do cenário do caso de uso no formato de roteiro detalhado, com o relato do seu cenário principal e eventuais cenários alternativos, ou no formato simplificado. Quem define o melhor formato de documentar um caso de uso é o analista de sistemas, com sua equipe de desenvolvedores, mediante a metodologia definida pela empresa de desenvolvimento.

E será que só a documentação de cada caso de uso, no formato de roteiro, é suficiente para o entendimento de seu funcionamento interno?

Não! Não é suficiente. Além do relato de funcionamento de cada caso de uso, independentemente do formato adotado, é importante especificar de que forma os objetos do sistema interagem no funcionamento do caso de uso e quais informações devem ser enviadas em uma mensagem de um objeto para o outro e, ainda, em que ordem!

Então, a partir desta seção, você compreenderá como os diagramas de interação consolidam o entendimento sobre os aspectos dinâmicos do sistema e avançaremos na modelagem comportamental dele para a agência de estágios e intercâmbios, elaborando o diagrama de sequência para o caso de uso que representa a funcionalidade de realização de entrevista de estágio.

Vamos avançar com a modelagem comportamental do sistema de hotelaria, módulo Recepção, descrita na Seção 3.1. A partir da descrição do contexto do módulo Recepção que concentra as rotinas da recepção do hotel, incluindo o controle de reservas, vamos priorizar a modelagem da interação que corresponde ao processo de reserva realizada por um hóspede. Assim, o relato de como funciona essa atividade de reserva, de responsabilidade da recepção do hotel, é descrito a seguir.

DESCRIÇÃO DO PROCESSO DE REALIZAÇÃO DE RESERVAS

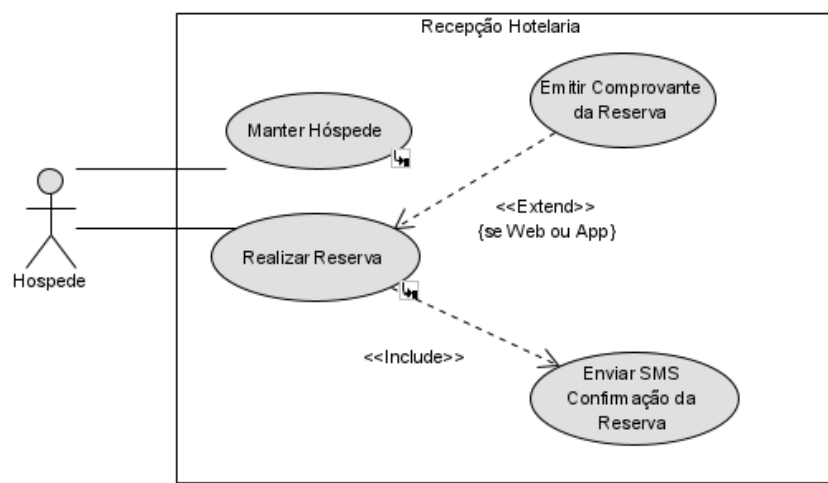
Um hóspede, para se hospedar no hotel, tem que ter um cadastro com seus dados pessoais efetuado previamente e tem que ter uma reserva prévia realizada por ele (por telefone, por aplicativo ou pela web) ou pela empresa que o hóspede tem

vínculo. Sendo a empresa responsável pela estada do hóspede, ela também é a responsável em custear integralmente as despesas referentes as diárias e as despesas referentes as refeições (ela pode custear ou não).

Uma reserva é mantida por: nome da(s) pessoa(s) que irá(ão) se hospedar em um apartamento, data que efetuou a reserva, data de entrada, hora prevista de entrada, data de saída, quantidade de adultos, quantidade de crianças, idade das crianças e nome da empresa que trabalha (se for a empresa responsável pela estada do hóspede). Um hóspede ou uma empresa pode realizar várias reservas.

A Figura 3.7 ilustra um recorte do Diagrama de Casos de Uso especificado para a atividade de Análise, ilustrando parte dos casos de uso identificados para atenderem as funcionalidades descritas no módulo de Recepção do sistema de hotelaria, correspondente ao caso de uso “Realizar Reserva”.

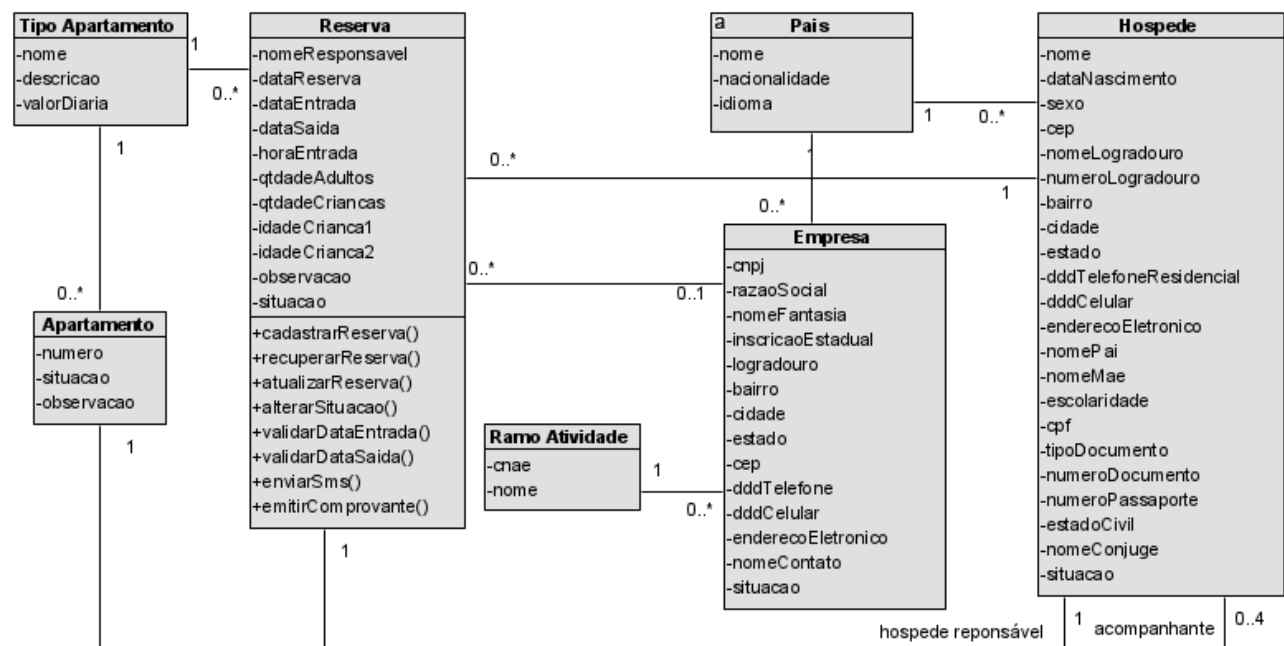
Figura 3.7 | Recorte do diagrama de casos de uso



Fonte: elaborada pela autora.

A próxima Figura apresenta um recorte do Diagrama de Classes, demonstrando as classes que foram definidas para a manipulação dos objetos “Reserva”.

Figura 3.8 | Recorte do diagrama de classes



Fonte: elaborada pela autora.

Elabore o diagrama de sequência correspondente ao caso de uso “Realizar Reserva”, representando o cenário principal e considerando as classes de objetos definidas na Figura 3.8.

Bom trabalho!

Boa leitura e bons estudos!

CONCEITO-CHAVE

Caro aluno, nesta seção avançaremos os estudos sobre a modelagem dinâmica do sistema!

Usualmente, a modelagem comportamental do sistema inicia-se com o modelo de casos de uso, o qual descreve os requisitos funcionais do sistema e os atores que interagem com o sistema, respondendo, assim, o que o sistema deve fazer e para quem. No entanto, o modelo de casos de uso não expõe o comportamento interno das funcionalidades especificadas. Para isso, a UML contempla os diagramas de interação, que representam um subgrupo dos diagramas comportamentais.

DIAGRAMA DE INTERAÇÃO

Os diagramas de interação mostram como os objetos do sistema agem internamente para apoiarem a realização das funcionalidades representadas pelos casos de uso, consolidando, assim, o entendimento dos aspectos dinâmicos do sistema. Os diagramas de interação da UML são: diagrama de sequência, diagrama de comunicação, diagrama de visão geral de interação e o diagrama de tempo.

Pense em como você poderá visualizar um sistema em execução e compreender a relação entre todos os objetos que participam da realização de cada funcionalidade. Uma melhor forma de representar esse funcionamento é construindo roteiros de cenários, descrevendo passo a passo a interação entre os objetos que participam da execução dos casos de uso e as mensagens que são trocadas entre eles. Outra opção é desenhar o protótipo correspondente à interface dos casos de uso para completar a compreensão dos cenários. Além dessas orientações, para se familiarizar com o funcionamento do sistema, é necessário completar e validar a modelagem de cada caso de uso com os diagramas de interação.

Segundo Guedes (2018), a interação entre objetos para dar suporte à funcionalidade de um caso de uso denomina-se realização de um caso de uso, o qual descreve o comportamento de um ponto de vista interno ao sistema, sendo que a realização de um caso de uso é representada por diagramas de interação.

Uma das formas mais utilizadas de se especificar a interação entre os objetos é a ênfase à ordenação temporal das mensagens, representando a sequência lógica da troca de mensagens formada por um conjunto de objetos e seus relacionamentos a partir da adoção do diagrama de sequência, o qual é uma forma de representar a ordem temporal em que as mensagens são trocadas entre os objetos envolvidos na execução de um processo.

DIAGRAMA DE SEQUÊNCIA

Conforme Guedes (2018), o diagrama de sequência descreve a ordem temporal em que as mensagens são trocadas entre os objetos envolvidos na execução de um processo que representa um caso de uso, bem como no ator responsável pela interação com os objetos.

De acordo com Booch, Jacobson e Rumbaugh (2006, p. 243), o diagrama de sequência é “um diagrama de interação que dá ênfase à ordenação temporal de mensagens. Graficamente, um diagrama de sequência é uma tabela que mostra objetos distribuídos no eixo X e mensagens, em ordem crescente no tempo, no eixo Y”.

O diagrama de sequência baseia-se no diagrama de casos de uso, em que é elaborado um diagrama de sequência para cada caso de uso, que se apoia no diagrama de classes para determinar os objetos das classes que realizam o caso de uso, com a indicação das mensagens trocadas entre os objetos, que são, na maioria das vezes, as operações das classes.

REFLITA


Os diagramas de interação representam um subgrupo dos diagramas comportamentais, enfatizando a modelagem dinâmica do sistema para demonstrarem o funcionamento interno dos casos de uso. Dessa forma, você conseguiu compreender como os diagramas comportamentais se complementam e por que devem ser consistentes com o diagrama estrutural – diagrama de classes?


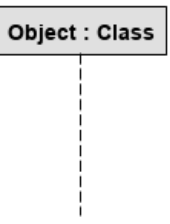


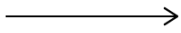
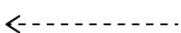
Agora vamos conhecer e compreender os elementos que constituem o diagrama de sequência.


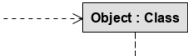
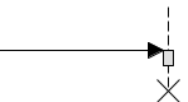
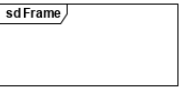

ELEMENTOS DO DIAGRAMA DE SEQUÊNCIA

Um diagrama de sequência é construído e representado pelos seguintes elementos: ator, mensagens, objetos, linha de vida e foco de controle. A notação gráfica dos principais elementos que podem compor o diagrama de sequência e uma breve descrição de cada elemento é apresentada no Quadro 3.3 a seguir.

Quadro 3.3 | Elementos do diagrama de sequência

Notação	Elemento
	Linha de vida: representa a existência de um elemento (objeto ou ator) participante da realização do caso de uso em um período de tempo. É representada por uma linha vertical tracejada abaixo do elemento, chamada de cauda.

Notação	Elemento
	Ator: representa os mesmos atores já criados no diagrama de casos de uso; são apoiados por uma linha de vida e enviam mensagens para os objetos como uma forma de interação para solicitarem a execução de uma operação ou simplesmente o envio de informações. No diagrama sempre representa o ator primário responsável por enviar a mensagem inicial, que começa a interação entre os objetos.
	Objeto: representa os objetos que participam da realização do caso de uso; são também apoiados por uma linha de vida, que juntamente com os atores, formam um cabeçalho para o diagrama. Um objeto pode existir desde o início da interação ou ser criado ao longo da dela. Um objeto é representado por um retângulo com um nome único, conforme o padrão da notação de objeto.
	Foco de controle: representa o período de tempo durante o qual um elemento executa uma ação, diretamente ou não. É representado por um retângulo estreito na vertical sobre a linha de vida, podendo aparecer diversas vezes ao longo dela.
	Mensagem síncrona: a mensagem é síncrona quando o emissor aguarda o retorno para continuar com a interação. São as mensagens comumente utilizadas no diagrama de sequência. É representada por uma linha horizontal com uma seta sólida na extremidade.
	Mensagem assíncrona: a mensagem é assíncrona quando o emissor continua enviando mensagens sem aguardar o retorno, com isso o elemento receptor da mensagem assíncrona não precisa atendê-la imediatamente. É representada por uma linha horizontal com uma seta aberta.
	Mensagem de retorno: é uma mensagem que um objeto envia ao outro em resposta à mensagem recebida após a execução de uma ação. As mensagens de retorno são representadas por uma linha tracejada com uma seta na extremidade, apontando para o elemento que recebe a resposta.

Notação	Elemento
	Mensagem reflexiva ou automensagem: é uma mensagem indicativa de que o objeto remetente da mensagem é também o receptor. A mensagem reflexiva é representada por uma seta que sai do objeto e retorna para ele mesmo.
	Mensagem construtora: indica o momento em que o objeto passa a existir no sistema, ou seja, o objeto é instanciado ao longo do processo por uma mensagem enviada. A mensagem construtora é representada por uma linha tracejada com seta na extremidade, apontando para o centro do objeto criado. O retângulo que representa o objeto é posicionado mais abaixo no diagrama.
	Mensagem destrutora: indica a destruição do objeto no decorrer da interação, o qual não se mostra mais necessário no processo. É representado pelo símbolo X na parte inferior da linha de vida do objeto que está sendo destruído.
	Quadro de interação: representa uma interação independente, possibilitando isolar um diagrama de sequência com um contexto específico, formando uma fronteira para que possa ser integrado aos demais diagramas de sequência. O quadro é representado graficamente por um retângulo com um rótulo no canto superior esquerdo, que identifica o tipo do quadro.
	Nota ou comentário: serve para escrever observações aos elementos que compõe o diagrama de sequência, contendo informações úteis para os desenvolvedores, porém não expressa força semântica específica aos elementos do diagrama.

Fonte: elaborado pela autora.

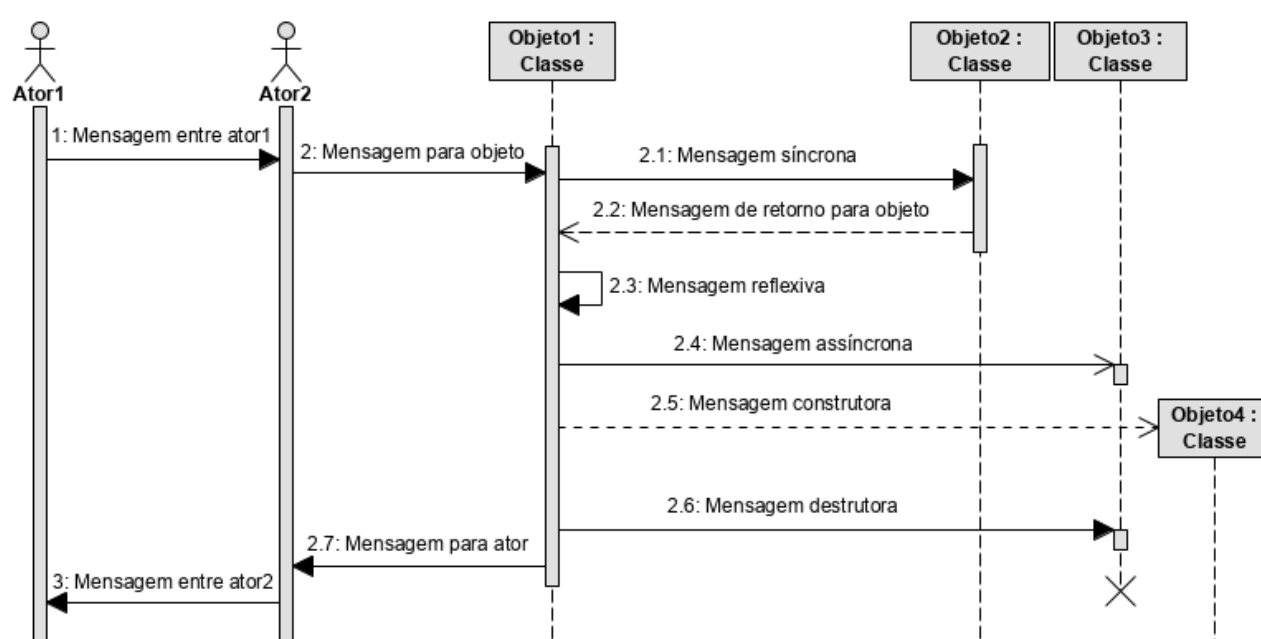
TROCAS DE MENSAGENS

Nos diagramas de interação, como no diagrama de sequência, as mensagens representam a solicitação que um elemento envia para o outro com o objetivo de executar uma ação, demonstrando a ocorrência de eventos. O objeto que envia a mensagem denomina-se objeto emissor ou remetente, e o objeto que recebe a mensagem denomina-se objeto receptor. As possíveis trocas de mensagens que podem acontecer entre os elementos do diagrama são:

- **Ator e ator:** indica uma comunicação entre os atores.
- **Ator e objeto:** geralmente o ator provoca um evento, enviando uma mensagem que dispara uma operação, contudo o ator pode simplesmente transmitir uma informação sem disparar uma operação.
- **Objeto e objeto:** indica o envio de uma mensagem, disparando uma operação. Quando um objeto envia uma mensagem para si mesmo, essa operação é denominada mensagem reflexiva.
- **Objeto e ator:** indica o retorno de uma mensagem com o seu resultado.

A Figura 3.9 ilustra a interação entre os elementos do diagrama de sequência com os tipos de mensagens possíveis.

Figura 3.9 | Elementos e tipos de mensagens do diagrama de sequência



Fonte: elaborada pela autora.

ASSIMILE

Nos diagramas de interação, as mensagens representam a solicitação que um elemento envia para o outro com o objetivo de executar uma ação, demonstrando a ocorrência de eventos. As possíveis trocas de mensagens que podem acontecer entre os elementos do diagrama de sequência são: ator e ator; ator e objeto; objeto e objeto; objeto e ator. Contudo, a troca de mensagens entre os elementos ator e objeto ou entre objeto e ator sempre deve referenciar um objeto que representa a interface do caso de uso.

Além da representação da troca de mensagens entre os elementos do diagrama de sequência, é importante indicar o conteúdo da mensagem enviada pelo remetente, o qual especifica informações a serem passadas para o elemento receptor denominado rótulo da mensagem. Os rótulos podem indicar:

- Informações que descrevem uma solicitação ou resposta entre os elementos ator e objeto, que representa uma interface correspondente à funcionalidade do caso de uso do diagrama, como “Solicitar cadastro do candidato”; “Solicitar inscrição para vaga” ou “Consultar vaga”.
- Uma mensagem, que é uma operação de uma classe, é representada com o nome da operação mais um par de parênteses no final “nomeOperacao()”, por exemplo, `inserirCandidato()` ou `obterVaga()`; tem exatamente a mesma nomenclatura da operação listada no diagrama de classes.
- Uma mensagem, que é uma operação de uma classe, assim como a do item anterior, e que precisa representar a necessidade de um retorno ou não, mostra-se como “nomeOperacao():void”, nesse caso, o “void” é a palavra reservada para representar que não haverá retorno; ou como “nomeOperacao():Boolean”, nesse caso, o tipo de dados “Boolean” representa que, ao executar essa operação, terá como retorno o valor “Boolean”.
- Uma mensagem, que é também uma operação de uma classe, assim como a dos itens anteriores, e que precisa representar uma mensagem com passagem de parâmetro, utiliza-se da nomenclatura “nomeOperacao(atributo: tipo de dado)”, por exemplo `recuperarInscricaoVaga(cpf:Long)`.

Juntamente com o rótulo da mensagem, também pode-se indicar uma condição de guarda que estabelece uma regra ou condições para que uma mensagem possa ser disparada. As condições de guarda são descritas entre colchetes na mensagem.

ESTERÉOTIPOS DAS CLASSES

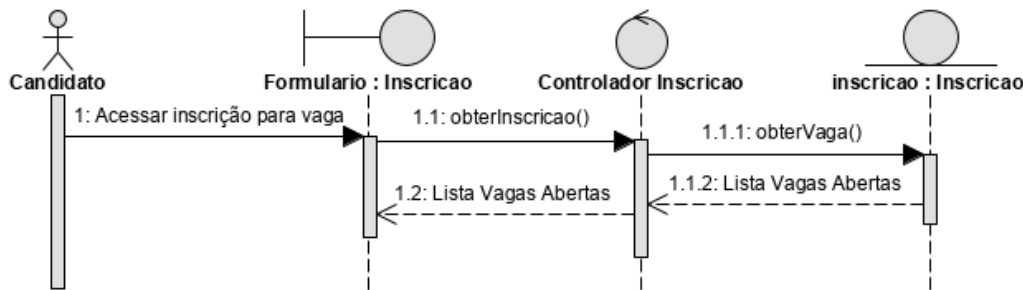
Na representação dos objetos do diagrama de sequência, pode-se adotar os estereótipos do diagrama de classes para melhor compreensão das interações entre os objetos, tais como: <<boundary>>, <<control>> e <<entity>>; sendo:

- **Estereótipo do tipo <<boundary>>:** denominado de classe de fronteira, é aquele que representa a interface do sistema, indicando a comunicação entre o ator primário e os demais objetos das classes que participam da interação.

- **Estereótipo do tipo <<control>>:** denominado de classe de controle, o qual serve de intermediário entre as classes definidas como <<boundary>> e <<entity>> para tratar as regras de negócio e o fluxo da aplicação.
- **Estereótipo do tipo <<entity>>:** denominado de classe de entidade, é aquele que mostra que as classes do sistema também são entidades, seja persistente ou transiente durante a execução do sistema.

A Figura 3.10 ilustra uma parte de um diagrama de sequência para exemplificar a notação gráfica dos estereótipos das classes. O elemento “Formulario:Inscricao” representa uma classe de fronteira que faz a interação com o ator “Candidato” e com o elemento “ControladorInscricao”, representando uma classe de controle responsável pela mediação do tratamento das regras de negócio com o elemento “inscrição:Inscricao”, que representa uma classe de entidade persistente na interação.

Figura 3.10 | Diagrama de sequência com estereótipos das classes



Fonte: elaborada pela autora.

FRAGMENTOS

Na representação do diagrama de sequência, também é possível utilizar **fragmentos de interação** e **fragmentos combinados** que possibilitam o alinhamento de interações, sendo que cada fragmento representa uma interação independente, formando uma fronteira entre os elementos do diagrama. Nesse sentido, é recomendável representar os fragmentos apenas se de fato contribuírem para a compreensão da interação.

Um **fragmento de interação**, também denominado de ocorrência de interação, representa a ocorrência de um outro diagrama de interação da UML, sendo que, na representação do diagrama de sequência, indica outro diagrama de sequência.

Na notação gráfica do quadro de interação, identifica-se o rótulo com a expressão “**ref**” e, no centro do quadro, descreve-se o nome do diagrama de sequência referenciado. Nos fragmentos de interação, pode-se utilizar do elemento chamado “portões (*gates*)” para relacionar uma mensagem fora de um fragmento de interação com uma mensagem dentro do fragmento de interação, estabelecendo uma interface entre fragmentos, ou seja, um ponto de conexão.

Um **fragmento combinado** é utilizado para definir o fluxo de controle da interação, correspondendo a uma sequência de mensagens encapsuladas em um fragmento, compondo um procedimento que pode ser reutilizado em demais

diagramas de sequência. Os fragmentos combinados são representados pelo elemento quadro de interação com uma identificação no rótulo, que descreve o tipo de operador de interação, que pode ser (BEZERRA, 2014):

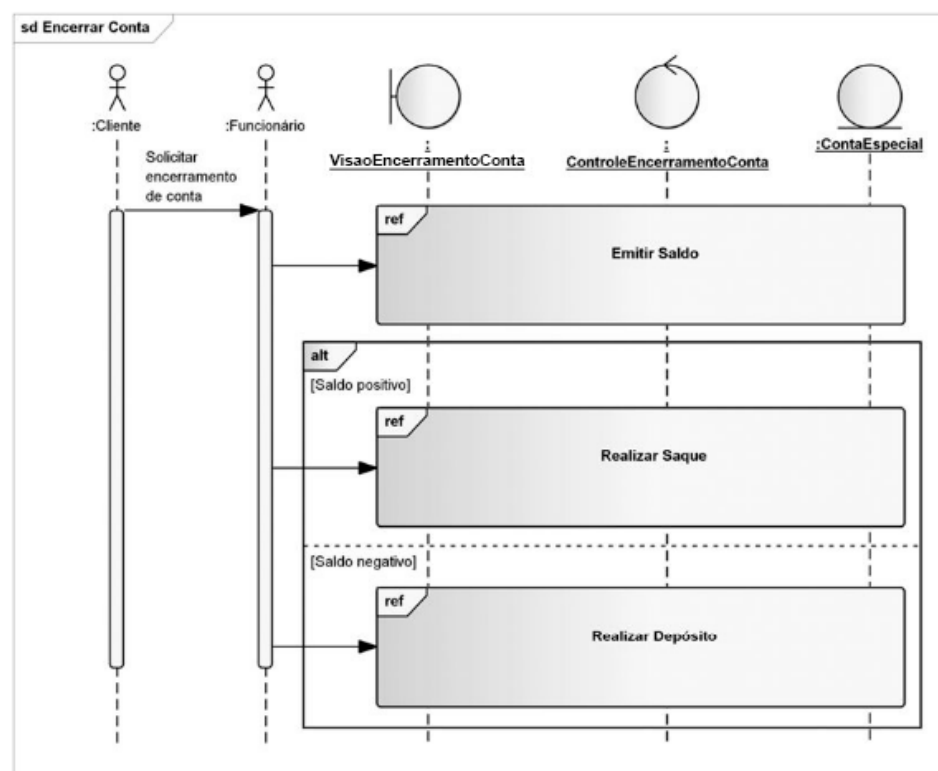
- ***alt*: abreviatura de Alternative (Alternativa)**: modela a construção procedimental do tipo se-então-senão, ou seja, uma escolha entre duas ou mais ações, sendo que o quadro de interação é dividido em partes por uma linha tracejada, representando cada operador com uma condição de guarda (texto entre colchetes que estabelece uma condição ou uma regra). Apenas um operador do quadro é executado.
- ***opt*: abreviatura de Option (Opção)**: modela a construção procedimental do tipo se...então, sendo que o operador *opt* representa uma escolha de comportamento que será ou não executado a partir de uma condição de guarda.
- ***loop*: abreviatura de Looping (Laço)**: representa que uma interação deve ser realizada zero ou mais vezes conforme indicação da expressão com os limites mínimo e máximo, definindo a quantidade de repetições.

ASSIMILE

Os fragmentos do diagrama de sequência podem ser de dois tipos: fragmentos de interação e fragmentos combinados. Um fragmento de interação representa a ocorrência de um outro diagrama de sequência e o fragmento combinado é utilizado para definir o fluxo de controle da interação, correspondendo a uma sequência de mensagens agrupadas em um fragmento, que compõe um procedimento.

A Figura 3.11 ilustra um exemplo de diagrama de sequência correspondente ao caso de uso “Encerrar Conta”, que contém o fragmento de interação “Emitir Saldo” e o fragmento combinado do tipo alternativa, contendo dois fragmentos de interação – “Realizar Saque” e “Realizar Depósito”.

Figura 3.11 | Exemplo de diagrama de sequência com fragmentos



Fonte: Guedes (2018, p. 208).

ELABORANDO O DIAGRAMA DE SEQUÊNCIA

Para elaborar um diagrama de interação, é necessário um diagrama de casos de uso com a descrição do roteiro dos cenários dos casos de uso e também do diagrama de classes. Em geral, durante a construção de um diagrama de interação, é comum identificar novas classes, atributos e principalmente a definição das operações, bem como uma descrição refinada dos cenários dos casos de uso. Dessa forma, os diagramas de interação da UML reforçam o apoio ao desenvolvimento incremental de sistemas orientados a objetos a partir de modelos que podem evoluir com a inclusão de novos detalhes.

A Figura 3.12 representa a relação entre o diagrama de casos, com sua documentação, e um projeto do protótipo da interface gráfica correspondente ao caso de uso descrito, além do diagrama de classes, cuja finalidade é identificar a classe ou classes de objetos que participam da realização de cada caso de uso para, assim, construir o diagrama de interação pretendido, especificando a troca de mensagens entre os objetos.

Figura 3.12 | Relação entre os modelos comportamental e estrutural


```

classDiagram
    class Patron {
        name : string
        address : string
        libraryCardID : string
        phone : string
        email : string
        lastBorrowDate : date
    }
    class Book {
        title : string
        author : string
        year : int
        isbn : string
        genre : string
        status : string
    }
    class Bookshelf {
        title : string
        author : string
        year : int
        isbn : string
        genre : string
        status : string
    }
    class BookCategory {
        title : string
        author : string
        year : int
        isbn : string
        genre : string
        status : string
    }
    Patron "1" -- "0..*" Book : borrow
    Patron "1" -- "0..*" Bookshelf : borrow
    Patron "1" -- "0..*" BookCategory : borrow
    Book "1" -- "0..*" Bookshelf : borrow
    Book "1" -- "0..*" BookCategory : borrow
    Bookshelf "1" -- "0..*" BookCategory : borrow
    
```

The diagram illustrates the relationships between four classes: Patron, Book, Bookshelf, and BookCategory. Each class has a set of attributes. Patron has attributes for name, address, libraryCardID, phone, email, and lastBorrowDate. Book, Bookshelf, and BookCategory all share the same attributes: title, author, year, isbn, genre, and status. The relationships are as follows: Patron has a 1-to-0..* relationship with Book, Bookshelf, and BookCategory, all labeled 'borrow'. Book has a 1-to-0..* relationship with Bookshelf and BookCategory, both labeled 'borrow'. Bookshelf has a 1-to-0..* relationship with BookCategory, labeled 'borrow'. There are also yellow sticky notes on the right side of the diagram: 'Library Patron' (name: John, address: 123 Main St, phone: 555-1234, email: john.doe@example.com, lastBorrowDate: 2023-10-27), 'Book' (title: The Great Gatsby, author: F. Scott Fitzgerald, year: 1925, isbn: 978-0-7432-7356-5, genre: Classic, status: Available), 'Bookshelf' (title: The Great Gatsby, author: F. Scott Fitzgerald, year: 1925, isbn: 978-0-7432-7356-5, genre: Classic, status: Available), and 'BookCategory' (title: The Great Gatsby, author: F. Scott Fitzgerald, year: 1925, isbn: 978-0-7432-7356-5, genre: Classic, status: Available).

Número: 01

Use Case: Manter Candidato

Descrição: Candidato informa seus dados para realizar seu cadastro pessoal.

Ator: Cliente

Curso Normal:

1. Candidato solicita seu cadastro pessoal.
2. Sistema exibe o cadastro de Candidatos.
3. Candidato informa seu CPF.
4. Sistema verifica se CPF é válido.
5. Sistema verifica se não existe Candidato associado ao CPF.
6. Candidato informa demais dados.
7. Sistema verifica se cep é válido.
8. Sistema verifica se e-mail é válido.
9. Sistema verifica se data de nascimento é válida.
10. Sistema atribui situação para "Ativo".
11. Sistema atribui data de cadastro.
12. Candidato confirma o cadastro.
13. Sistema verifica se atributos requeridos foram preenchidos.
14. Sistema registra candidato.
15. Sistema emite Msg 01, informando que o Candidato foi cadastrado com sucesso.
16. Sistema encerra o use case.

Curso Alternativo 1:

4. Sistema verifica que CPF não é válido.
- 4.1 Sistema emite Msg 01, informando que o CPF não é válido.
- 4.2 Sistema permite informar novo CPF.

```

sequenceDiagram
    participant Ent as Entrevistador
    participant Form as Form_Realizar Entrevista
    participant Func as Funcionario: Funcionario
    participant Candidato as candidato: Candidato
    participant Vaga as vaga: Vaga
    participant Entrevista as entrevista: Entrevista

    Ent->>Form: 1.1: Selecionar entrevista
    Form->>Ent: 2: Retornar dados da entrevista
    Form->>Func: 2.1: getFuncionariosPendentes()
    Form->>Func: 2.2: obterFuncionarioSelecionado()
    Form->>Func: 2.3: dados do funcionario
    Form->>Candidato: 2.4: selecionarCandidato()
    Form->>Candidato: 2.5: dados do candidato
    Form->>Vaga: 2.6: selecionarVaga()
    Form->>Vaga: 2.7: dados da vaga
    Form->>Entrevista: 2.8: selecionarFuncionario()
    Form->>Entrevista: 2.9: dados do entrevistador
    Form->>Entrevista: 2.10: validarDadosEntrevista()
    Form->>Entrevista: 2.11: salvarNovoEntrevista()
    Form->>Entrevista: 2.12: obterFuncionarioSelecionado()
    Form->>Entrevista: 3: Confirmar candidato
    Form->>Entrevista: 3.1: Seleção final (entrevista candidato realizado)
    Form->>Entrevista: 3.2: resultado
  
```

Fonte: elaborada pela autora.

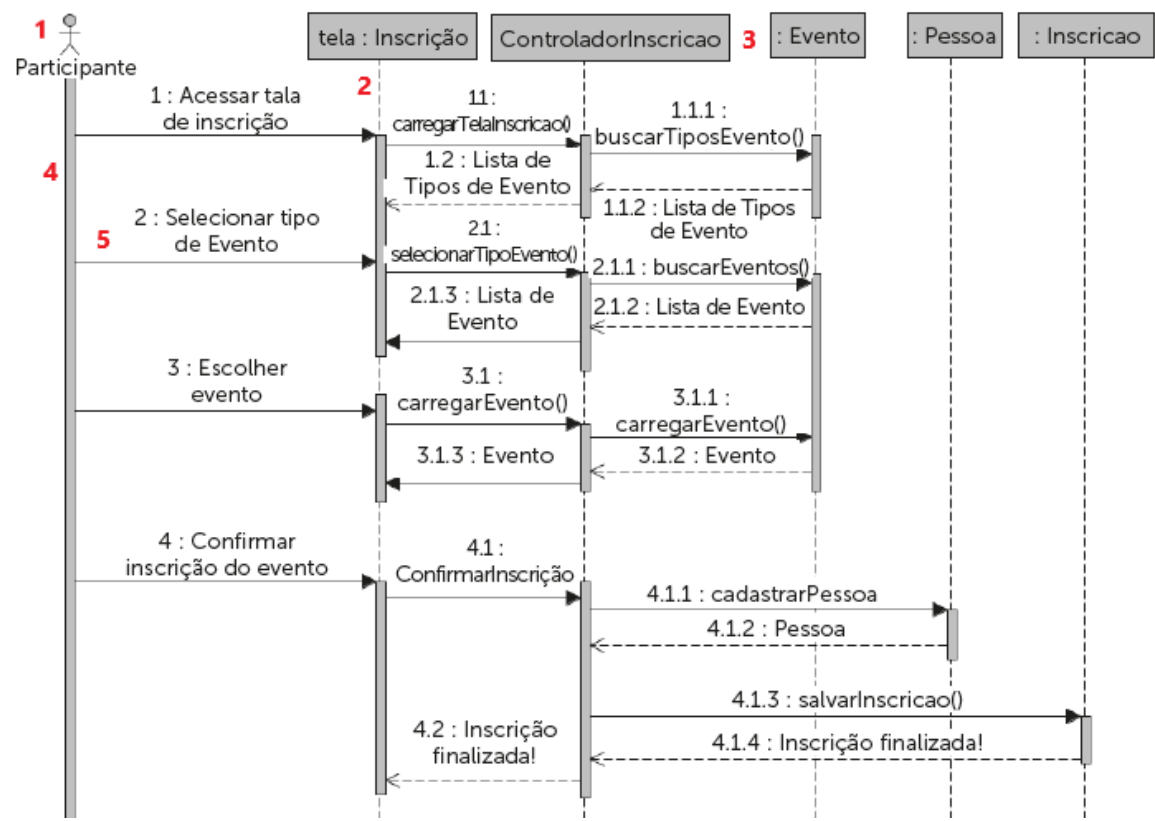
Os seguintes passos são recomendados para construir o diagrama de sequência correspondente a cada caso de uso:

1. Utilizar os roteiros dos cenários da documentação do caso de uso para identificar os objetos que participam da realização do caso de uso.
2. Analisar o diagrama de classes para definir as classes correspondentes aos objetos identificados no item 1.
3. Rever os roteiros dos cenários da documentação do caso de uso para identificar as mensagens que são trocadas entre os objetos das classes definidas.
4. Analisar as operações das classes definidas e ser consistentes com as mensagens identificadas no item 3.
5. Elaborar o diagrama de sequência, compondo o cabeçalho do diagrama com o ator primário, que interage com o caso de uso e com os objetos das classes definidas no item 2, e, se necessário, particionar o diagrama em fragmentos de interação e/ou fragmentos combinados.
6. Verificar a consistência do diagrama de sequência com o diagrama de casos de uso e, a partir do conhecimento adquirido com a construção do diagrama de interação, se necessário, aperfeiçoar a descrição dos cenários.
7. Verificar a consistência do diagrama de sequência com o diagrama de classes e, a partir do conhecimento adquirido com a construção do diagrama de interação, se necessário, aperfeiçoar as classes de objetos que integram o diagrama com novos atributos, operações e associações. Geralmente a ferramenta CASE, com a qual o diagrama foi modelado, atualiza automaticamente o diagrama de classes, adicionando as operações às classes dos objetos que foram complementadas no diagrama.

A Figura 3.13 ilustra um exemplo de diagrama de sequência correspondente ao cenário principal do caso de uso Realizar Inscrição do Evento com a indicação de seus elementos. O elemento 1 representa o ator “Participante”, já indicado no diagrama de casos de uso. O elemento 2 representa o elemento linha de vida, que acompanha cada objeto ou ator do diagrama. O elemento 3 representa o objeto “Evento” e, na sequência, os demais objetos “Pessoa” e “Inscricao”. O elemento 4 representa o foco de controle sobre a linha de vida do ator “Participante”. O elemento 5 representa uma mensagem enviada pelo ator participante e recebida pelo objeto “tela : Inscrição”, que dispara mensagens simples e mensagens que

invocam operações, sendo a mensagem identificada pela numeração 3.1 “carregarEvento()”, enviada pelo objeto “tela : Inscrição” para o objeto “ControladorInscricao”, é uma mensagem síncrona que dispara a operação “carregarEvento()”.

Figura 3.13 | Exemplo de diagrama de sequência

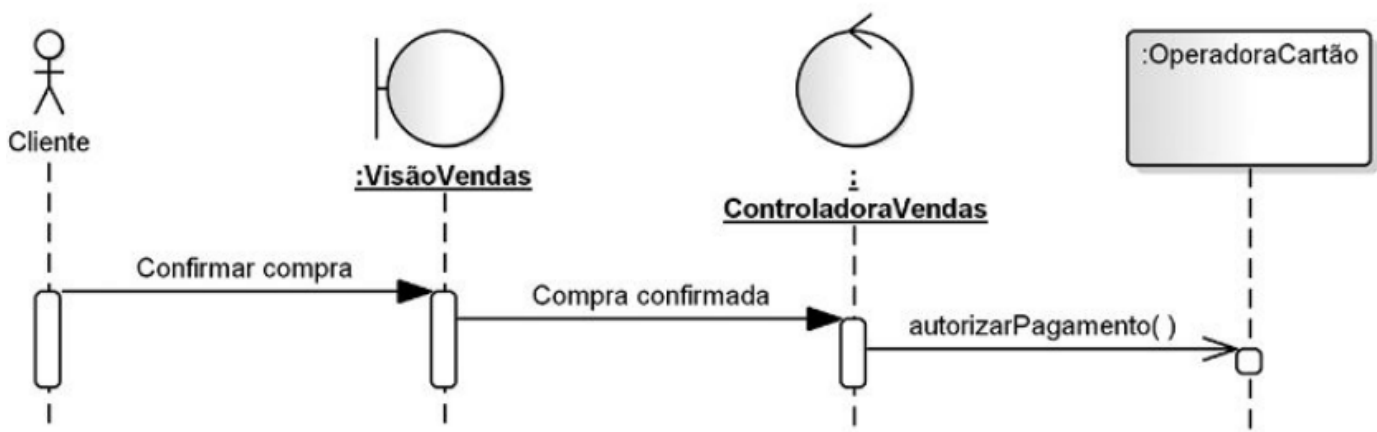


Fonte: elaborada pela autora.

EXEMPLIFICANDO

Nos diagramas de interação, a mensagem assíncrona ocorre quando o emissor continua enviando mensagens sem aguardar o retorno. Com isso, o elemento receptor da mensagem assíncrona não precisa atendê-la imediatamente. A Figura 3.14 ilustra um exemplo de mensagem assíncrona representada em uma parte de um diagrama de sequência, cujo objeto ControladoraVendas envia a mensagem assíncrona autorizarPagamento() para a classe OperadoraCartão e não aguarda um retorno dessa mensagem para continuar com o processo.

Figura 3.14 | Exemplo de mensagem assíncrona



Fonte: Guedes (2018).

Bezerra (2014) afirma que a modelagem dinâmica de um sistema não é algo simples e que, em sistemas complexos, há vários caminhos que podem ser adotados para definir a execução do sistema. Nesse contexto, a melhor alternativa é trabalhar iterativamente, ou seja, elaborar os modelos de casos de uso e de

classes inicialmente e, posteriormente, desenvolver os modelos de interações e de estados para só então retornar aos modelos iniciais e verificar a consistência entre os modelos.

REFERÊNCIAS

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

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

DRLJAČA, D.; LATINOVIĆ, B.; STARČEVIĆ, D. Modelling the Process of Is Auditing in the Public Administration Using Uml Diagrams. **Journal of Information Technology & Applications**, [S. l.], v. 7, n. 1, p. 32-41, 2017.

Disponível em: <https://bit.ly/357w9Pz>. Acesso em: 21 maio 2020.

FOWLER, M. **UML essencial**: um breve guia para a linguagem-padrão de modelagem de objetos. 3. ed. Porto Alegre: Bookman, 2005.

GUEDES, G. T. A. **UML**: uma abordagem prática. 3. ed. São Paulo: Novatec, 2018.