

MODELAGEM DE ATIVIDADES

Maurício Acconcia Dias

0

Ver anotações

DIAGRAMA DE ATIVIDADES

Permite que as atividades e os fluxos de controle do sistema sejam representados em um diagrama e, posteriormente, possam auxiliar no desenvolvimento do sistema.



Fonte: Shutterstock.

Deseja ouvir este material?

Áudio disponível no material digital.

PRATICAR PARA APRENDER

Seja bem-vindo à última seção desta unidade. Neste espaço, o diagrama UML abordado é o diagrama de atividades. A partir dele, pode-se representar o conjunto de atividades a serem executadas em um processo do sistema, assim como os fluxos de controles.

Como exemplo, imagine que você está em um terminal de um caixa eletrônico de um banco. Neste terminal, é necessário que exista um software capaz de gerenciar todas as operações possíveis em relação a saques, depósitos, transferência e gerenciamento da conta em geral. Pense sobre o que deve ser envolvido no processo: você, enquanto cliente, executará algumas ações; o computador do

terminal será responsável por interpretar essas ações e responder de acordo com o que for solicitado; por fim, um servidor do banco informará os resultados das operações solicitadas, se obtiveram sucesso ou não, e conferirá os dados finais da conta, para que possa salvar essas informações.

Imagine, agora, todas as atividades que você pode realizar nesta situação: iniciar o acesso, digitar a senha, escolher a operação desejada, confirmar sua identidade (geralmente, com mais de um método, por exemplo, senha numérica e biometria), realizar a operação provendo as informações necessárias para o sistema, pegar o comprovante da operação gerado após a conclusão e encerrar seu acesso ao sistema, levando seu cartão com você. Essas são as atividades realizadas por você, porém, enquanto cada operação é processada, um acesso ao servidor do banco é feito para consulta de saldo e outras informações da sua conta. Esse acesso é efetuado pelo programa, o qual é executado no terminal de autoatendimento.

As ações descritas são modeladas pelo diagrama de atividades, que cria um fluxograma com início e final definidos para cada sequência de ações.

O diagrama de atividades é importante porque permite que as atividades e os fluxos de controle do sistema sejam representados em um diagrama e, posteriormente, podem auxiliar no desenvolvimento do sistema.

Nesta seção, você aprenderá os principais conceitos do diagrama de atividades, seus fluxos de controle, componentes, partições de atividades e, principalmente, como elaborar um diagrama a partir de um problema.

Você foi contratado como desenvolvedor de software em uma empresa que pudesse utilizar seus conhecimentos de UML e programação. Ao se candidatar às vagas que encontrou, foi selecionado e admitido para fazer parte do time de desenvolvimento de um banco.

Ao estudar as principais atribuições de um desenvolvedor em uma equipe de um banco, descobriu que existem diversos tipos de sistemas de informação sendo executados diariamente em diferentes partes de uma agência bancária. Além disso, existem os softwares de gerência, os quais são executados em servidores e são responsáveis pela atualização de informações dos movimentos diários de cada agência.

Para o início de seu trabalho, é necessário que você passe pelas fases iniciais do desenvolvimento do sistema, para, então, estar habilitado a contribuir em outros projetos. Os sistemas mais simples dentre os existentes no banco são os que envolvem o caixa eletrônico. Nele existem diversas operações que podem ser realizadas, dentre elas, a mais simples é a operação de sacar dinheiro com o cartão de crédito.

Você foi selecionado para modelar o diagrama de atividades do saque de dinheiro com cartão de crédito. Imagine uma situação em que uma pessoa para no caixa eletrônico de posse de seu cartão para sacar dinheiro. Pense em todas as operações que ocorrem com o cliente, as consultas ao servidor e as respostas da máquina. Para guiar o desenvolvimento, alguns requisitos foram apresentados:

- O diagrama deve conter as *swimlanes* do cliente, da máquina ATM e do servidor do banco.
- A senha é digitada apenas uma vez para verificação.
- Após o saque, a máquina mostra o saldo na tela.
- Ao finalizar a operação, a máquina libera o cartão, e a atividade é encerrada.

Pronto para aprender como transformar a ideia de fluxo de ações em um diagrama? Bons estudos!

CONCEITO-CHAVE

DIAGRAMA DE ATIVIDADES

Dentre os diagramas da linguagem UML, o diagrama de atividades desempenha um papel fundamental por mostrar aspectos comportamentais na modelagem de sistemas. Basicamente, ele consiste em um gráfico na forma de fluxo que mostra as características dinâmicas do sistema, podendo ser visualizado o fluxo de ações que serão realizadas em uma determinada atividade relacionada ao sistema que será desenvolvido.

Esta perspectiva sobre o sistema é importante ser considerada, pois, muitas vezes, o produto de software desenvolvido pode apresentar melhorias no fluxo de ações das atividades necessárias em um sistema. Além disso, a sequência das atividades pode não ter sido implementada de forma otimizada e poderia ter melhor performance se realizada de outra forma, por exemplo, concorrentemente. Nesse

sentido, o diagrama de atividades, em conjunto com outros diagramas UML, torna-se uma alternativa capaz de evitar esse tipo de problema, podendo auxiliar no desenvolvimento e na melhoria da qualidade do produto de software.

Dentre os **principais objetivos** do diagrama de atividades, podemos ressaltar: torna mais claro o fluxo de controle em operações complexas de casos de uso do sistema; mostra, com relação à lógica de execução, como uma tarefa deve ser feita e pode decompor atividades maiores em subatividades, facilitando o entendimento do sistema e suas operações necessárias.

Algumas de suas **características** também são interessantes de se discutir antes de mostrarmos as partes do diagrama. Podemos entender o diagrama de atividades como casos especiais do diagrama de casos de uso. Uma vez que uma atividade representa um possível caso de uso do sistema de forma detalhada, os diagramas estão relacionados de certa forma. É possível elaborar diagramas de atividades com base nos casos de uso já descritos para o sistema.

Esse diagrama modela atividades que ocorrem de forma concorrente durante a execução do sistema, isto é, ações que ocorrem paralelamente em diferentes partes do sistema a partir do início da atividade. Esta visão completa do sistema auxilia no desenvolvimento e na procura por causas de erros, visto que apresenta todas as partes necessárias para a execução de cada atividade.

Assim, como podemos representar o sistema? O diagrama de atividades exibirá passo a passo as ações do sistema, considerando cada uma das partes que está processando alguma operação enquanto a atividade é realizada. Este diagrama é orientado a fluxos de controle, tendo uma similaridade com os fluxogramas de programas. Todavia, essa semelhança está apenas na aparência, devido a uma maior quantidade de informações, já que, em um diagrama de atividades, pode-se englobar representações concorrentes e sincronização de fluxos de controles. É importante destacar que o diagrama de atividades possibilita expandir a análise de um caso de uso.

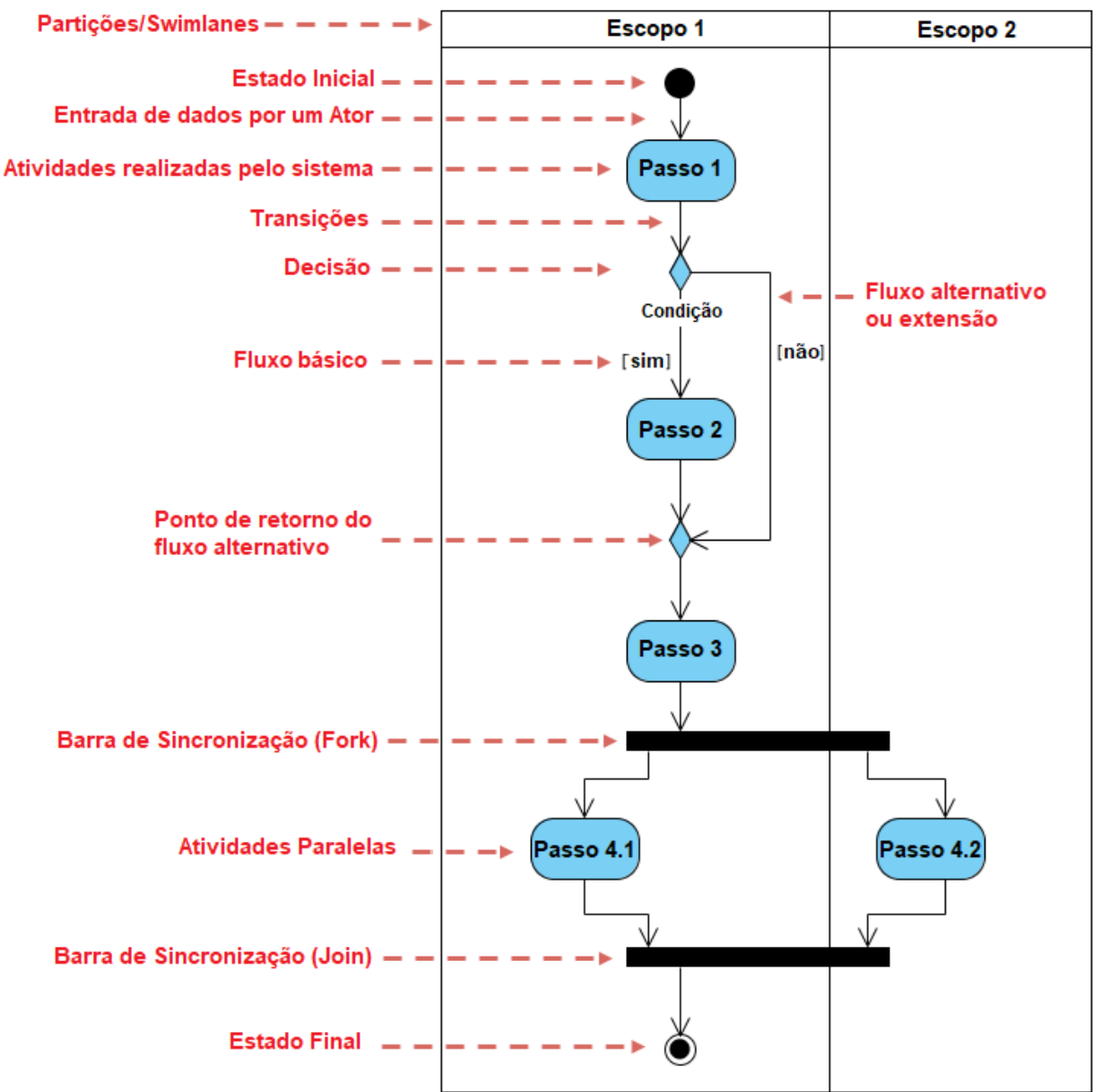
ELEMENTOS DO DIAGRAMA DE ATIVIDADES

Em geral, a criação dos diagramas de atividades é realizada do topo para o final e, normalmente, possui os seguintes elementos principais:

- Estados iniciais e finais.
- Atividades.
- Decisões.
- Transições.
- Barras de sincronização.
- *Swimlanes* (partições).

Todos os elementos listados serão abordados na sequência, trazendo alguns exemplos de diagramas. Antes desse detalhamento, a Figura 2.30 apresenta uma visão do diagrama com os elementos a serem descritos.

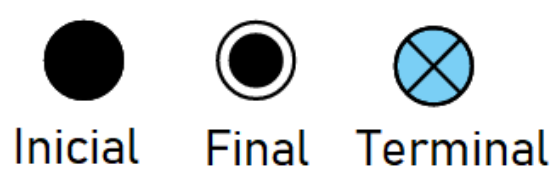
Figura 2.30 | Exemplo de um diagrama de atividades com exemplos de seus elementos



Fonte: adaptado de Visual Paradigm (2020).

Todo diagrama possui os estados iniciais e finais em sua notação, conforme notação ilustrada na Figura 2.31. Em um diagrama de atividades, pode-se ter um símbolo inicial e um símbolo final, porém o símbolo terminal pode ser utilizado mais de uma vez.

Figura 2.31 | Símbolos inicial, final e terminal de um diagrama de atividades



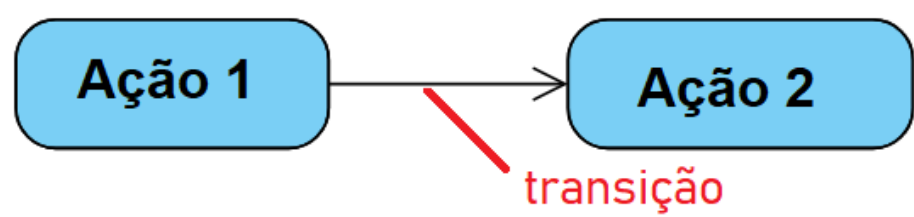
Fonte: elaborada pelo autor.

A diferença entre o símbolo final e o símbolo terminal é que, para um determinado fluxo de controle, se for utilizado o símbolo terminal, significa o final de um fluxo de processos específicos. Nesse caso, não representa o final de todos os fluxos de uma atividade. Por exemplo, caso tenha mais de um fluxo em paralelo, deve ser executado em seguida até que se atinja o símbolo final. Dessa forma, se utilizarmos o símbolo final, representará a finalização de todos os fluxos de um processo. Ao encontrar o símbolo final, a atividade está encerrada e não é necessário que nenhuma outra ação seja executada.

■ AÇÕES INTERNAS DAS ATIVIDADES

As ações internas em uma atividade são representadas por um retângulo de bordas arredondadas, e as setas entre eles, chamadas de transições, representam a mudança de uma atividade para a outra. Esses símbolos estão representados na Figura 2.32.

Figura 2.32 | Exemplos de atividades e transições



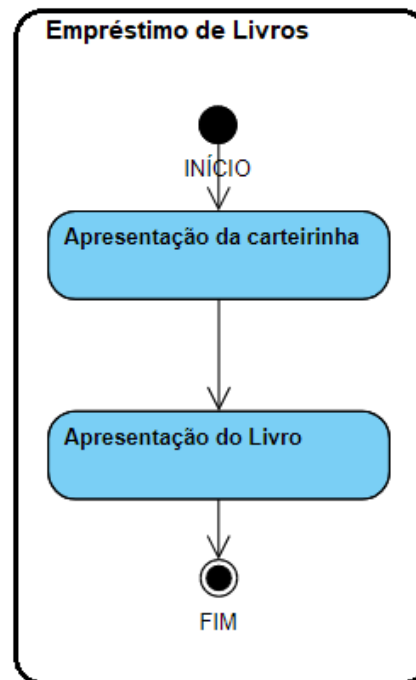
Fonte: elaborada pelo autor.

EXEMPLIFICANDO

Para exemplificar, imagine uma situação em que uma pessoa retira um livro de uma biblioteca. A atividade tratada nesse caso é o empréstimo de um livro. A ideia é que a pessoa apresente o cartão e o livro, então a atividade é

encerrada. O diagrama da Figura 2.33 apresenta a atividade de retirar o livro.

Figura 2.33 | Diagrama de atividade simplificado de empréstimo de livros



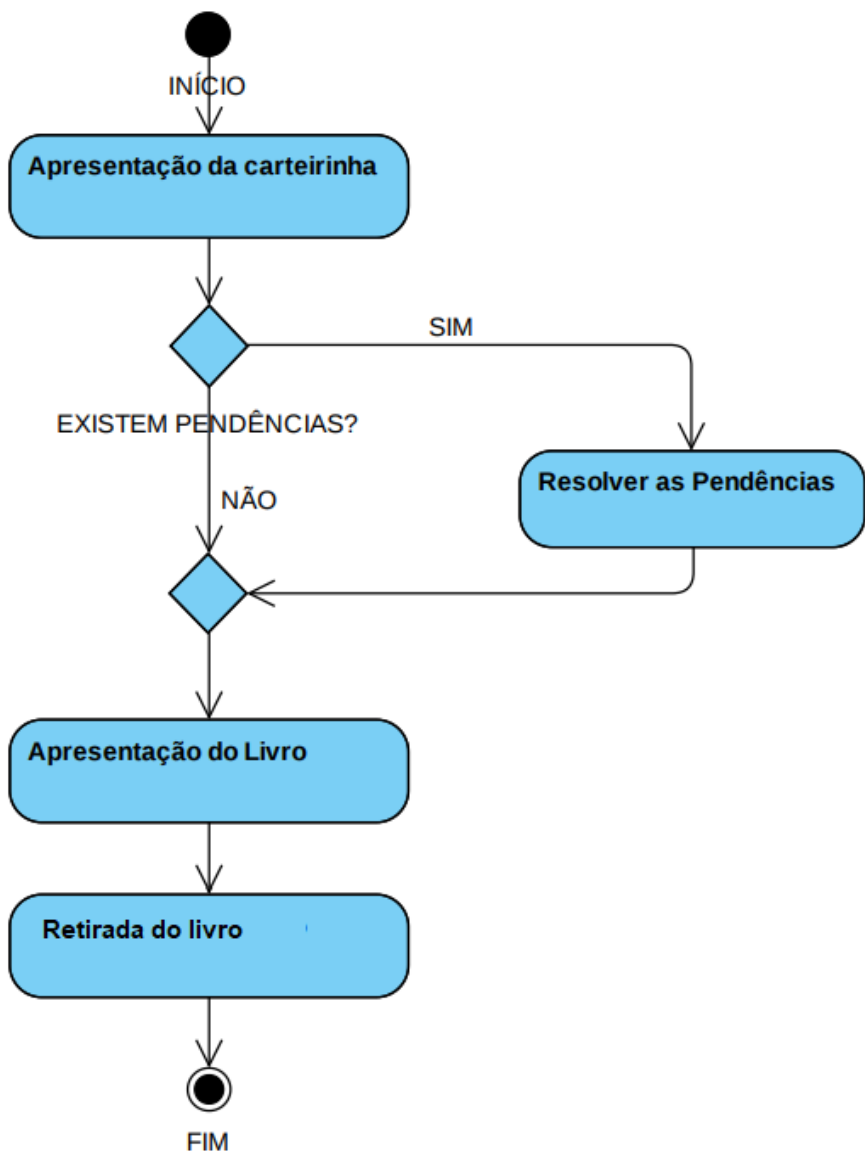
Fonte: elaborada pelo autor.

Neste exemplo, veja que as ações internas, normalmente, possuem nomes que resumem o que estão representando no fluxo de controle.

Agora, considere que existe uma condição a ser expressa no diagrama de atividades, ou seja, existe um teste a ser realizado e, se ele for verdadeiro, o fluxo de controle executará uma ação, mas, se for falso, outra ação. As estruturas condicionais são expressas nos diagramas de atividades como losangos, os quais tanto podem ser utilizados para representar a divisão do fluxo devido a uma condição como para unir dois fluxos para resultarem em uma mesma atividade.

Para entender como construir esta situação, imaginaremos, novamente, a questão do empréstimo de um livro de uma biblioteca. Pense que, ao apresentar a carteirinha para retirar o livro, existe uma verificação para saber se existe alguma pendência no cadastro do usuário (um livro sem devolução ou multas a pagar, por exemplo). Caso não haja, é possível retirar o livro, porém, caso haja, é necessário resolver as pendências antes de retirar o livro. A Figura 2.34 apresenta uma possibilidade de diagrama de atividades considerando esta situação nova que aumenta a complexidade do diagrama.

Figura 2.34 | Empréstimo de livros com condicional



Fonte: elaborada pelo autor.

Nota-se, na Figura 2.34, que o primeiro losango representa a condicional se existem ou não pendências no cadastro do usuário. Caso haja, o usuário deve resolvê-las para, depois, pegar o livro; caso não haja, é possível pegar o livro de qualquer forma. O segundo losango é responsável por reunir os fluxos que foram divididos, já que resultam na mesma atividade seguinte.

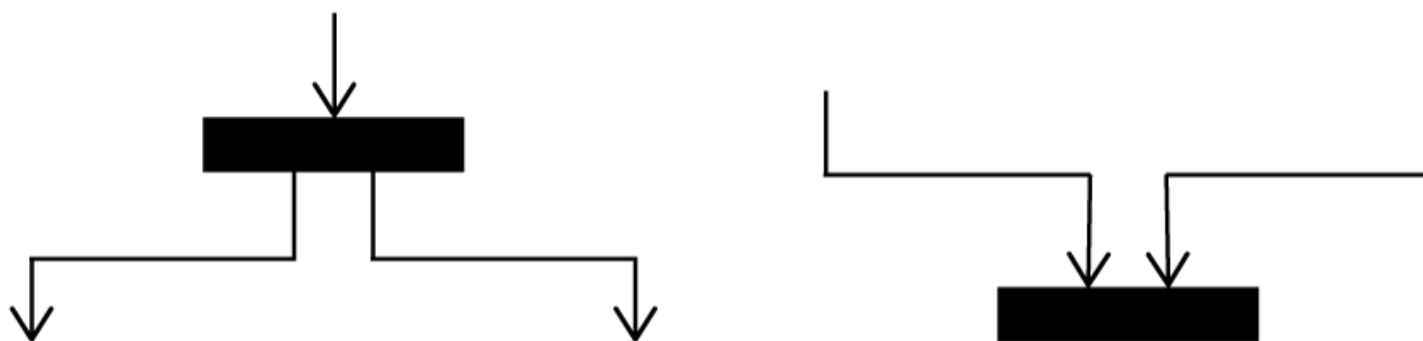
Observe que a utilização de operadores condicionais está condicionada à execução de um determinado requisito por vez, no entanto, em muitos casos, há a necessidade de representar dois ou mais fluxos de controle que podem ser executados concorrentemente.

■ BARRAS DE SINCRONIZAÇÃO

Além disso, podemos dividir um fluxo de controle em múltiplos fluxos executados concorrentemente. Nesse caso, pode-se utilizar as barras de sincronização. Para isso, existem dois componentes: um chamado **FORK** e outro chamado **JOIN**. Ambos são representados por uma barra transversal sólida. A diferença é que o FORK representa a bifurcação (divisão) de um fluxo de controle em múltiplos fluxos que podem ser executados simultaneamente. Já o JOIN mostra a união de dois ou mais fluxos executados concorrentemente. A Figura 2.35 ilustra a notação de ambos, respectivamente.

o
Ver anotações

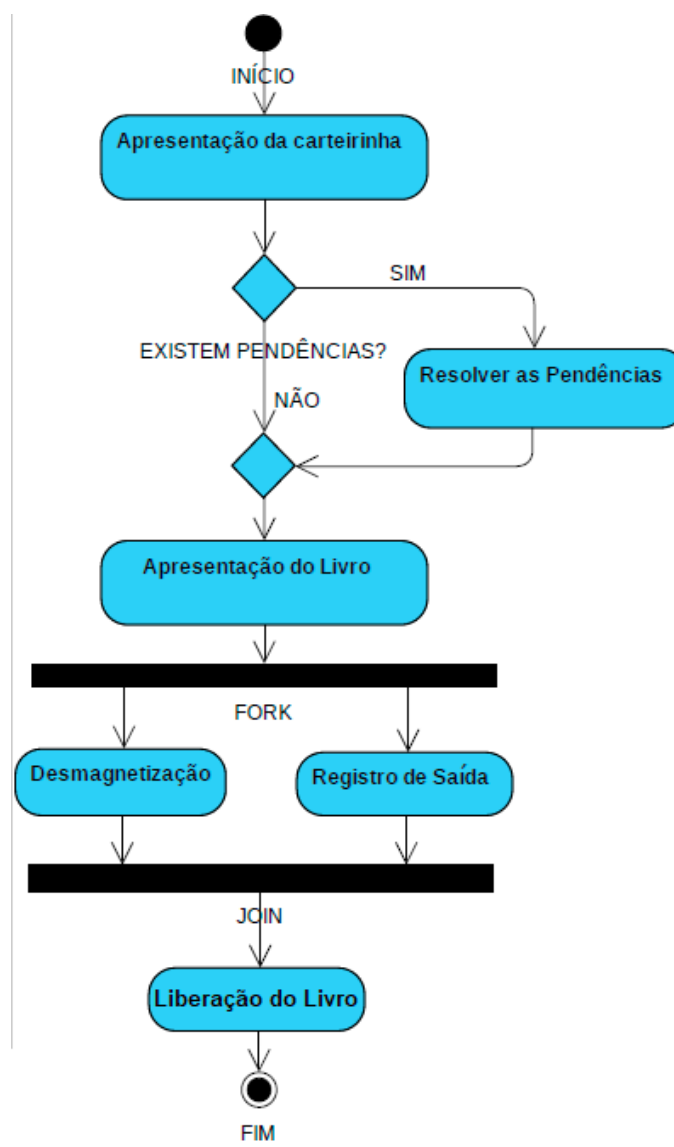
Figura 2.35 | Notação dos componentes FORK e JOIN



Fonte: Visual Paradigm (2020).

Um exemplo de aplicação de FORK e JOIN pode ser apresentado na retirada do livro. Imagine que, após a verificação, temos duas operações para serem realizadas: a **desmagnetização do livro** e o **cadastro da saída no sistema**. Pode-se considerar que ambas as operações podem ser executadas simultaneamente para a **liberação do livro**, assim temos a aplicação do JOIN. O FORK pode ser visto na bifurcação (divisão) da apresentação do livro para a desmagnetização e o registro de saída. A Figura 2.36 apresenta o diagrama com estas mudanças.

Figura 2.36 | Empréstimo de livros com FORK e JOIN



Fonte: elaborada pelo autor.

ASSIMILE

As operações representadas por componentes FORK e JOIN são paralelas e ocorrem de forma concorrente. Isso quer dizer que ocorrem ao mesmo tempo em dois fluxos de controle diferentes. Esse processo pode também ser entendido como uma sincronização.

Sincronizar operações é o ato de iniciá-las ao mesmo tempo e somente continuar o fluxo após todas terem terminado.

No exemplo da Figura 2.36, ao realizar um JOIN antes de liberar o livro, estamos querendo dizer que o livro só será liberado após o cadastro de saída do sistema e a desmagnetização terem terminado. Não importa se uma operação demorar mais tempo do que outra, a questão é que o fluxo só continua quando as duas terminarem, portanto é necessário sincronizá-las.

Ao realizar um FORK e um JOIN, pode-se afirmar que naqueles pontos deve existir uma barreira que sincroniza operações tanto de início de fluxos paralelos quanto de junção de fluxo após o término de operações.

Agora que aprendemos os componentes básicos de um diagrama de atividades, precisamos entender algumas questões sobre os fluxos.

SWIMLANES

O fluxo de controle visto até o momento era sequencial até que sofria uma divisão em um FORK e uma junção em um JOIN. Essa divisão tornava o fluxo de controle paralelo em certo ponto. Entretanto, é possível pensar em um fluxo sequencial e/ou paralelo que ocorra em mais de um local.

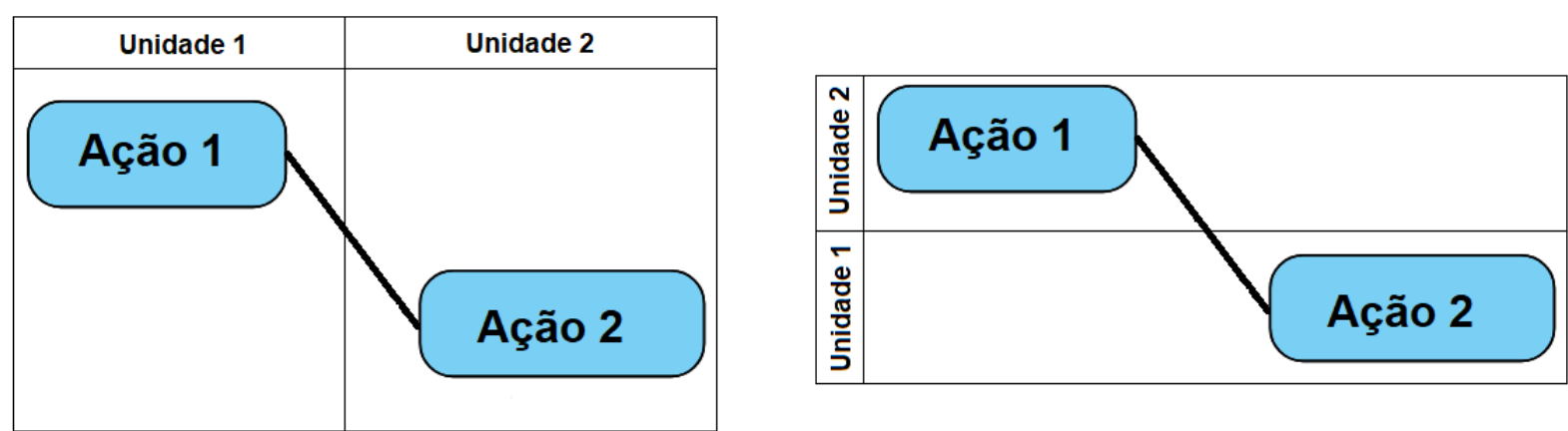
Imagine, ainda no caso do sistema de empréstimo de livros, que existe um servidor que roda o banco de dados da biblioteca e pode ser tratado em três dimensões no diagrama de atividades: do usuário, do atendente e do sistema. Para esse caso, podemos organizar de uma forma que possibilite que o diagrama de atividades possua as três visões.

O componente utilizado para realizar essa divisão é chamado de *swimlane*, que é uma partição lógica. Existem diversas possibilidades de se organizar o diagrama de atividades com relação aos casos de uso, aos processos e até mesmo aos objetos.

Em cada uma das “pistas” (do inglês, *lane*) do *swimlane* é colocado o nome da

unidade que se deseja mapear as atividades e, logo abaixo, são organizadas as atividades relativas àquela entidade/objeto/unidade. A Figura 2.37 apresenta duas formas de se indicar as *swimlanes*.

Figura 2.37 | Exemplos de utilização de swimlanes



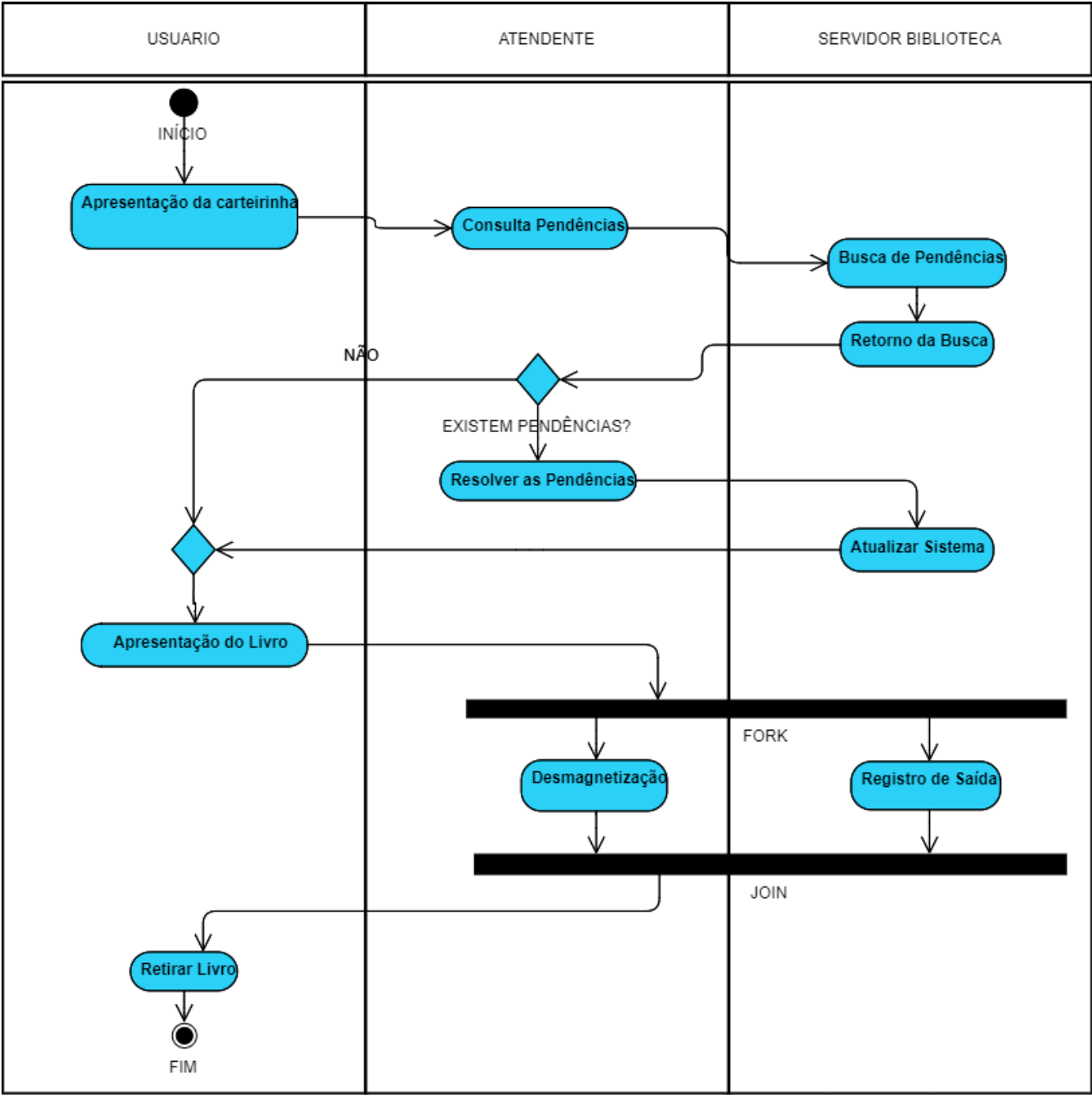
Fonte: elaborada pelo autor.

As *swimlanes* apresentadas na Figura 2.37 estão indicadas por retângulos que envolvem as ações da atividade relativas a cada unidade. Podem ser descritas de forma horizontal e vertical, devendo ser escolhida a disposição que apresenta o melhor entendimento do diagrama. Ao se construir *swimlanes* no diagrama de atividades, é necessária a reorganização das atividades de modo que cada unidade possua aquelas relativas a ela sem prejudicar o fluxo de controle.

Aplicaremos o conceito ao diagrama de atividades do empréstimo de livros. Considerando o que foi proposto, seriam necessárias três dimensões: uma para o usuário, uma para a atendente (ou para o sistema da biblioteca) e uma para o servidor que contém o banco de dados da biblioteca.

A Figura 2.38 ilustra uma das possibilidades de representação com o diagrama de atividades. No caso do usuário, ele apresenta a carteirinha; em seguida, o livro; e, por fim, realiza a retirada. A atendente consulta no sistema se existem pendências que poderiam impedir a realização do cadastro do usuário para o empréstimo. Caso existam pendências, ela poderá resolver e, logo depois, o usuário poderá apresentar, desmagnetizar e, finalmente, retirar o livro. Além disso, pode-se descrever os passos do sistema relacionados à busca sobre as pendências retornando o resultado para a atendente, a atualização dos dados do usuário, caso as pendências tenham sido resolvidas, e o registro da saída do livro.

Figura 2.38 | Utilizando swimlanes para organizar o diagrama de empréstimo de livro



Fonte: elaborada pelo autor.

ATENÇÃO

Importante observar que, ao adicionar as *swimlanes* neste diagrama de atividades, algumas partes foram modificadas, sendo necessárias algumas adições para a completude do diagrama. Essas mudanças mostram que a representação anterior possuía diversas omissões que poderiam comprometer o desenvolvimento do sistema. Este caso mostra claramente que, quando o diagrama de atividades é implementado de forma correta, pode-se obter uma representação completa do sistema, sendo possível até visualizar possíveis falhas na etapa de desenvolvimento.

No diagrama da Figura 2.38, também é possível observar que existem partes do fluxo que são sequenciais até que se encontra o FORK, o qual transforma o fluxo de controle em paralelo. A parte paralela do diagrama ocorre na desmagnetização do livro pela atendente, na qual o sistema automaticamente registra a saída do material. A sequência de ações é retomada após o JOIN e resulta na retirada do

livro pelo usuário. Todas as outras atividades apresentadas no diagrama ocorrem de forma sequencial, apesar de estarem em partições diferentes. Repare na representação dos dois componentes, cujas barras estão entre duas pistas, indicando que cada atividade ocorre em um local diferente.

REFLITA

Nesta seção, discutimos a relação entre o diagrama de atividades e os diagramas de casos de uso. Sabemos que estes sofrem modificações ao longo do processo de desenvolvimento, principalmente no início. Reflita sobre qual é o momento certo de fazer os diagramas de atividades. Seria no início, para auxiliar no desenvolvimento dos casos de uso? Após a primeira versão do diagrama de casos de uso validada pelo cliente, para que não haja trabalho desnecessário? Ou seria interessante fazer os diagramas de atividades em todas as etapas do processo e ver suas mudanças? Lembre-se de que o tempo necessário para fazer os diagramas pode impactar no tempo de desenvolvimento de software.

Chegamos ao final da seção sobre os diagramas de atividades. Neste momento, você é capaz de identificar os principais componentes de um diagrama, analisar e construir um diagrama a partir de um problema e fazer o processo de melhorias até que se obtenha um diagrama completo, cuja representação é o que se esperava. Aliando este diagrama aos diagramas de casos de uso e de classes, agora, você possui conhecimento sobre três importantes representações existentes na UML. Chegou a hora de testar seus conhecimentos e praticar o que você aprendeu. Preparado? Bons estudos!

REFERÊNCIAS

REGGIO, G.; LEOTTA, M.; RICCA, F.; CLERISSI, D. What are the used activity diagram constructs? A survey. *In*: INTERNATIONAL CONFERENCE ON MODEL-DRIVEN ENGINEERING AND SOFTWARE DEVELOPMENT, 2., 2014, Lisboa. **Anais [...]**. Lisboa: MODELSWARD, 2014.

UML-DIAGRAMS. **UML Activity Diagram Examples**. 2016. Disponível em: <https://bit.ly/2EXl88e>. Acesso em: 8 jul. 2020.

VISUAL PARADIGM. **What is Activity Diagram?**. 2020. Disponível em: <https://bit.ly/3bjs9wh>. Acesso em: 27 ago. 2020.