

UML Atividades

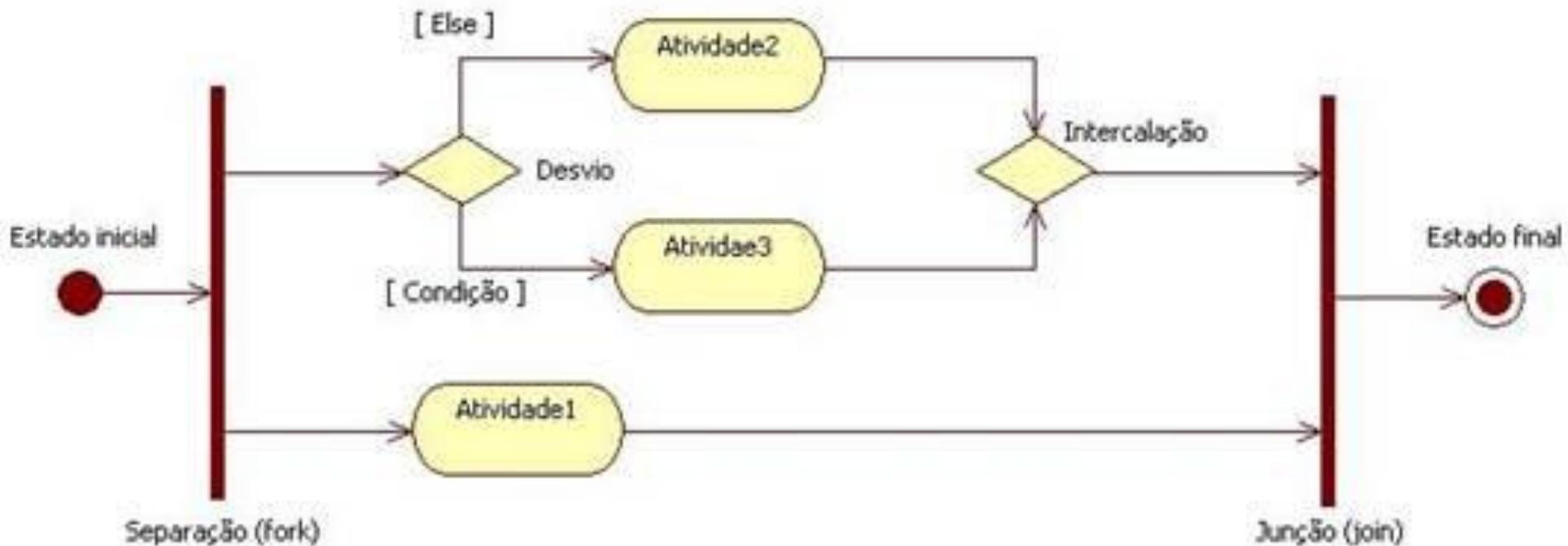
UML Atividades

O diagrama de atividades tem por objetivo demonstrar a perspectiva comportamental de processos. Uma atividade é composta por uma sequência estruturada de ações. Nesta estrutura podem seguir um ou mais fluxos, que podem por sua vez tomar outros caminhos através de desvios. Em outras palavras, este diagrama mostra as atividades que compõem um processo e seu fluxo de controle. Mostra também a execução de atividades sequenciais ou concorrentes.

Em seu conceito mais simples o diagrama de atividades até pode ser confundido com um fluxograma, porém o que o distingue de um fluxograma é a possibilidade de criar fluxos que seguem em paralelo, podendo se encerrar ao mesmo tempo ou até terem finais separados. Esta funcionalidade é muito importante, pois com isso pode-se seguir com ações do mesmo segmento e de setores distintos como, por exemplo, a preparação de um produto para a venda, pode-se ter o setor de produção e o setor de vendas dispostos no mesmo. A possibilidade de separar o diagrama por partições/setores é também uma das funcionalidades que o distinguem de um fluxograma, pois o torna muito mais organizado, portanto, mais legível.

Veja na figura 1 um exemplo da estrutura e de alguns elementos do diagrama de atividades.

UML Atividades



Neste exemplo já podemos ver vários elementos do diagrama de atividades como o estado inicial e o estado final, a atividade, a separação (*Fork*) e a junção (*Join*), o desvio (deve ter condições de guarda para as suas saídas) e a intercalação. Para facilitar o entendimento sobre estes elementos e outros iremos descrever abaixo alguns conceitos e particularidades sobre cada um deles.

UML Atividades

Elementos do diagrama de atividades

Nos próximos tópicos nos aprofundaremos mais nos conceitos de cada elemento demonstrado no diagrama da figura 1 e mais outros elementos.

Action – ação

Uma *Action* é representada por um retângulo com os cantos arredondados, o nome que descreve a atividade fica disposto no interior do retângulo (ver figura 2). Este elemento representa um passo a ser executado, dentre vários que possam ter a fim de executar um processo ou uma atividade completa, cada um destes passos é uma ação a ser executada.

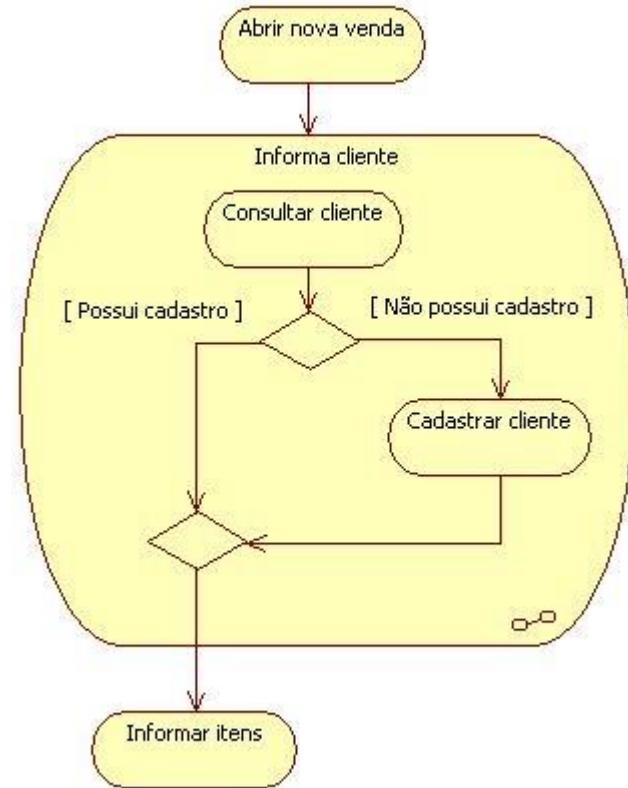
Figura 2 – Exemplo de Action



UML Atividades

Subactivity – subatividade

A representação gráfica da *Subactivity* é semelhante a uma *Action*, com a diferença de ter um símbolo no canto inferior direito que indica que se trata de uma *Subactivity* (ver figura 3). Este elemento é uma atividade que se divide em ações para se completar. É utilizado quando há necessidade de especificar uma atividade mais abrangente.



UML Atividades

Transições

As transições são representadas por uma seta contínua (ver figura 3). As transições fazem a ligação entre um elemento e outro do diagrama. Elas determinam o sentido que o fluxo deve seguir até que se chegue ao final do processo.

Eventos

Os eventos são alterações deliberadas de estado que dão início a uma ação. Falaremos de dois tipos de eventos, o *Signal Accept State* e o *Signal Send State*.

• **•*Signal Accept State***: este elemento representa a execução instantânea de um evento e logo em seguida a execução de uma *Action* subsequente ao evento;

• **•*Signal Send State***: este elemento representa a execução automática de um evento logo após o término da execução de uma *Action*.

exemplos da representação gráfica dos dois eventos citados.



UML Atividades

Decision or Merge – decisão ou fusão

Este elemento é representado graficamente por um losango. O que determina se é uma decisão ou uma fusão é a quantidade de entradas e saídas (transições). Por exemplo, se tivermos um elemento com uma entrada e duas ou mais saídas, este por sua vez é um elemento do tipo *Decision*. Se tivermos um elemento com duas ou mais entradas e apenas uma saída, este será o elemento do tipo *Merge*.

Quando for um elemento do tipo *Decision* cada transição de saída deverá ter uma condição de guarda entre colchetes. Ao chegar num elemento do tipo *Decision* o fluxo deverá seguir pela condição que for satisfeita (verdadeira). Estas condições não podem ser redundantes, isto é, o fluxo deverá seguir por apenas um caminho, não poderão existir duas condições verdadeiras.

Em outras palavras, o uso do elemento *Decision* se dá em situações condicionais, ou seja, dependendo de uma condição o fluxo segue por um lado, senão vai por outro. Já o uso do elemento *Merge* se dá para situações em que se faz necessário fazer a junção dos fluxos para seguir em um único fluxo.

Veja na figura 7 um exemplo de utilização do elemento *Decision/Merge*.



UML Atividades

Synchronization (fork/join) – Sincronização (bifurcação/junção)

O elemento *Synchronization* é representado por uma barra sólida, ela pode ser disposta tanto na horizontal quanto na vertical, isso irá depender do sentido pelo qual os fluxos irão seguir. Este elemento, assim como o elemento anterior, se divide em duas funções denominadas *Fork* e *Join*.

O uso do tipo *Fork* se dá para dividir o fluxo do diagrama em atividades paralelas, ou seja, que serão executadas em modo concorrente. Esta função é uma das que tornam este diagrama tão dinâmico e flexível. O outro tipo de sincronização, o *Join*, é utilizado para unir as atividades que estão correndo em paralelo, seguindo assim em apenas um fluxo. Simplificando um pouco mais o entendimento deste elemento, o *Fork* faz a divisão das atividades e o *Join* faz a sincronização das mesmas.

Veja na figura 8 um exemplo de uso destes elementos.



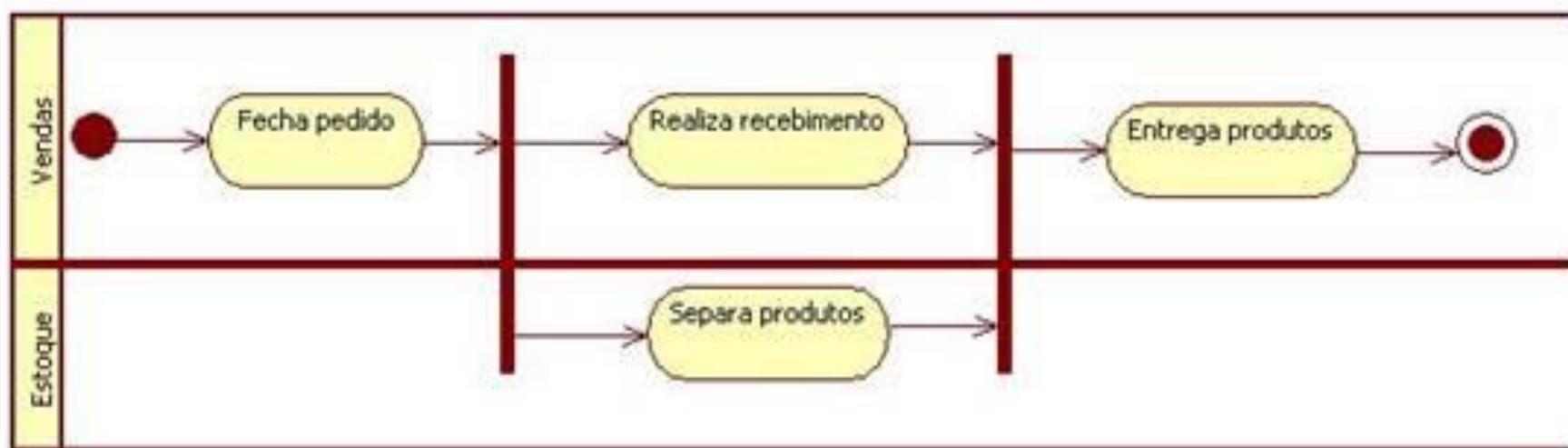
UML Atividades

Swimlanes – raias (de natação) ou partições

Este elemento é muito importante para a organização do diagrama. Com o *Swimlane* é possível demonstrar como as atividades podem ocorrer com diferentes agentes dentro de um processo, cada *Swimlane* recebe o nome do agente que representa.

Com o uso deste elemento pode-se por exemplo detalhar as atividades de um caso de uso, definindo cada ator do caso de uso como uma *Swimlane*.

Veja na figura 9 o exemplo anterior com a utilização dos elementos *Swimlanes*, demonstrando os diferentes agentes, Vendas e Estoques, atuando de forma concorrente no decorrer do processo.



UML Atividades

As *Swimlanes* podem ser utilizadas de várias formas. A utilização deste componente facilita o entendimento do processo do diagrama. Veja na figura 10 outro exemplo de utilização deste elemento.

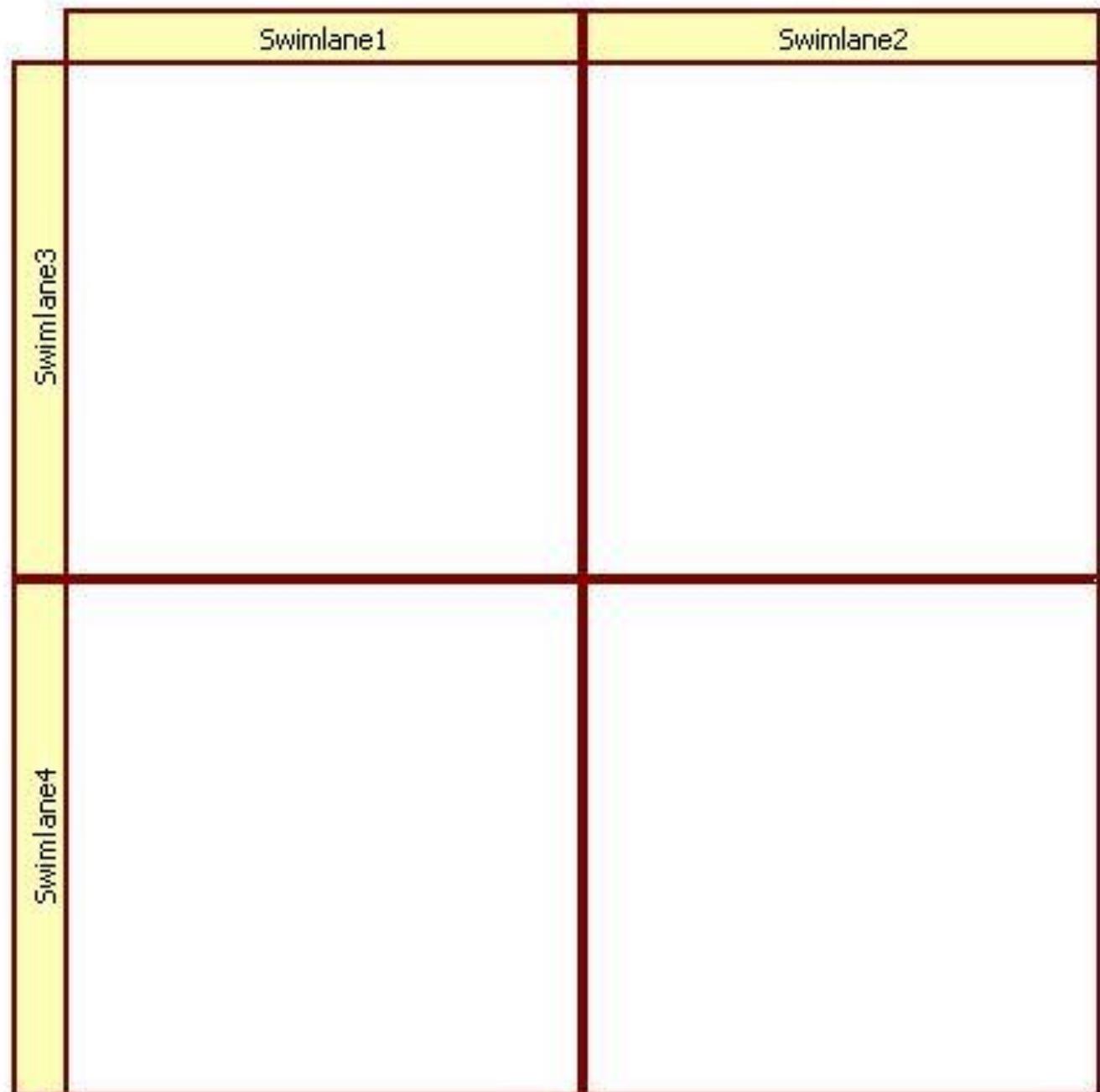


Diagrama de Atividades

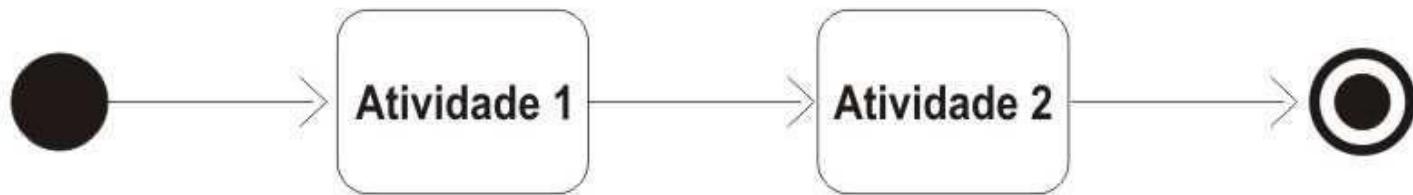
Diagrama de Atividades

- Um diagrama de atividades é um caso particular de um diagrama de estados, no qual todos ou a maioria dos estados são “estados de atividades” e todas ou a maioria das transições são desencadeadas pela conclusão das atividades dos estados anteriores;
- Uma atividade corresponde a uma execução não atômica dentro de uma máquina de estados, ou de outra forma, corresponde à execução de um conjunto de ações.

Diagrama de Atividades x Estados

- Ambos os tipos de diagramas são utilizados para modelar o tempo de vida de um objeto ou sistema.
- Contudo, um diagrama de atividades ilustra o **fluxo de controle entre atividades**, enquanto que um diagrama de estados **ilustra o fluxo de controle entre estados**.

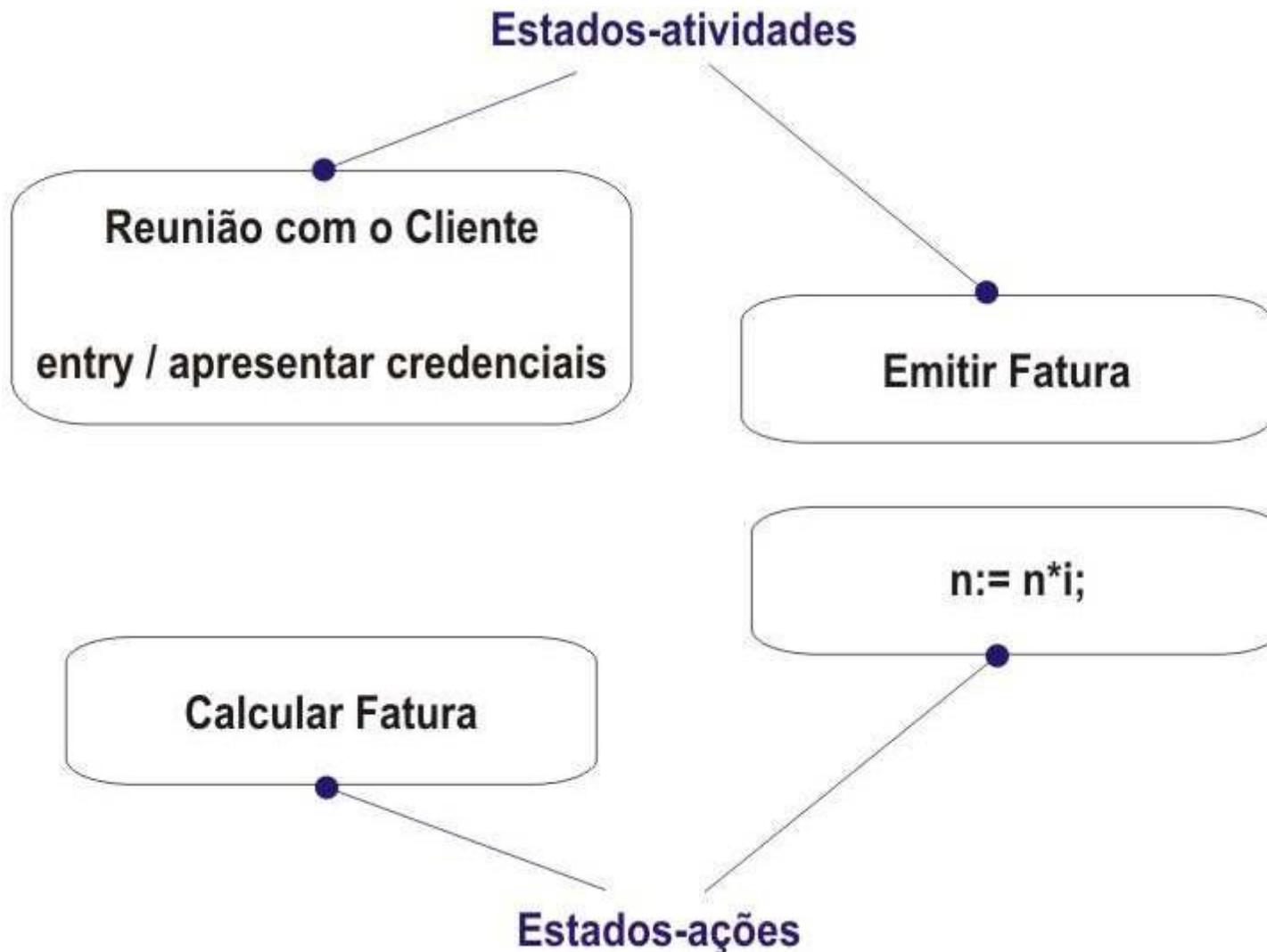
Exemplo Genérico de Diagrama de Atividades



Diagramas de Atividades

- Estes diagramas contêm genericamente:
 - **Estados-ação**: execuções atômicas, não interrompíveis, com tempo de execução irrelevante.
 - **Estados-atividade**: execuções não atômicas (decompostas), interrompíveis, em que o tempo de execução é normalmente relevante.
 - **Transições**,
 - **Objetos**.

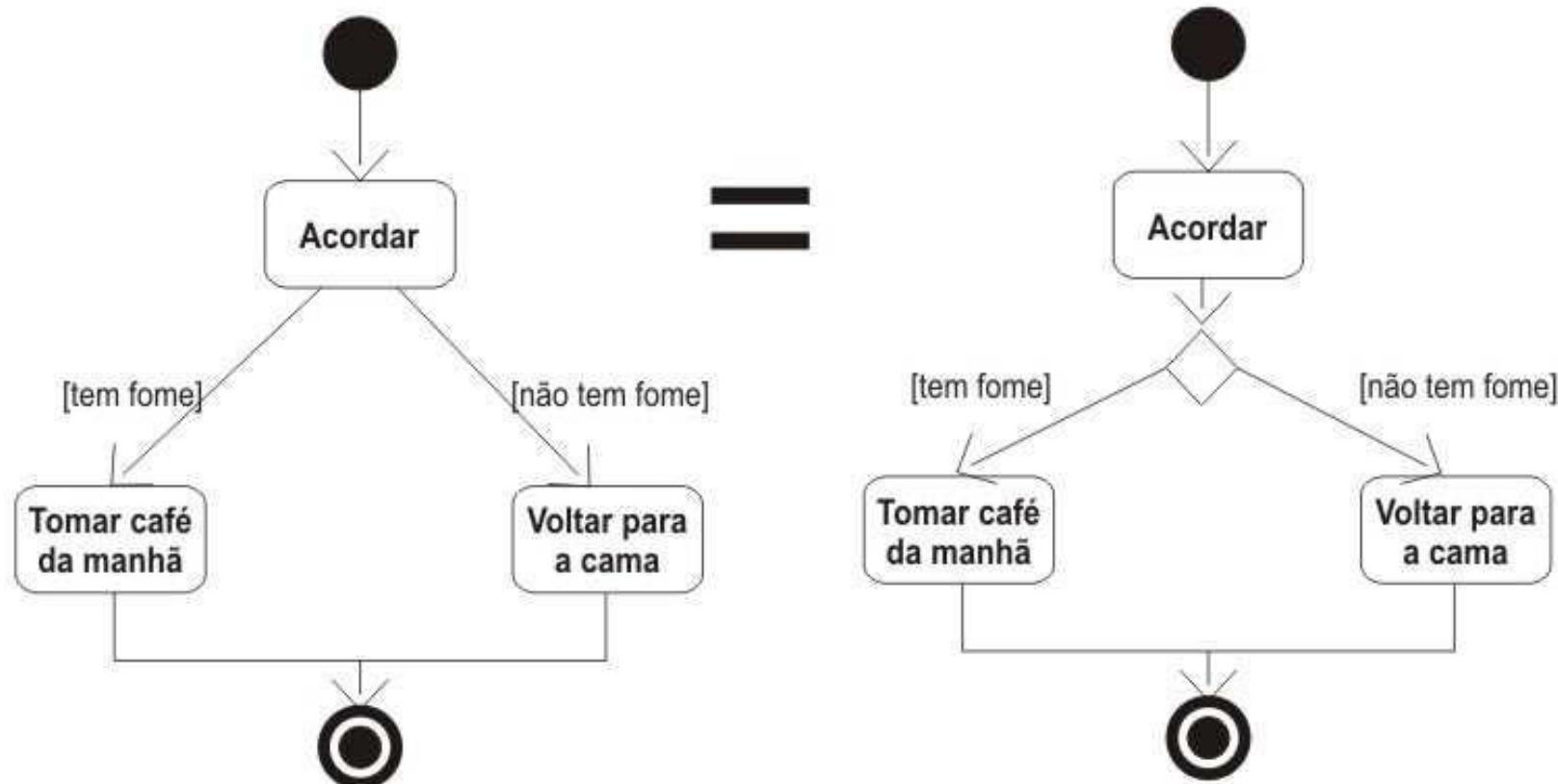
Estados-atividades X estados-ações



Diagramas de Atividades (Decisões)

- A tomada de decisão é um mecanismo comum no projeto de diagramas de atividades (e de estado), que consiste em especificar que atividade deve ser realizada após a execução da atividade corrente.
- Tal especificação é suportada por uma condição com guarda (ex.: expressão lógica) que é colocada de forma adjacente à transição de estado correspondente.

Diagramas de Atividades (Decisões) - Exemplo

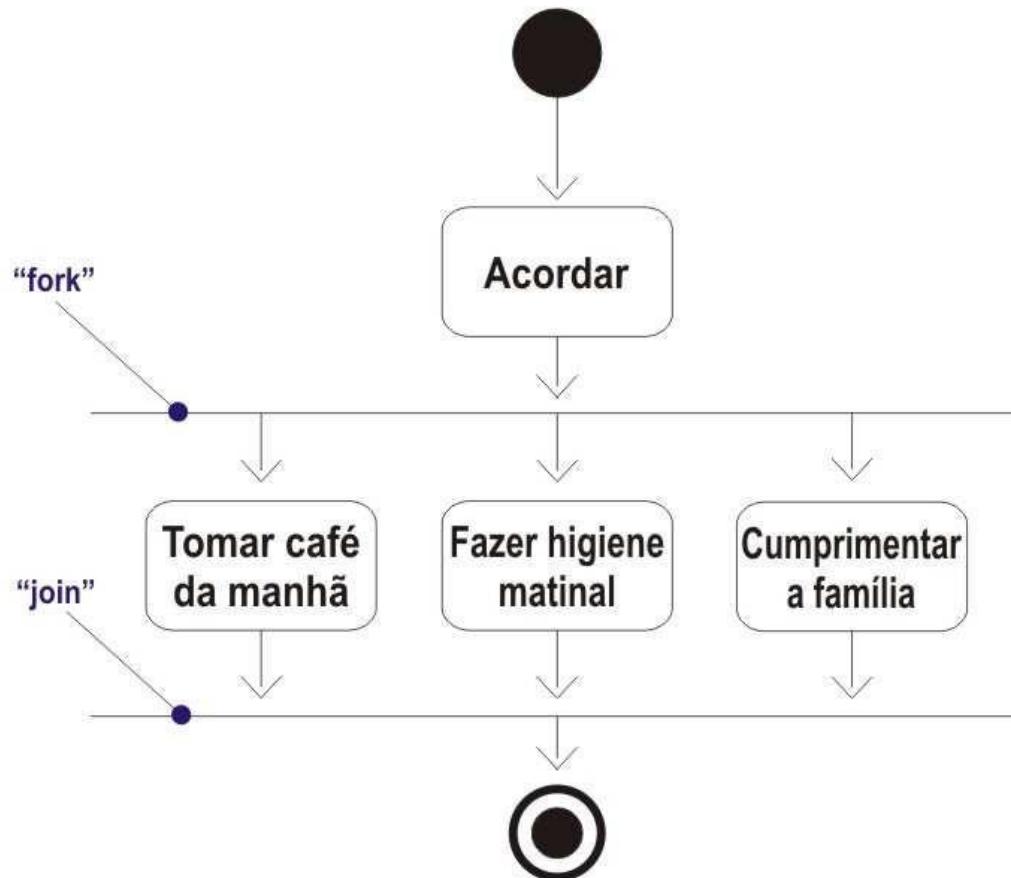


Diagramas de Atividades (Caminhos Concorrentes)

- Considere que o processo de “levantar da cama” implica a execução das seguintes atividades “**tomar café da manhã**”, “**fazer a higiene matinal**” e “**cumprimentar a família**”.
- Considere que essas atividades têm de se realizar obrigatoriamente, embora não seja relevante a sua ordem de execução.
- O problema colocado representa uma situação típica na modelagem de workflows: **representar a execução independente e concorrente de um conjunto de atividades**.

Diagramas de Atividades (Caminhos Concorrentes) – cont.

A UML providencia a solução a esta questão através dos conceitos de difusão (fork) e de junção (join) de atividades, representados graficamente por linhas



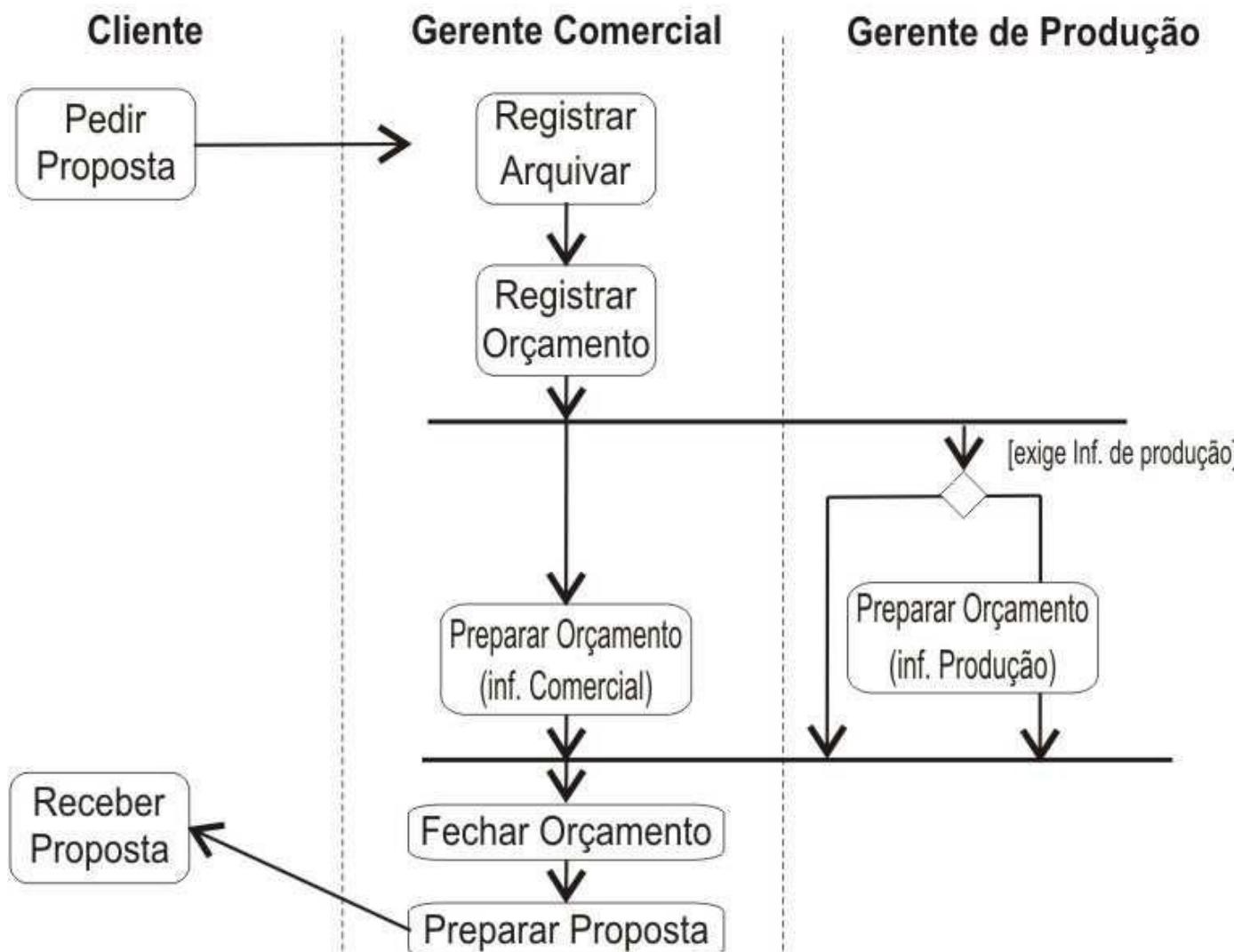
Diagramas de Atividades (Raias)

- Na modelagem de processos de negócio é comum a realização de atividades por várias entidades, participantes no dito processo.
- A UML propõe o conceito de pistas (swimlanes) como elemento que permite agrupar as várias atividades da responsabilidade de cada entidade participante. Cada grupo é separado por uma linha vertical.

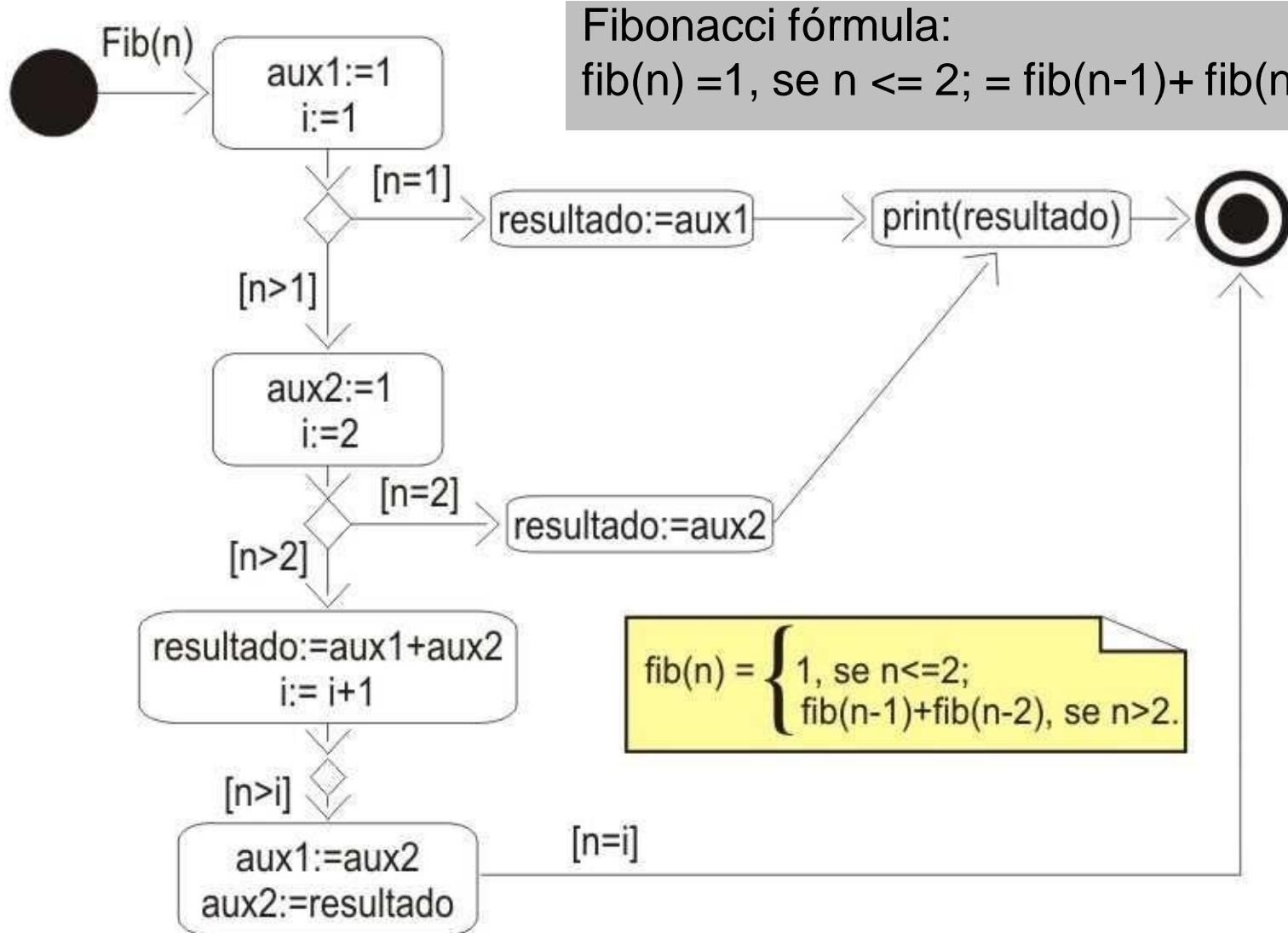
Diagramas de Atividades (Raias) – Cont.

- Cada pista tem um nome único dentro do seu diagrama, que deve corresponder ao nome da entidade participante, a qual deve ser uma entidade do mundo real.
- Por exemplo, o nome de um perfil de usuário, o nome de uma organização, ou o nome de um sistema de informação.

Diagramas de Atividades (Pistas) – Exemplo



Exemplo de Diagrama de Atividades da Operação de Fibonacci



Exercício

- Faça um diagrama de Atividades para a situação:
 - Retirando dinheiro de um caixa eletrônico (para cartões de crédito)

