

Tópico 5 – UML - Diagrama de Atividade

Luiz Antônio M. Pereira

lpereira@uninet.com.br

lpereira@luizantoniopereira.com.br



DA - Motivação

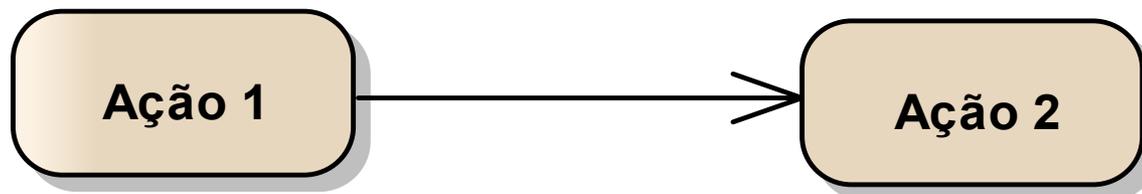
- Em muitas situações precisamos especificar
 - Os passos necessários para a realização de uma tarefa
 - Executada em paralelo por mais de um executor;
 - Onde os passos se estruturam de forma não sequencial;
 - Com desvios causados por diversas condicionalidades;
 - Onde artefatos são produzidos e trocados entre os executores.
 - Um algoritmo complexo
 - Envolvendo processamento em paralelo, pontos de sincronismo, desvios múltiplos (*switches*), intercalações, etc.

DA - Motivação

- Diagramas de Atividade da UML são expressivos o suficiente para a modelagem completa de processos com muitas ações executadas em paralelo.

DA - Conceitos

- Antes da UML 2.0 eram entendidos como variantes do DTE, onde os estados são estados de atividade;



- Transição entre "Ação 1" e "Ação 2" só é trilhada quando "Ação 1" termina, ou seja, o evento que provoca a transição é o evento de fim de "Ação 1".



DA - Conceitos

- Dos diagramas que já existiam na UML, foi o que mais sofreu inovações na versão 2.0;
- DAs incorporaram nos últimos anos mecanismos para tratamento de paralelismo, partições hierarquizadas e em duas dimensões, pinos etc.

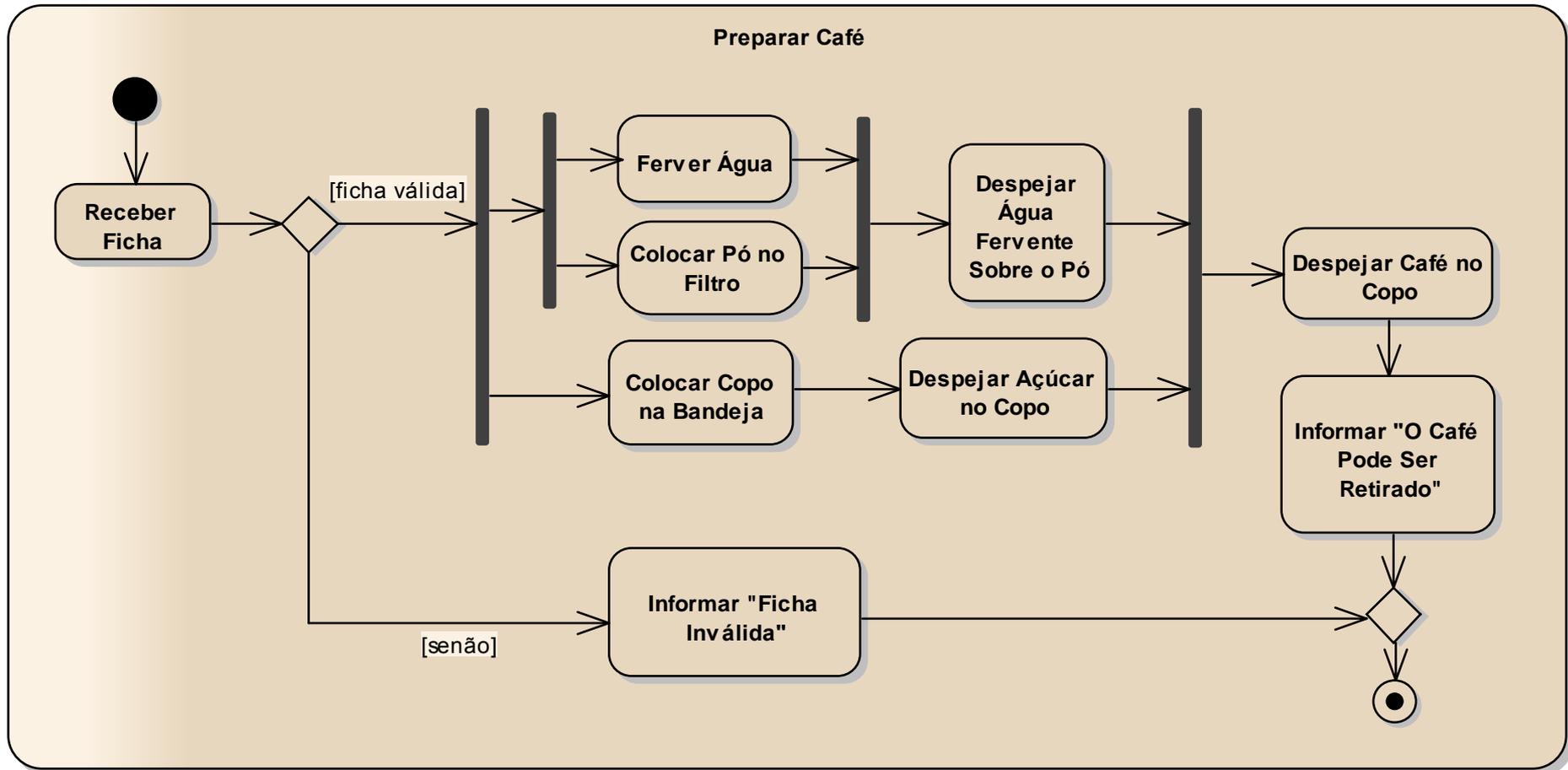
DA - Conceitos

- Enfocam o fluxo de controle entre ações do sistema (visão dinâmica);
- Esses diagramas especificam a ordem de execução das ações, cobrindo, portanto, parte da dimensão temporal do modelo de um sistema;
- Úteis para modelagem de programas concorrentes, onde se projetam graficamente as *threads* e pontos de sincronismo.

DA - Conceitos

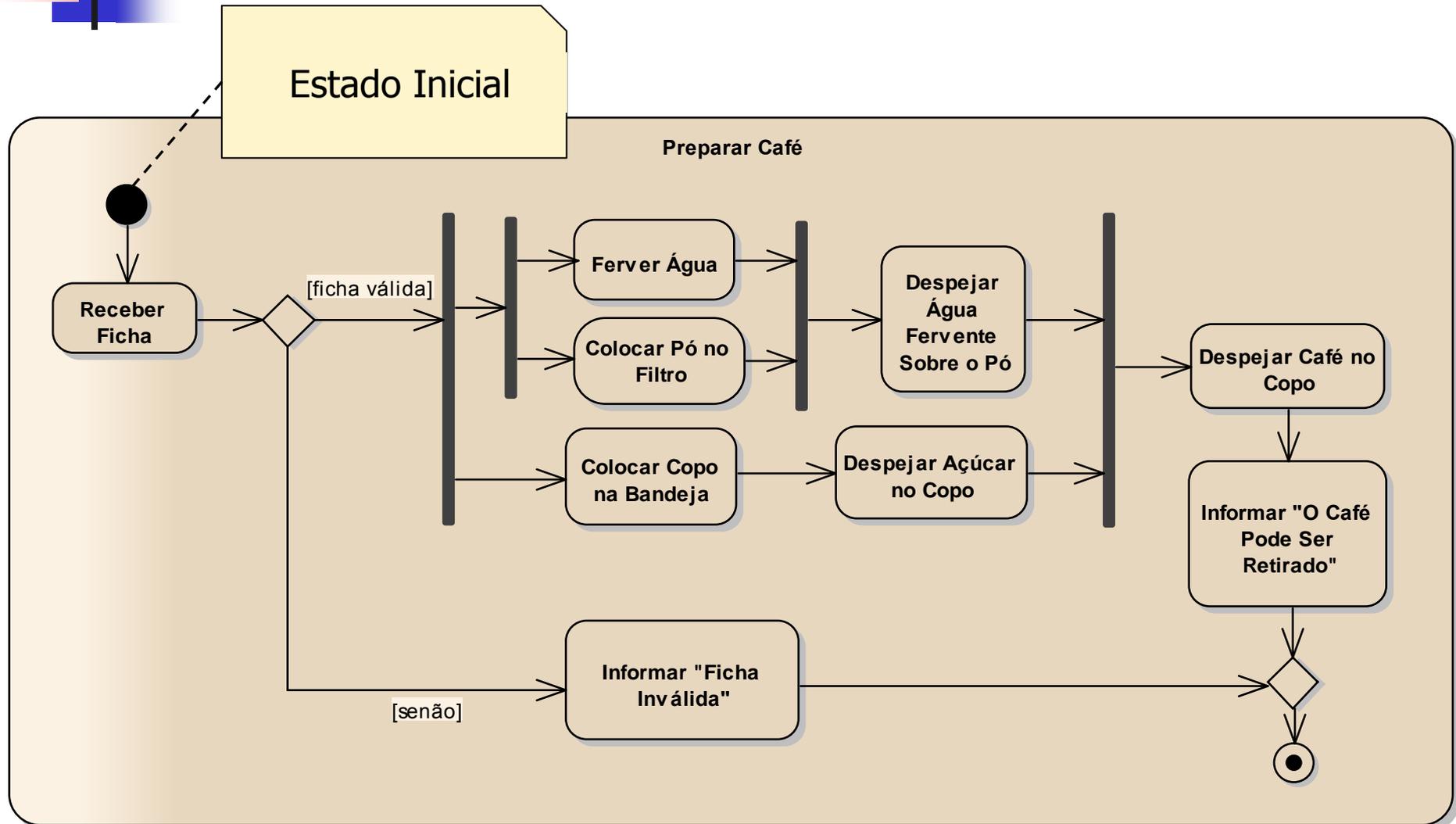
- Vêm sendo usados em inúmeras outras aplicações, além das tradicionais especificações de algoritmos complexos:
 - Processos de engenharia;
 - Processos de negócios (fluxos de trabalho - *workflows*);
 - ...

DA - Elementos da Notação

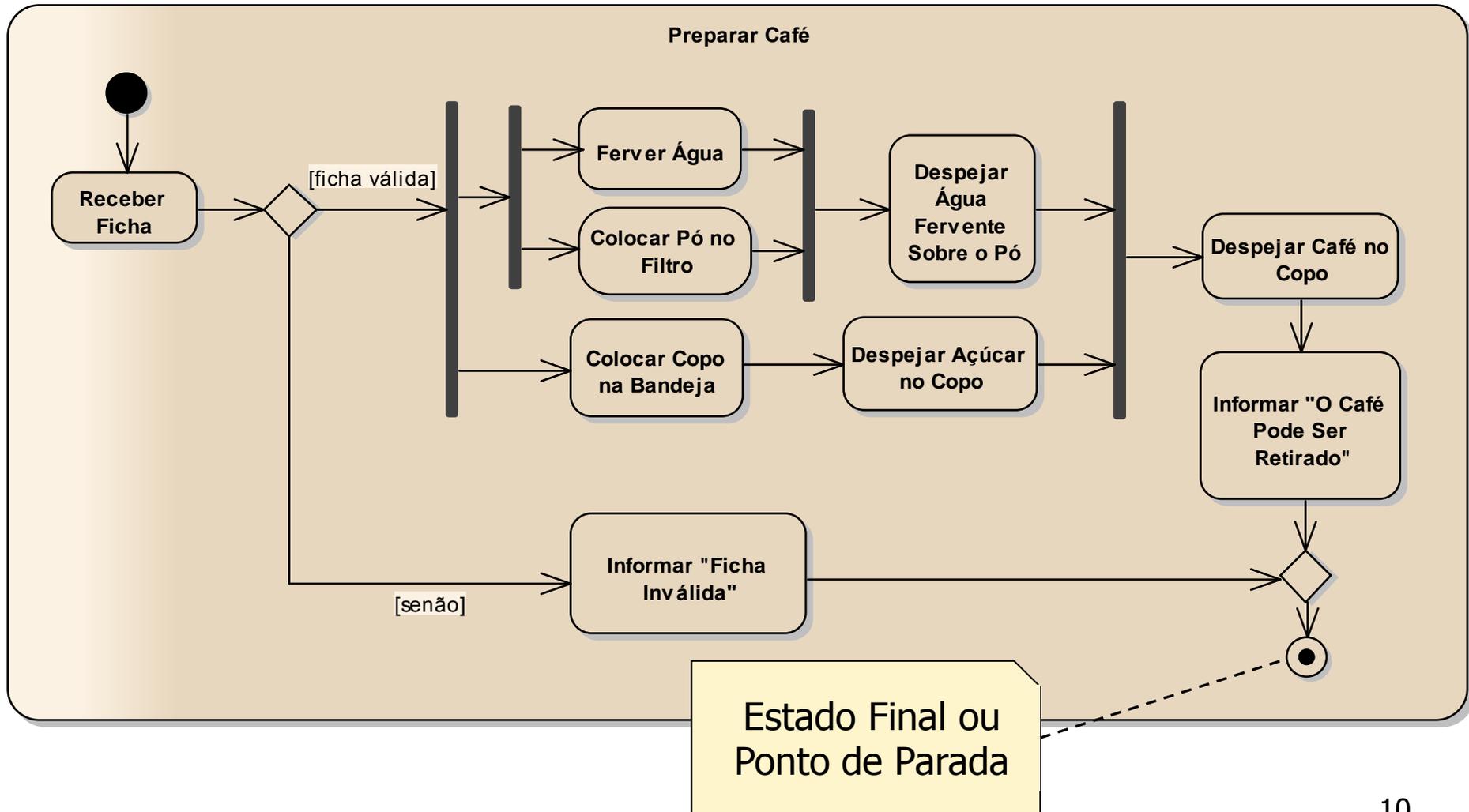




DA - Estados Inicial e Final



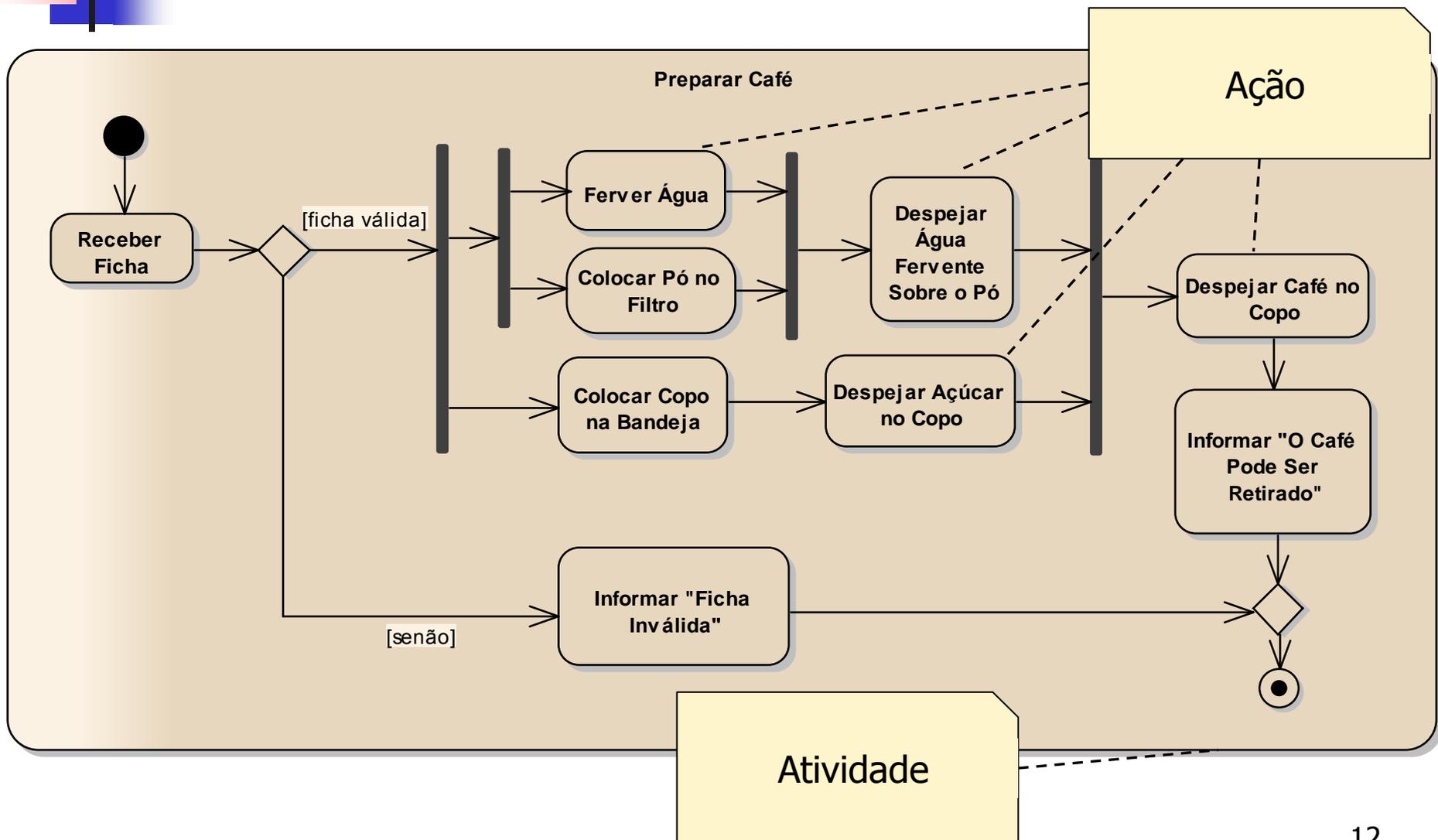
DA - Estados Inicial e Final



DA - Estados Inicial e Final

- Estado inicial:
 - Indica o ponto de início da atividade;
 - Só pode ter um no mesmo contexto.
- Estado final:
 - Pode haver mais de um no mesmo diagrama;
 - Quando atingido, a máquina é encerrada, indicando o ponto de término da atividade.

DA – Atividade e Ações



DA – Atividade e Ações

- Exemplos de ações:

Preparar
Café

Imposto = Renda
Líquida * 0,20

- Preferencialmente usar nomes com verbos no infinitivo.

DA – Atividade e Ações

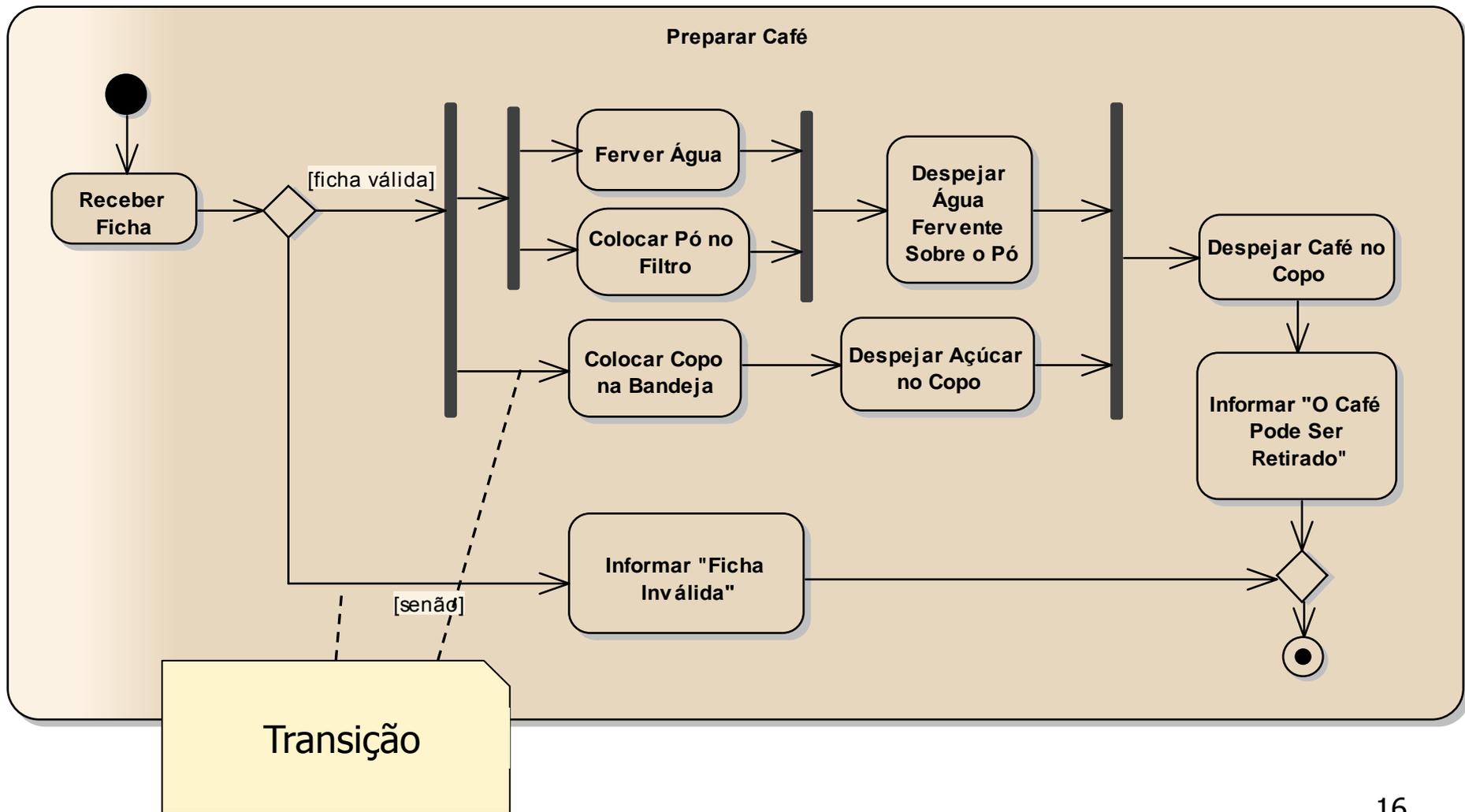
- Ação é a unidade fundamental para a especificação de um comportamento (UML);
 - Representa algo indivisível (atômica);
- Atividade é composta de ações e/ou sub-atividades.



DA – Atividade e Ações

- Uma ação recebe um conjunto de entradas e as converte em um conjunto de saídas, embora ambos os conjuntos possam ser vazios;
- O conjunto de entradas para uma ação pode ser resultado das saídas de uma ou mais ações executadas anteriormente.

DA - Transições



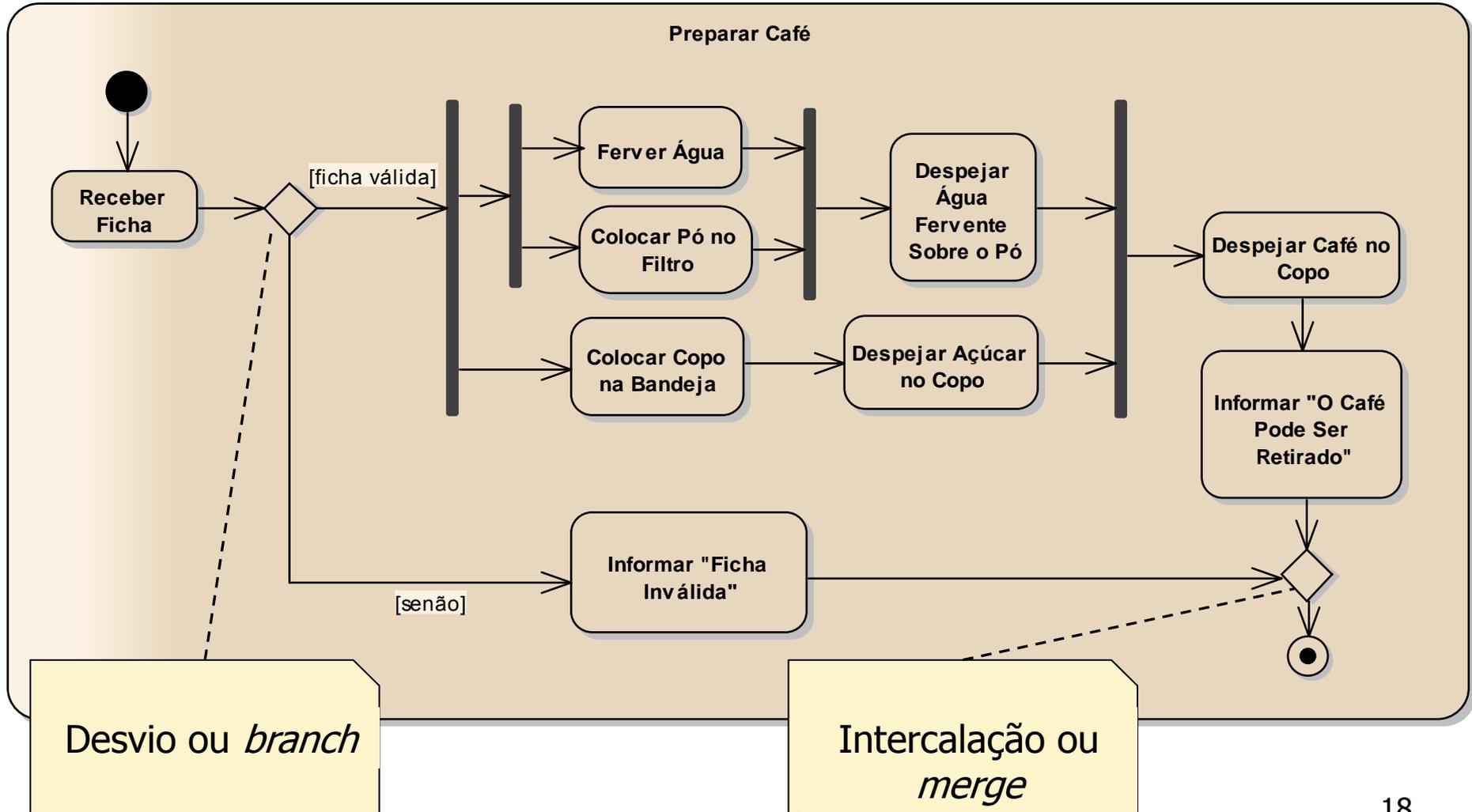


DA – Transições

- Também chamadas de fluxos;
- São trilhadas quando as atividades origem terminam.
- São não-qualificadas; ou
- Qualificadas com “guardas”: expressões lógicas que indicam as condições em que são trilhadas. As guardas são colocadas entre “[” e “]”;



DA – Transições





DA – Transições

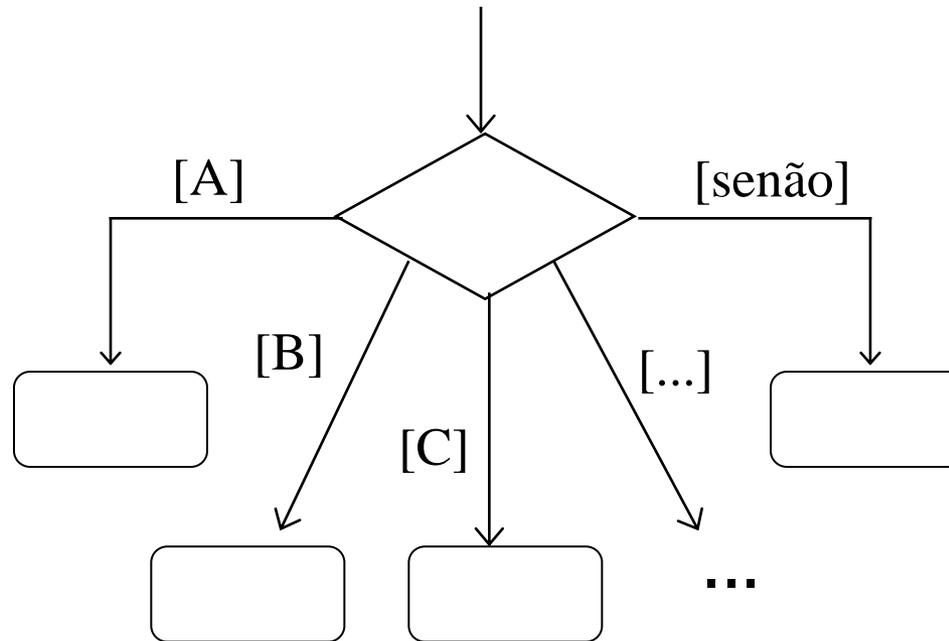
- Um desvio ou nó de decisão é um nó de controle que escolhe entre fluxos de saída;
- Fluxos de saída obrigatoriamente guardados com as respectivas condições de saída;



DA – Transições

- De um desvio podem partir vários fluxos, desde que as guardas especifiquem condições lógicas mutuamente excludentes.
- Em outras palavras: um desvio corresponde a **um** fluxo de controle de entrada com **um** fluxo de controle de saída.

DA – Transições



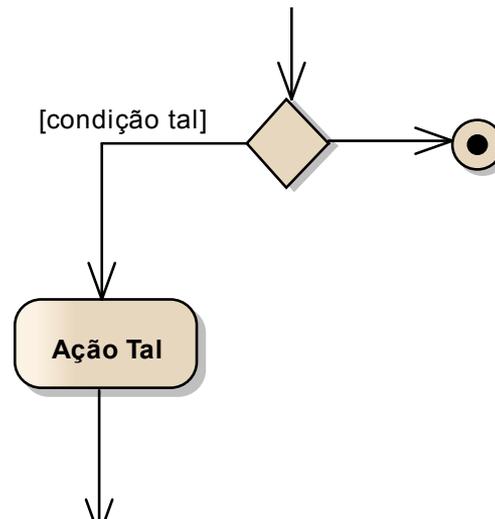
DA – Transições

- A contrapartida de um nó de decisão é um nó de intercalação;
- Recebe os vários fluxos de entrada (que correspondem aos desvios) e tem um único fluxo de saída.



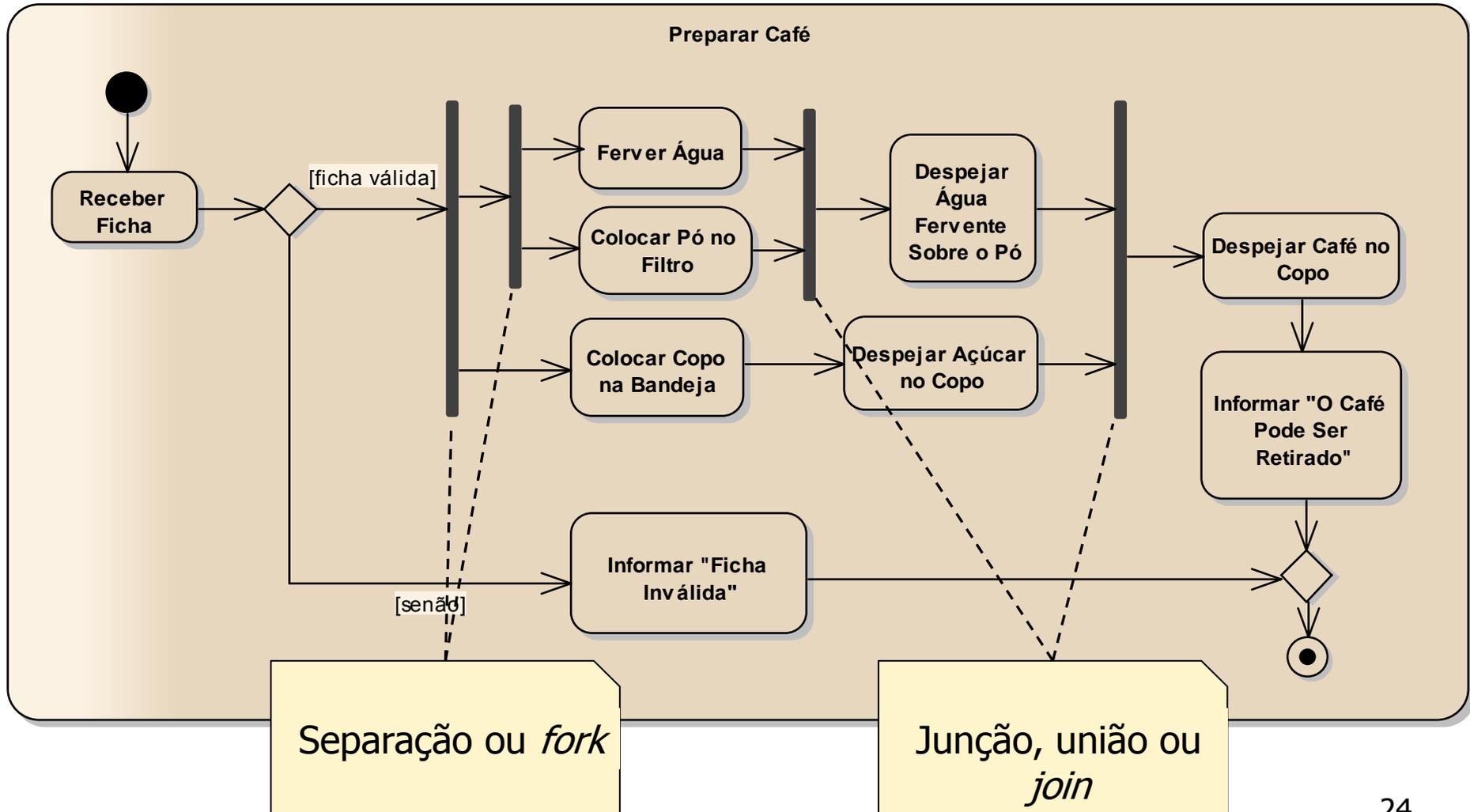
DA – Transições

- Uma intercalação tem que ser colocada sempre que colocamos um nó de decisão, a menos que o fluxo termine ou que outra decisão precise ser tomada.





DA – Separações e Junções





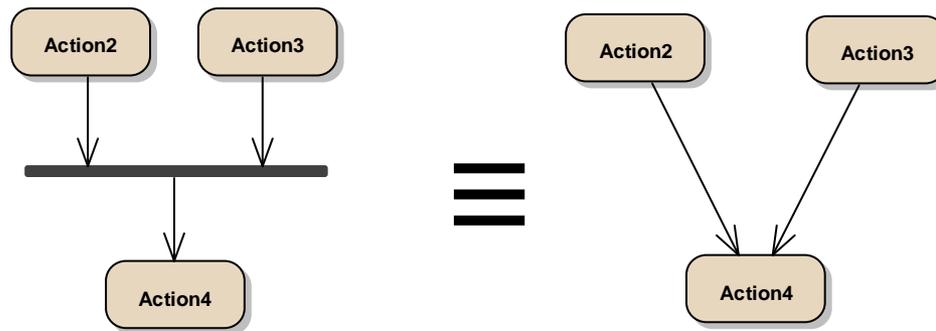
DA – Separações e Junções

- Separações:
 - Uma transição de entrada e várias transições de saída;
 - As atividades ligadas às transições de saída são executadas em paralelo (*threads* ou fios de execução);
 - Em outras palavras: um *fork* corresponde a **um** fluxo de controle de entrada com **dois ou mais** fluxos de controle de saída.

DA – Separações e Junções

■ Junções:

- São as contrapartidas das separações;
- Marcam pontos de sincronismo;
- Processamento só passa da junção quando **todos** os fios de execução que nela convergem são terminados;
- A representação no modelo não é obrigatória pela UML. Múltiplos fluxos de entrada em uma ação caracterizam uma junção implícita.





DA - Partições

- Partições:
 - Partições ou raias (*swimlanes*) são usadas quando há necessidade de se indicar quem executa as atividades;
 - As ações executadas por determinado executor/setor são colocadas na raia desse determinado executor/setor;
 - As raias podem ser hierarquizadas;
 - As raias podem ser bi-dimensionais.

DA - Partições

Processamento	Despacho	
	Estoque	

		B	
		B1	B2
A	A1		
	A2		

DA – Fluxo de Objetos

- Objetos (artefatos) podem ser passados de uma ação a outra durante a execução de uma atividade;
- Objetos são denotados por retângulos que se situam no fluxo entre as ações que os passam e as que os recebem:



DA – Fluxo de Objetos

- O nome do objeto deve ser colocado no interior do retângulo na forma *objeto:classe*,
- A referência à instância ou à classe (uma ou outra) é opcional.
- Nomes aceitáveis:
 - copiaFatura:Fatura, copiaFatura, :Fatura

DA – Atividades Aninhadas

- Exemplos:

Emitir Boleto de Mensalidade



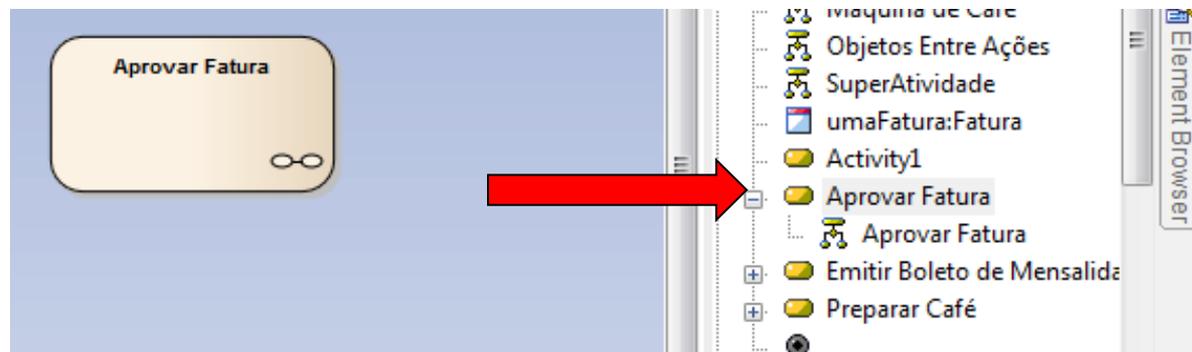
Emitir Boleto de Mensalidade



Atividades aninhadas são detalhadas em outros diagramas

DA – Atividades Aninhadas

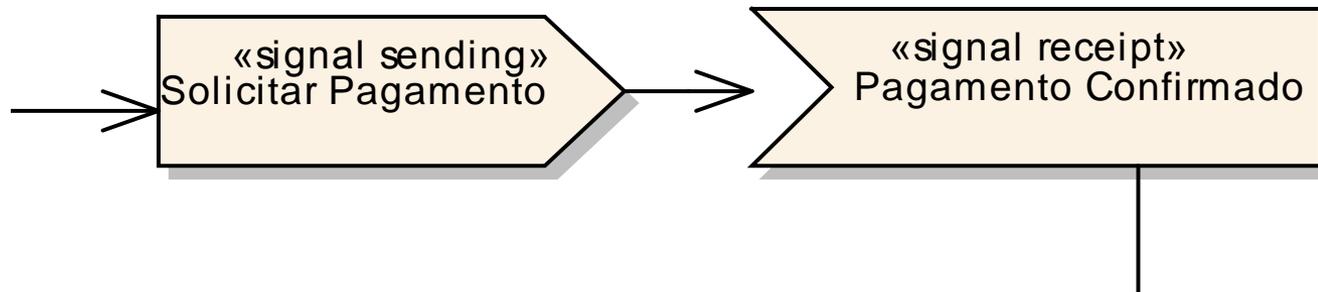
- Para manterem a consistência do modelo, os CASEs
 1. Criam um diagrama para modelarmos a atividade aninhada;



2. Permitem que arrastemos um diagrama para dentro de outro.

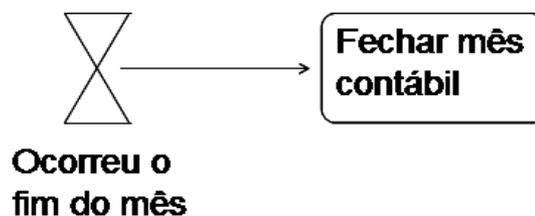
DA – Sinais e Eventos Temporais

- O envio de um sinal em um modelo de sistema pode estar associado ao lançamento de uma interrupção;
- A recepção de um sinal pode estar associada à execução da ação de tratamento dessa interrupção.

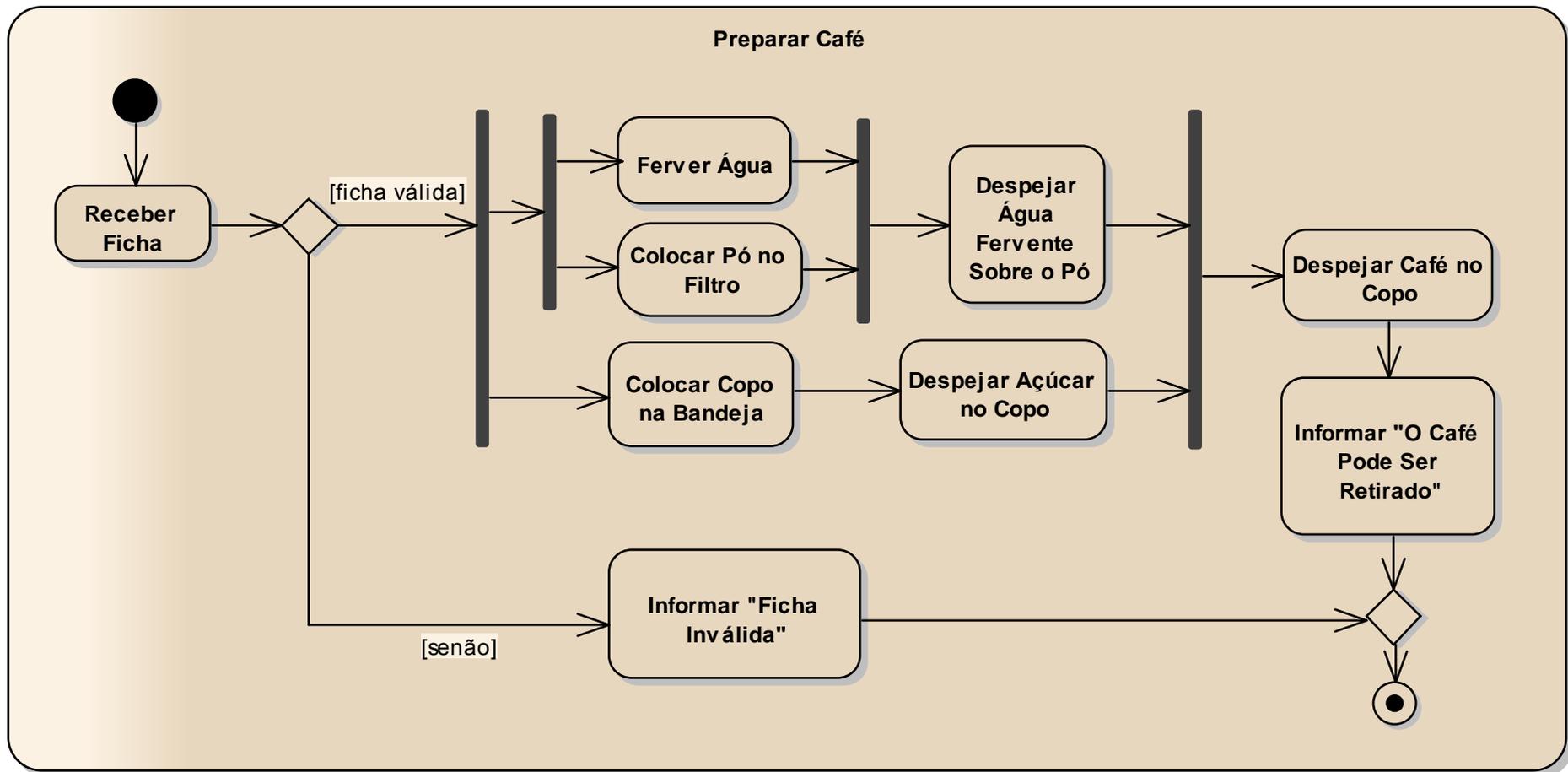


DA – Sinais e Eventos Temporais

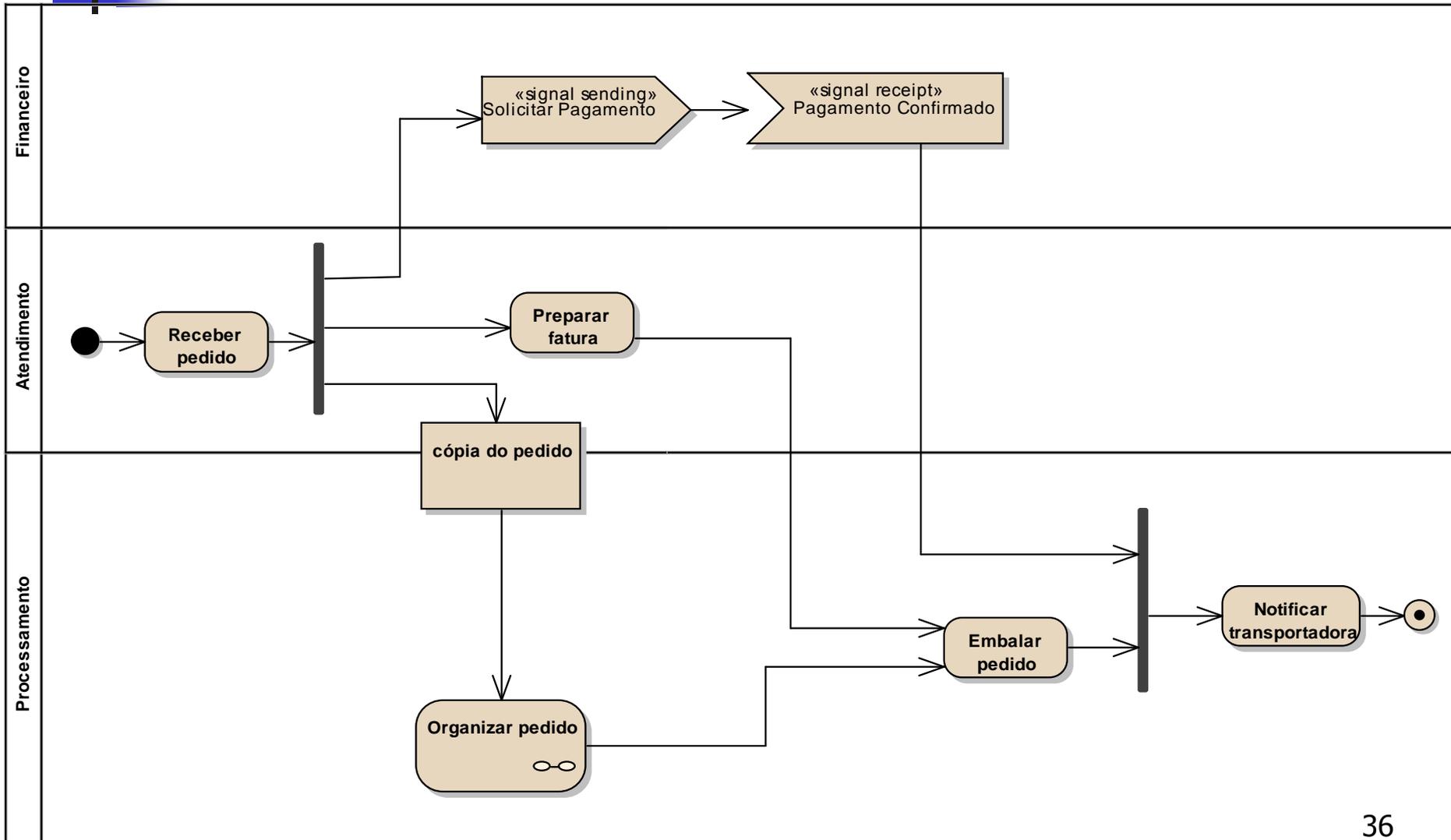
- Um evento temporal é algo que ocorre
 - em um instante determinado (evento temporal fixo), ou
 - após determinado tempo medido com relação à ocorrência de outro evento (evento temporal relativo);
- Um evento temporal pode dar origem a um fluxo.

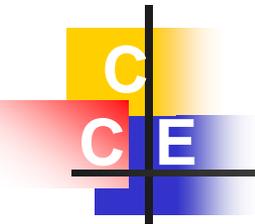


DA – Consolidando...



DA – Consolidando...





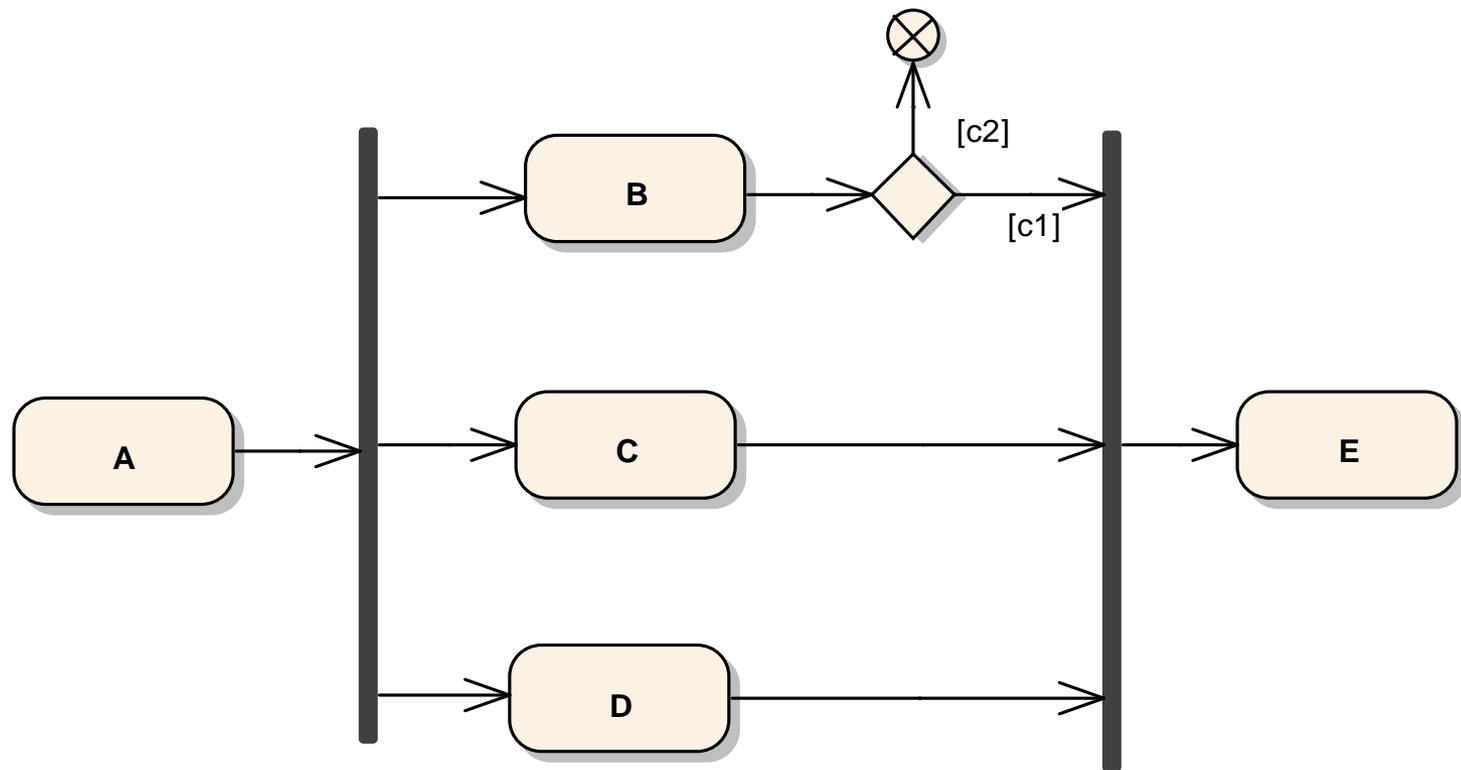
Aspectos Complementares

DA – Fim de Fluxo

- Também chamado de *cancelamento*;
- Usado para denotar que um fio de execução terminou, mas a máquina continua ativa;
- Diferente do ponto de parada, que especifica que TODA a máquina se encerra naquele ponto;
- Notação:

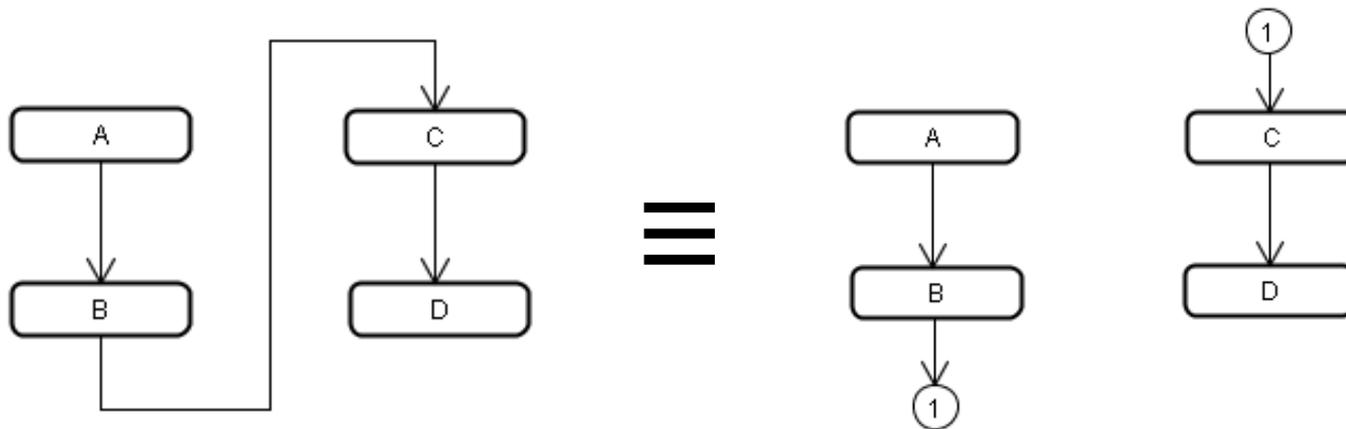


DA – Fim de Fluxo



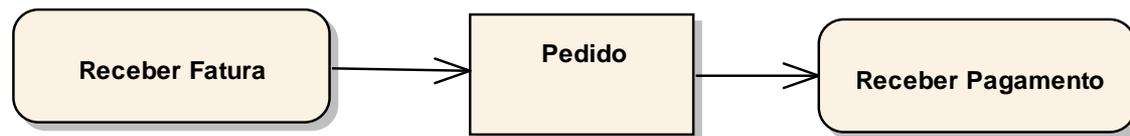
DA - Conectores

- Usados para evitar longas setas de fluxos;
- Usados em quebras de páginas.

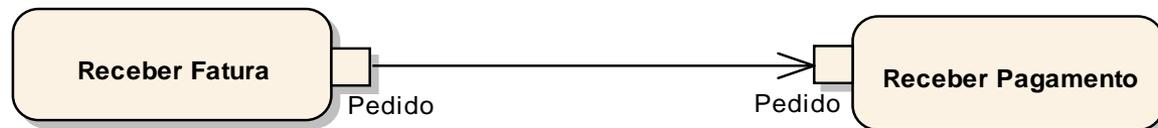


DA - Pinos

- Uma ação pode receber parâmetros (objetos, valores, listas de objetos ou de valores etc.) da ação anterior
- Uma ação pode passar parâmetros para a ação seguinte na sequência.

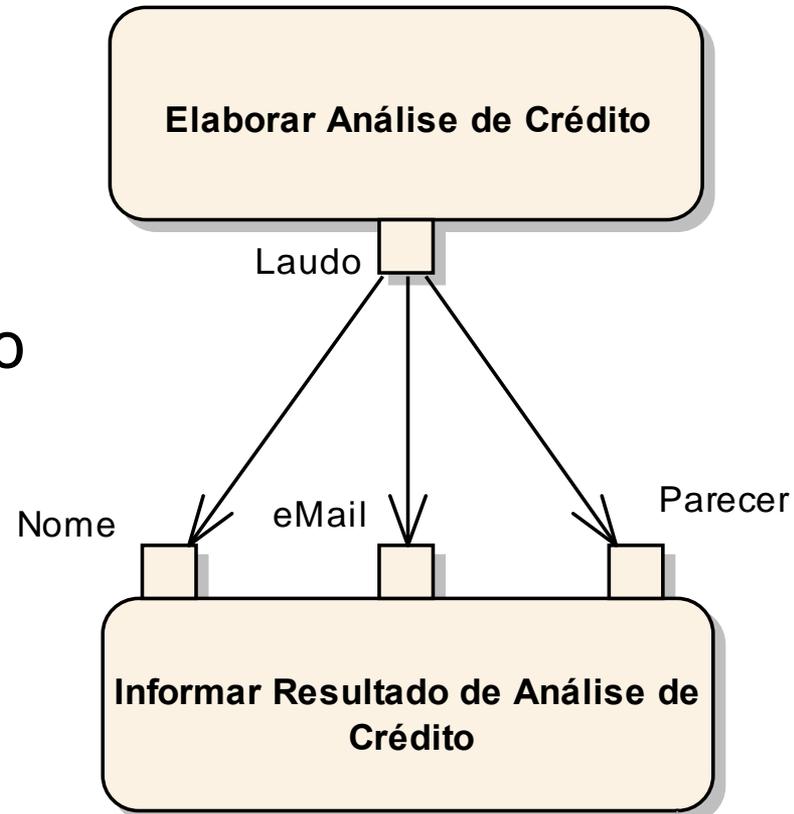


- Podemos usar a notação de pinos.



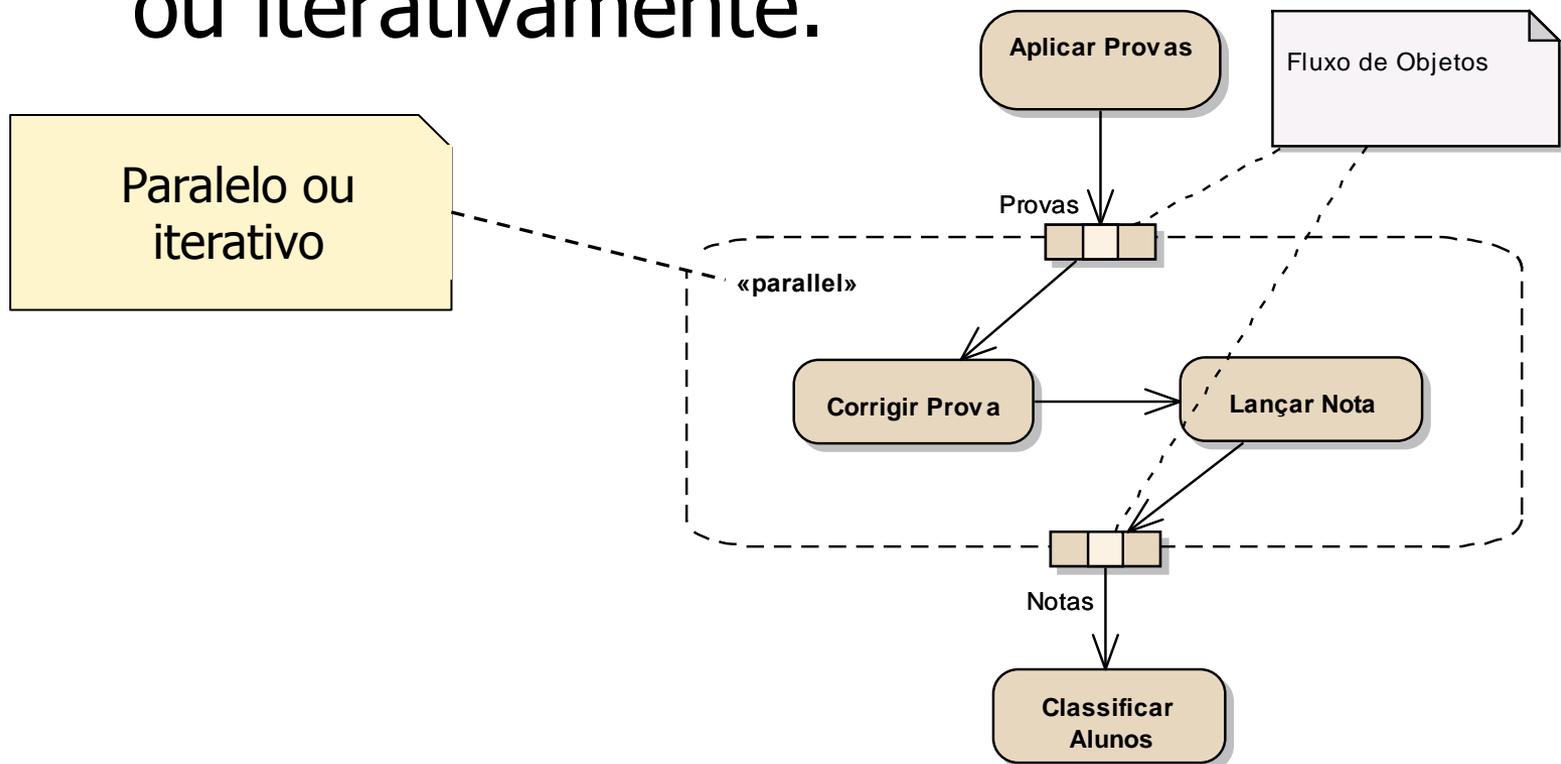
DA - Transformações

- Usadas quando o parâmetro de entrada de uma ação é o resultado de uma operação aplicada no parâmetro de saída da ação anterior na sequência.
- As transformações devem ser livres de efeitos colaterais (não podem alterar o estado do ambiente).



DA – Regiões de Expansão

- Usadas para especificar conjuntos de ações que são realizadas em paralelo ou iterativamente.





DA – Especificando UCs

- Problemas da especificação textual:
 - Não é concisa;
 - Dificilmente é completa;
 - Pouco manutenível.
 - Inserção, alteração e remoção de passos facilmente provoca inconsistências na especificação (a rastreabilidade);
 - A tecnologia tipicamente usada (tabelas do Word com referências cruzadas) é de baixa usabilidade.

Continua ...

DA – Especificando UCs

- Problemas da especificação textual (cont):
 - Precisam ser interpretadas e convertidas em diagramas pelos projetistas (é nessa etapa que a incompletude é evidenciada);
 - Demandam retrabalho (novas entrevistas).



DA – Especificando UCs

- Vantagens de uso:
 - Concisão;
 - Completude mais facilmente buscada;
 - Cenários podem ser facilmente identificados;
 - Maior manutenibilidade;
 - Especificação da colaboração Ator(es) Sistema feita visualmente, com o uso de ferramenta gráfica.

DA – Especificando UCs

- Desvantagem de uso:
 - Usuário em geral não é familiarizado com a notação dos DAs da UML.
- O caminho do meio:
 - Analista especifica em DA e depois converte para texto, para homologação pelo cliente.



DA – Especificando UCs

- Duas formas básicas de especificação com DAs:
 - Especificando unicamente as ações do sistema;
 - Vantagem: melhor estruturação do diagrama.
 - Desvantagem: as ações dos atores não são evidenciadas.
 - Incluindo também as ações dos atores na especificação.
 - Em raias distintas;
 - Usando cores distintas para as ações.

DA – Onde Mais Usar?

- Quando utilizar DAs (dicas do Fowler):
 - Descrevendo um algoritmo complicado;
 - Lidando com aplicações de processamento paralelo.
- Modelagem de negócios.