



Análise e Projeto Orientados a Objetos

Aula III – Concepção – Visão Geral do Sistema

Prof. Bruno E. G. Gomes
IFRN





- Fase de concepção do UP
 - ♦ Analista vai em busca das primeiras informações sobre o sistema
- Assume-se que o analista tem pouco conhecimento sobre o sistema
- Objetivo principal é descobrir se vale a pena fazer a análise
- A visão geral do sistema deve gerar, em poucos dias, documentação necessária para:
 - ♦ Entender o negócio e o problema proposto
 - ♦ Decidir se vale a pena continuar ou parar





Visão Geral do Sistema

- Informações sobre o negócio podem ser levantadas através de:
 - ♦ Entrevistas com os usuários e clientes
 - ♦ Exames de documentos, relatórios, sistemas e bibliografia
- Artefatos gerados não precisam ser necessariamente completos e organizados
 - ♦ Diagramas, atas de reunião com o cliente, relatórios, etc.
- Deve responder:
 - ♦ O que o cliente quer com o projeto (qual a visão da empresa)?
 - ♦ Comprar ou construir do zero?





Visão Geral do Sistema

- Um dos produtos desta fase é um documento de *visão geral do sistema* ou *sumário executivo*
 - ♦ Formato livre, no qual o analista descreve o que ele conseguiu descobrir de relevante sobre o sistema
- Descrição deve ser resumida em 1 ou 2 páginas de texto
 - ♦ Mais que isso, pode-se entrar em detalhes demais, que deve ser tratados em outros documentos posteriores
- O analista deve ter em mente que este documento descreve as principais preocupações do cliente
 - ♦ Nas fases posteriores, essas ideias serão melhor estruturadas





Exemplo – Sistema Livraria Virtual (Livir)

O sistema deve gerenciar todos os processos de uma livraria virtual, desde a aquisição até a venda de livros. O acesso dos compradores e gerentes deve ser feito através de um site *Web* e possivelmente com outras tecnologias. Os compradores fazem as transações pagando com cartão de crédito.

Existem promoções eventuais pelas quais os livros podem ser comprados com desconto.

De início, a livraria vai trabalhar apenas com livros novos a serem adquiridos de editoras que tenham sistema automatizado de aquisição. O sistema a ser desenvolvido deve conectar-se aos sistemas das editoras para efetuar as compras.

O sistema deve calcular o custo de entrega baseado no peso dos livros e na distância do ponto de entrega. Eventualmente pode haver promoções do tipo “entrega gratuita” para determinadas localidades.

O sistema deve permitir ao gerente emitir relatórios de livros mais vendidos e de compradores mais assíduos, bem como sugerir compras para compradores com base em seus interesses anteriores. Quando um livro é pedido, se existe em estoque, é entregue imediatamente, senão o livro é solicitado ao fornecedor, e um prazo compatível é informado.





Diagramas auxiliares

- É possível utilizar diagramas para melhor compreender o negócio da empresa ou partes importantes do sistema proposto
- Os diagramas são complementares à descrição geral do sistema
- Diagramas UML
 - ◆ Atividades
 - ◆ Estado





Modelagem de negócio - Diagrama de Atividades

- Modelam as atividades de negócio realizadas por um conjunto de pessoas, sistemas ou organizações
- Representam processos em nível organizacional
 - ♦ de forma ampla, em alto nível
 - ♦ não entra em detalhes funcionais
- Uso
 - ♦ Representar pessoas e/ou sistemas interagindo na execução de um processo
 - ♦ Ajudar o analista a entender quais são as atividades e os atores envolvidos nos principais processos de negócio da empresa





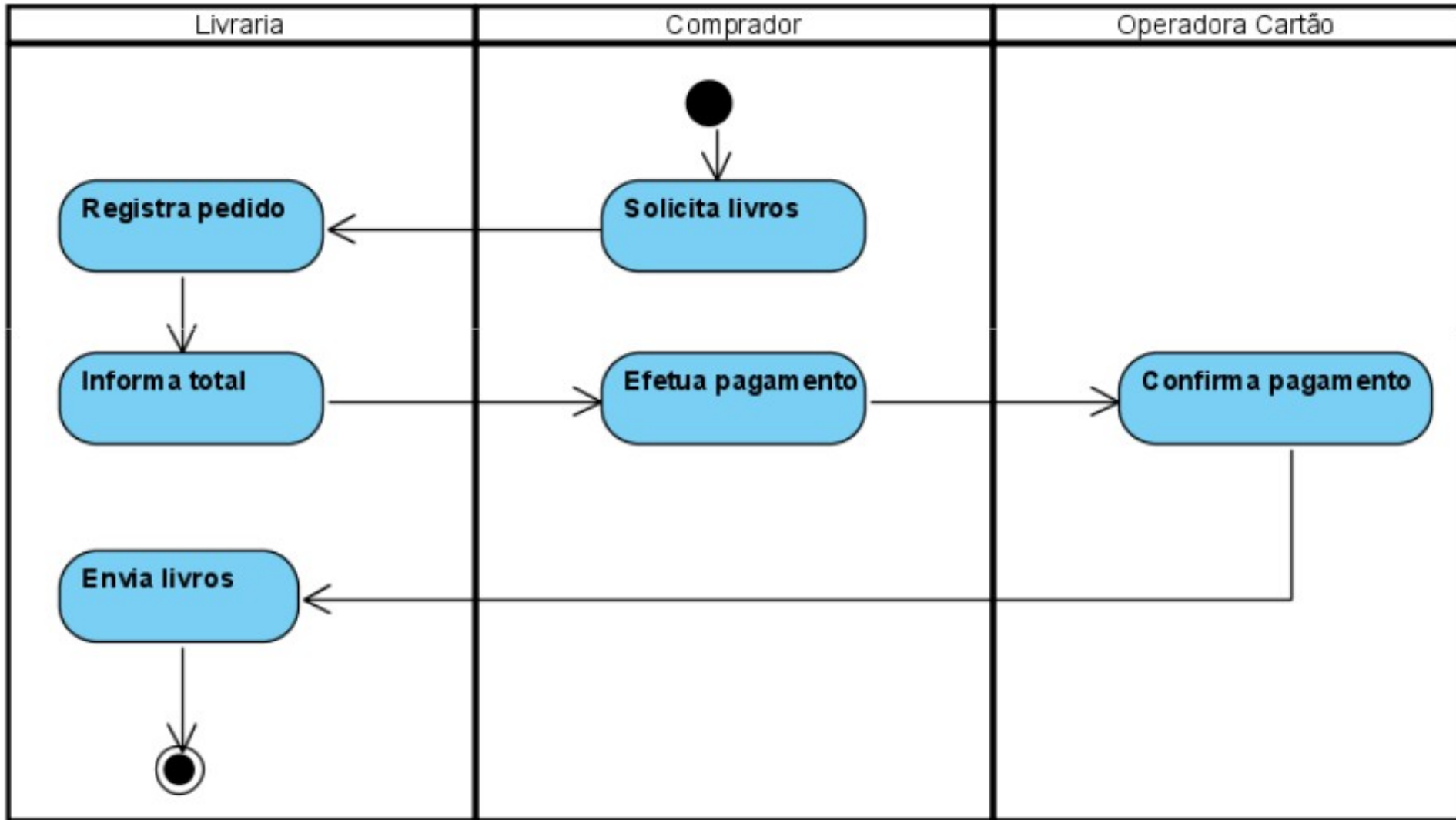
Diagrama de Atividades

- É dividido em raias (*swimlanes*)
 - ♦ Representa um ator um sistema que participa de um conjunto de atividades
- O *processo* deve ter uma pseudoatividade inicial (círculo preto) e uma pseudoatividade final (círculo preto dentro de outro círculo)
- *Atividades* são representadas por figuras retangulares dentro das raias
- *Fluxos* ou dependências entre atividades são representados por setas, indicando precedência entre elas
- Um *caminho* é uma sequência de atividades ligadas por fluxos





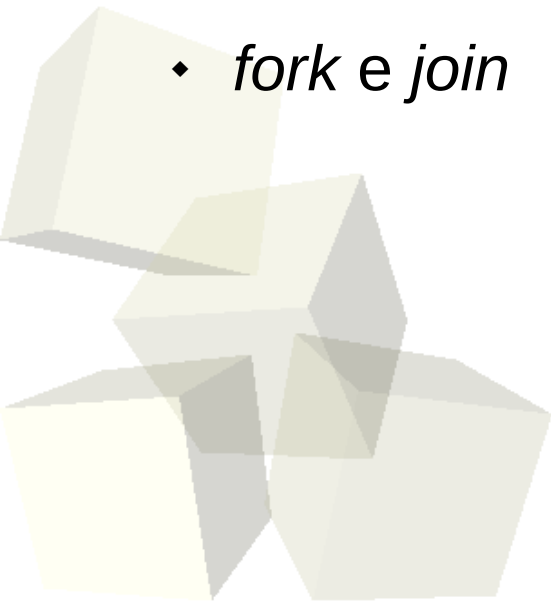
Processo de venda de livros – versão I





Estruturas de controle

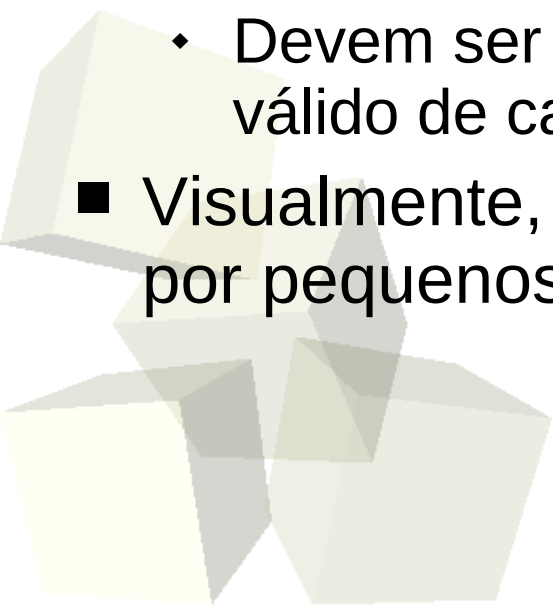
- No diagrama de atividades é possível usar estruturas de controle para seleção e paralelismo
- Seleção
 - ♦ *branch e merge*
- Paralelismo
 - ♦ *fork e join*





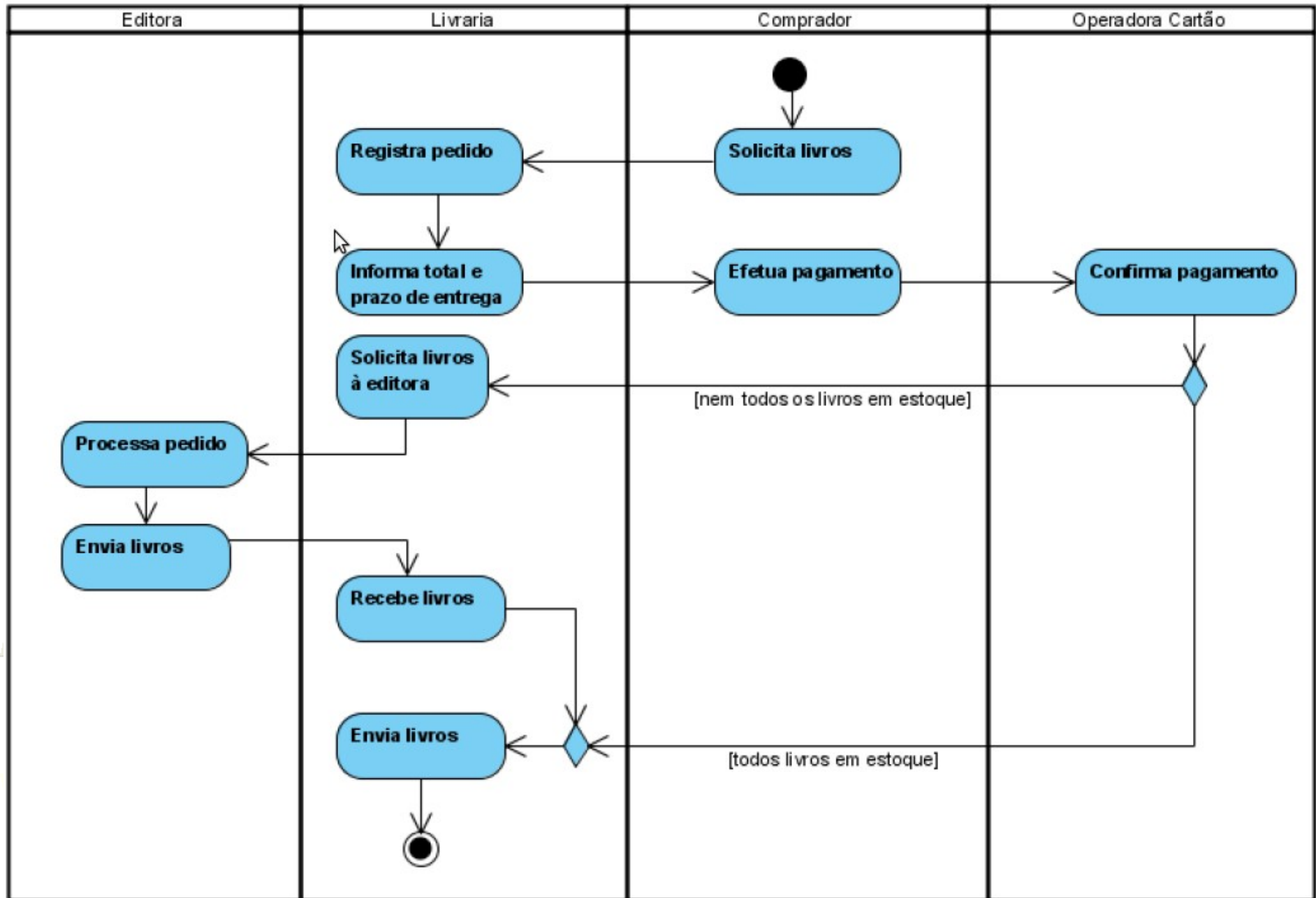
Estrutura de seleção *branch* e *merge*

- Representa *fluxos* alternativos no processo
- Do nó *branch* saem fluxos com condições de guarda (expressões lógicas entre colchetes)
- Todos os caminhos devem voltar a se encontrar em um nó *merge*
- Dois ou mais fluxos podem sair de uma estrutura de seleção
 - Devem ser mutuamente excludentes (apenas um é válido de cada vez)
- Visualmente, nós *branch* e *merge* são representados por pequenos losangos





Processo de venda de livros – versão II





Estrutura de paralelismo *fork* e *join*

- Execução de caminho em paralelo, ou seja, sem dependências entre suas atividades
- Visualmente, são representados como barras pretas
 - ♦ *fork* – inicia caminhos paralelos
 - ♦ *join* – finaliza (sincroniza) caminhos paralelos iniciados em um *fork*
- Depois do nó *join* o processo só pode prosseguir se todos caminhos que entram nele tiverem sido executados



Processo de venda de livros – versão III

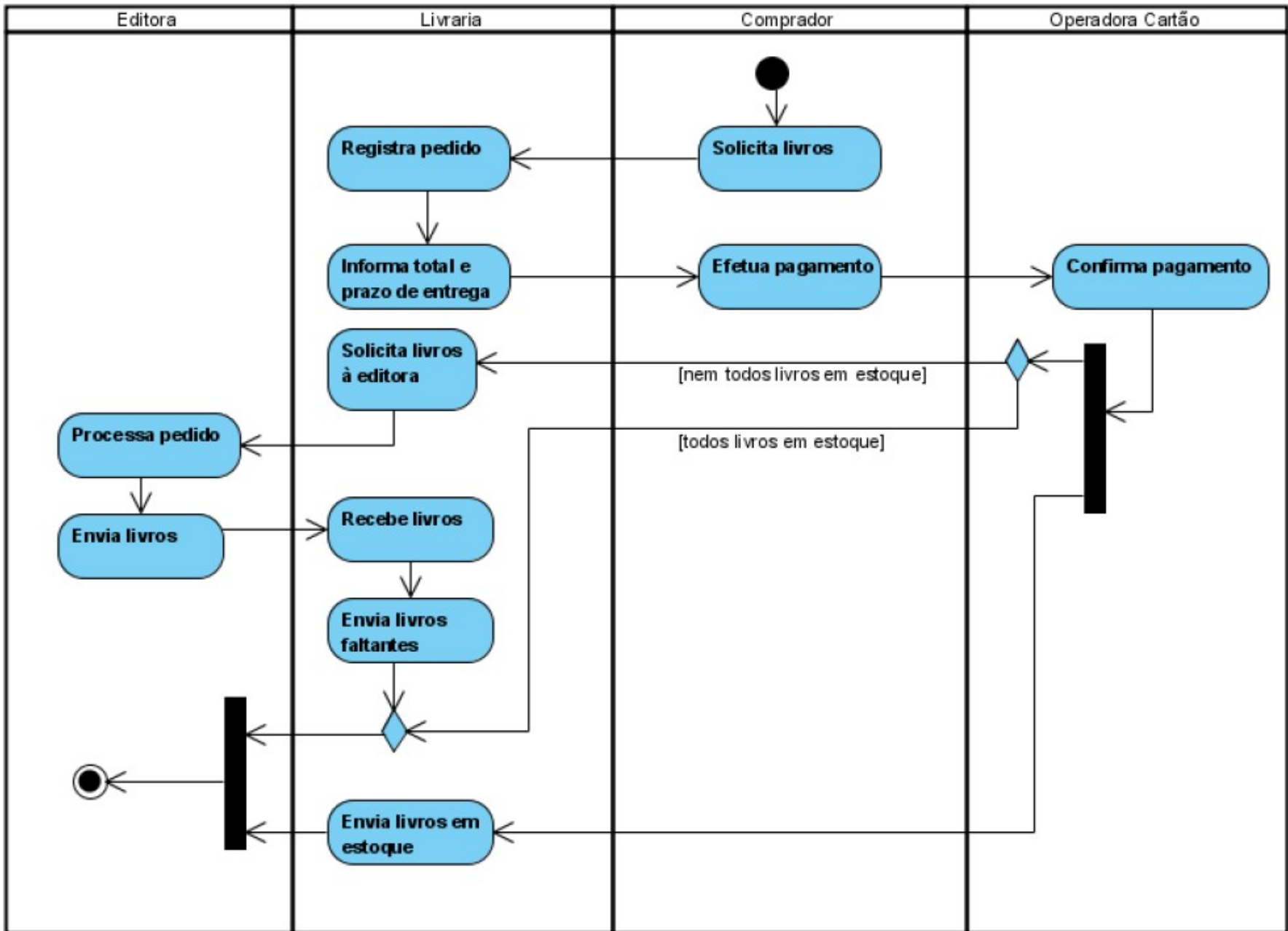




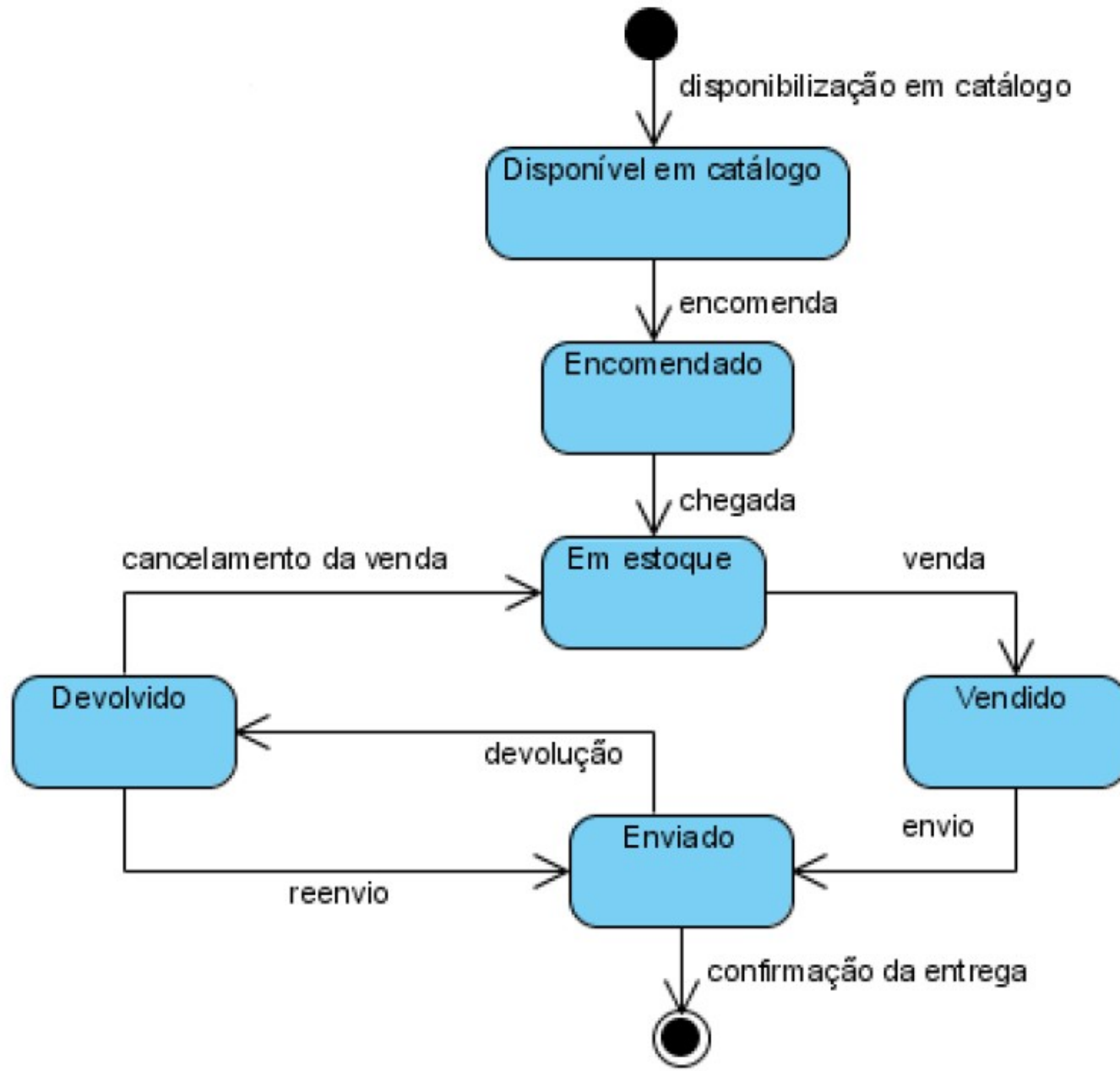
Diagrama de atividades - observações

- Diagrama sintaticamente correto
 - A cada nó *branch* deve corresponder um nó *merge*
 - A cada nó *fork* deve corresponder um nó *join*
 - Os nós *branch*, *merge*, *fork* e *join* devem ser perfeitamente aninhados
 - Só pode existir um nó inicial
 - Só pode existir um nó final
 - Cada *atividade* só pode ter um fluxo de entrada e um fluxo de saída (não vale para os nós, que não são atividades)
- Diagrama de atividades não é um modelo do sistema
 - É uma forma de ajudar a entender processos de negócio
 - Ajuda na captura de requisitos pelo analista – examina cada atividade e analisa se ela tem relação com o sistema a ser desenvolvido



- Apresenta estados de um sistema, ator ou entidade que se queira examinar em mais detalhe
- Uso
 - Compreender uma entidade que passa por vários estados
- Possui um estado inicial, mas pode ter mais de um estado final
- *Eventos* provocam a mudança de estado
 - Rotulados nos fluxos como expressões simples (ex.: login, encomenda, venda)
 - Fazem com que a entidade passe de um estado a outro
 - Podem ser restringidas por condições de guarda, representadas por condições expressões entre colchetes ao lado do evento (ex.: [senha correta] login)

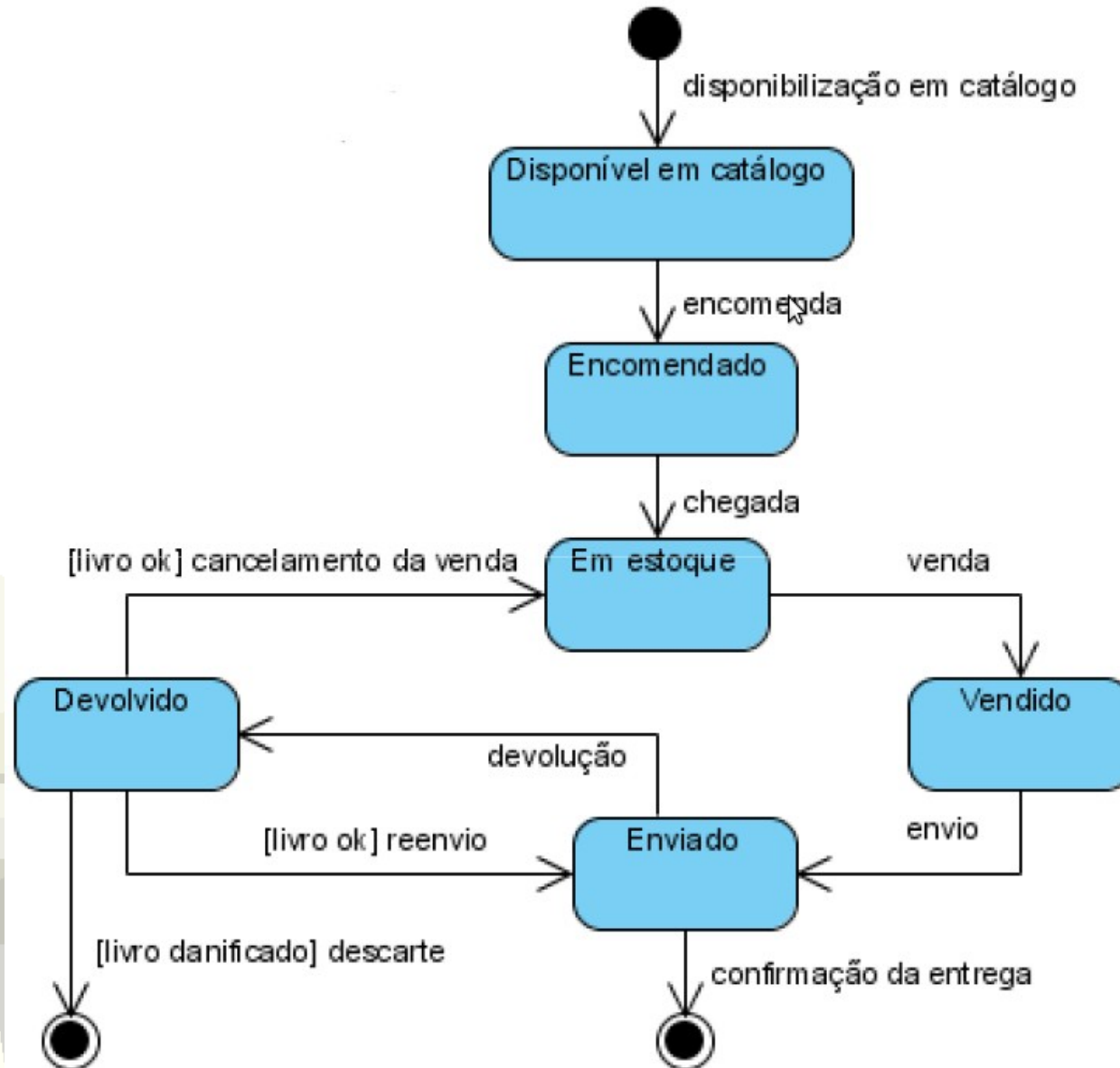
Máquina de estados – ciclo de vida de um livro, v.I





Ciclo de vida de um livro – v.II

Livro que retorna danificado – exemplo de uso de condições de guarda.



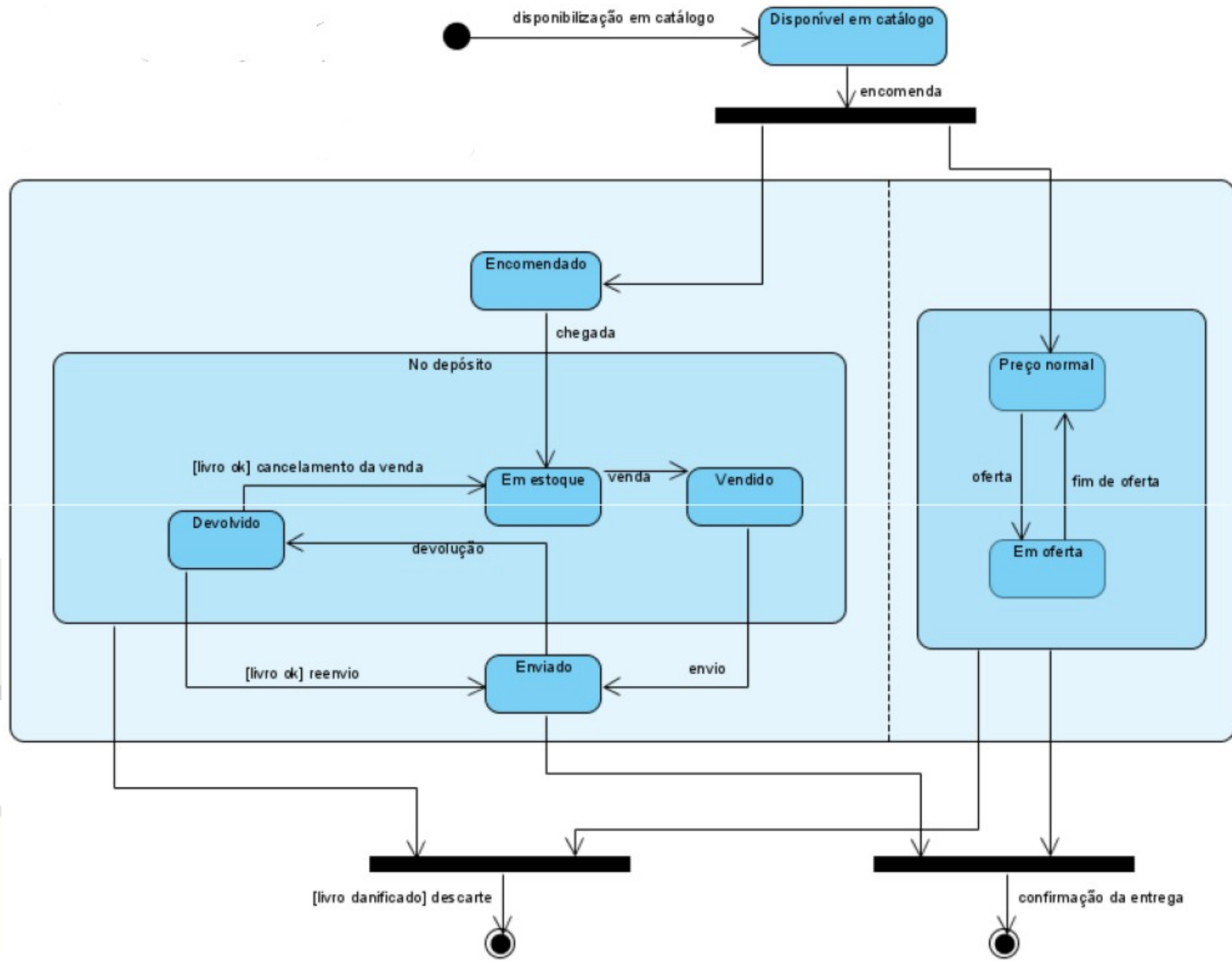


Máquina de estados – mais opções

- Evento que pode ocorrer em mais de um estado
 - ♦ Exemplo: um livro também pode ser danificado – e descartado – a partir dos estados “em estoque” e “vendido”
 - ♦ *Superestado*: conjunto de transições com o mesmo evento de/ou para o mesmo estado
 - ♦ Transição para um superestado corresponde ao conjunto de transições de cada um dos seus subestados
- Estados paralelos
 - ♦ Regiões concorrentes
 - ♦ Transição para dentro das regiões concorrentes é feita por um nó *fork*
 - ♦ Transições para fora das regiões concorrentes devem ser representadas por um nó *join*



Ciclo de vida de um livro – v. III





Considerações Finais

- Objetivo desta etapa é obter uma visão geral do sistema, e não uma especificação detalhada do seu funcionamento
 - ♦ Deve ser breve e objetiva
- Preocupação em descobrir informações sobre o sistema
 - ♦ O que o cliente quer? É viável prosseguir com o desenvolvimento do sistema?
- Para quais elemento do sistema devo fazer diagramas ?
 - ♦ Apenas de elementos chave para a sua compreensão
 - ♦ Exemplos
 - livraria – processo de venda, processo de entrega, livros, etc.
 - Hotel – hospedagens, processo de interação com fornecedores, etc.

