



Diagrama de Atividades Máquina de Estado

Herysson R. Figueiredo
herysson.figueiredo@ufn.edu.br



Sumário

- Diagrama de Atividade
 - Para que serve?
 - Elementos
 - Exemplo
- Diagrama
 - Para que serve?
 - Elementos
 - Exemplo

Diagrama de Atividade



Diagrama de Atividade

É um tipo de diagrama comportamental que representa graficamente o fluxo de controle de uma atividade para outra, com descrição de ações passo-a-passo em um sistema.



Diagrama de Atividade

Especifica a transformação de entradas em saídas por meio de uma **sequência** controlada temporal de ações. Semelhante a um fluxograma, porém com suporte a concorrência (paralelismo) e sincronismo de atividades.



Diagrama de Atividade

Atividade é um processo de negócio, como por exemplo a venda de livros online. Muitas vezes descreve a implementação de um caso de uso.



Para que serve um diagrama de atividade

- Descrever como um processo começa, evolui, bifurca, sincroniza e termina.
- Evidenciar regras de negócio, condições (guards) e exceções no fluxo.
- Comunicar responsabilidades (p. ex., por meio de *swimlanes*).
- Preparar terreno para análise, melhorias e automação de processos.



Para que serve um diagrama de atividade

Seu emprego mais comum é na captura de trabalhos que vão ser executados quando uma operação específica do sistema é disparada (ação)

Muito úteis para modelagem de fluxo de trabalho e processos.



Quando usar

- Para explicar um processo de negócio (ex.: matrícula, compra, manutenção).
- Para detalhar o comportamento de um caso de uso ou de uma operação.
- Para alinhar equipe sobre regras, caminhos alternativos e concorrência.



Elementos de um diagrama de atividades

Ação é um passo individual (atômico) dentro de uma atividade, como por exemplo pesquisar livros.



Ação



Elementos de um diagrama de atividades

Nó inicial: Ponto de início da atividade modelada

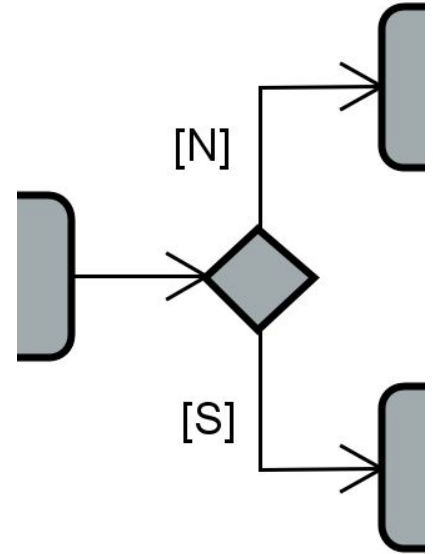


Fluxo / Aresta: Ou ainda transição. Descreve a sequência na qual as atividades se realizam. Conexões entre duas ações. Representado por uma seta.



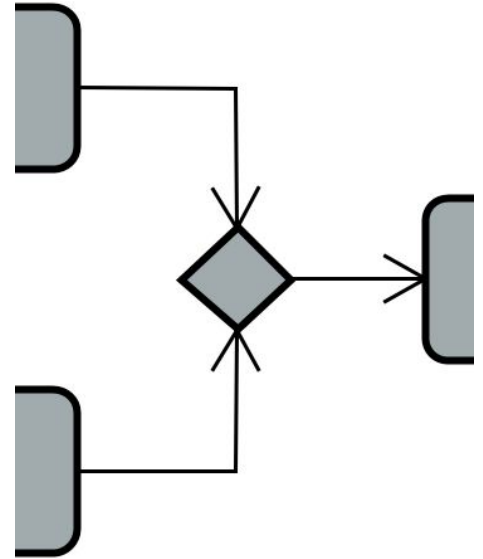
Elementos de um diagrama de atividades

Decisão: Um único fluxo de entrada e vários fluxos de saída. Cada fluxo de saída possui uma sentinela ou guarda, que é condição booleana, entre colchetes. Sentinelas são mutuamente exclusivas.



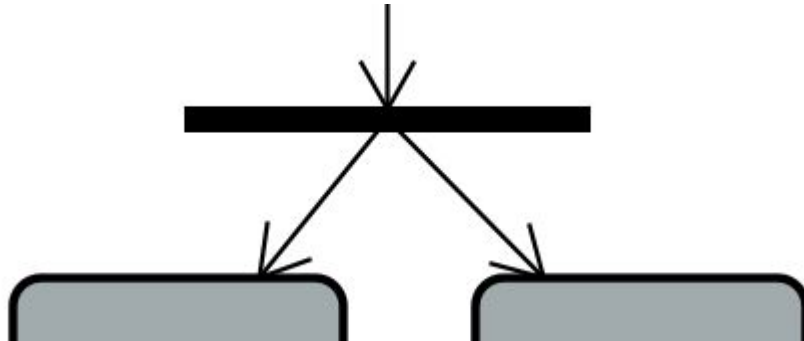
Elementos de um diagrama de atividades

Intercalação: Vários fluxos de entrada e uma única saída. Marca o final de um condicional iniciado por uma decisão.



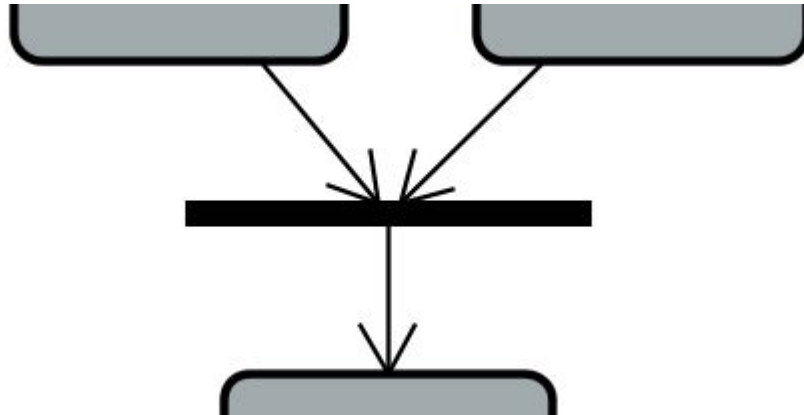
Elementos de um diagrama de atividades

Divergência: (*Fork*) ponto no qual duas atividades ou mais tarefas podem se iniciar em paralelo.



Elementos de um diagrama de atividades

Convergência: (*Join*) ponto no qual duas atividades ou mais tarefas paralelas se unem para dar início a uma nova tarefa.





Elementos de um diagrama de atividades

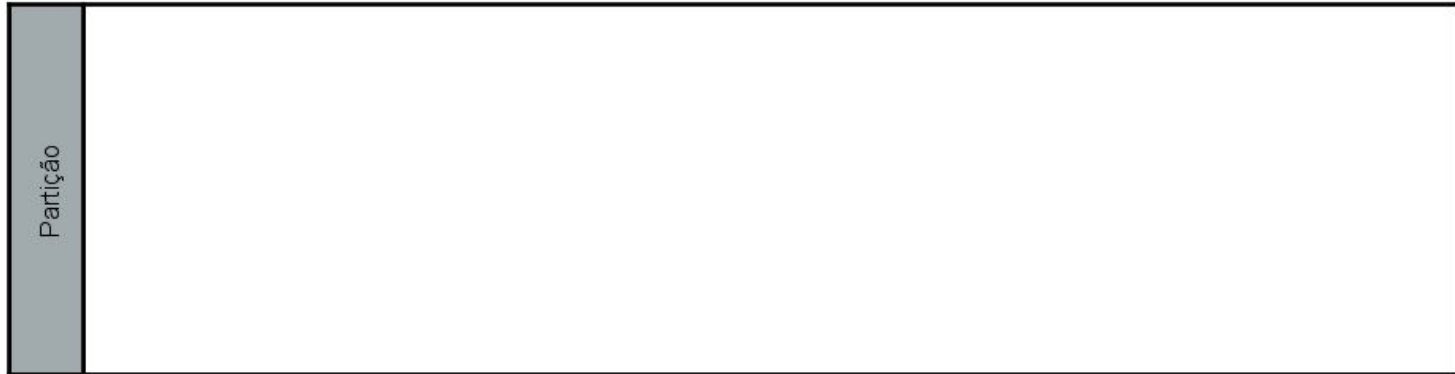
Nó Final de Atividade: Ponto onde termina a atividade modelada.





Elementos de um diagrama de atividades

Partições: Mostra quem faz o que (quem realiza cada ação ou conjunto de ações). Na UML 1.1 eram chamadas de “raias” (*swimlanes*).





Elementos de um diagrama de atividades

Final de Fluxo: Uma parte dos fluxos da atividade terminaram, mas a atividade ainda prossegue.





Elementos de um diagrama de atividades

Para o diagrama estar sintaticamente correto, deve-se observar que:

- a cada nó branch deve corresponder um nó merge quando possível;
- a cada nó fork deve obrigatoriamente corresponder um nó join;
- os nós fork e join devem estar perfeitamente aninhados (ou seja, um branch não pode terminar com join e um fork não pode terminar com merge nem podem estar entrelaçados);



Elementos de um diagrama de atividades

Para o diagrama estar sintaticamente correto, deve-se observar que:

- só pode existir um nó inicial;
- só pode existir um nó final;
- Cada atividade só pode ter um único fluxo de entrada e um único fluxo de saída (isso não vale para os nós join, fork, merge e branch, que não são atividades).

Os nós fork, join, branch e merge podem estar em qualquer uma das raíais, pois seu significado não é afetado por elas.



Prática 1

Bolo comum

- Leve à batedeira os ovos, manteiga, açúcar, bata até formar um creme.
- Acrescente o leite e a farinha de trigo e torne a bater até obter uma massa homogênea, reserve.
- Ligue o forno em alta temperatura, unte a forma com manteiga.
- Despeje na massa reservada o pó royal e misture com uma espátula levemente sem bater a massa, depois de tudo misturado despeje a massa na forma untada e leve ao forno.

Obs.: Depois de colocar o bolo, não abra o forno nos primeiros 20 minutos. Se quiser pode acrescentar 1 xícara de chocolate em pó antes de colocar o pó royal, o bolo se tornará um maravilhoso bolo de chocolate.



Prática 2

Receita de bolo simples:

- Bata as claras em neve e reserve.
- Misture as gemas, a margarina e o açúcar até obter uma massa homogênea.
- Acrescente o leite e a farinha de trigo aos poucos, sem parar de bater.
- Por último, adicione as claras em neve e o fermento.
- Despeje a massa em uma forma grande de furo central untada e enfarinhada.
- Asse em forno médio 180 °C, preaquecido, por aproximadamente 40 minutos ou ao furar o bolo com um garfo, este saia limpo.



Visão geral de um sistema

Sistema Livir: Livraria Virtual

O sistema deve gerenciar todos os processos de uma livraria virtual, desde a aquisição até a venda dos 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.



Visão geral de um sistema

Sistema Livir: Livraria Virtual

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 a um gerente emitir relatórios de livros mais vendidos e de compradores mais assíduos, bem como sugerir compras para compradores baseadas 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 ao comprador.



Prática

Modele o diagrama de atividade para o processo de venda de livros.

Diagrama de Máquina de Estado



Diagrama de Máquina de Estado

Também conhecido como **Diagrama de Transição de Estados** ou simplesmente **Diagrama de Estados**.

É um diagrama comportamental empregado para descrever como um sistema se comporta quando um evento ocorre, considerando todos os estados, transições e ações possíveis de um objeto.



Diagrama de Máquina de Estado

Representa o estado ou situação na qual um objeto pode se encontrar ao longo da execução dos processos em um sistema. Mostra como um elemento se comporta por meio de um conjunto de transições de estado, a chamada "**máquina de estados**".



Para que serve?

Representar o ciclo de vida de um objeto

Mostra os estados possíveis de um objeto (ex: um pedido, um usuário, uma ordem de serviço) e como ele transita entre esses estados ao longo do tempo.

Exemplo: Um pedido pode estar nos estados: Criado → Confirmado → Enviado → Entregue → Finalizado.



Para que serve?

Modelar sistemas reativos

É ideal para sistemas que respondem a eventos, como sistemas embarcados, interfaces interativas ou controladores de processos.

Exemplo: Um semáforo inteligente que muda de estado com base em sensores ou tempo.



Para que serve?

Ajudar na detecção de falhas lógicas

Permite visualizar transições inválidas, lacunas de tratamento de eventos, ou estados que não têm saída, tornando mais fácil depurar o sistema.



Para que serve?

Documentar regras de negócio

Facilita a comunicação entre desenvolvedores, analistas e stakeholders, descrevendo regras como:

- O que acontece quando o usuário clica em "Cancelar"?
- Quais eventos levam ao encerramento de uma sessão?



Para que serve?

Complementar outros diagramas UML

É um diagrama comportamental, que complementa diagramas estruturais como:

- Diagrama de Classes (estrutura estática)
- Diagrama de Casos de Uso (interações com atores)
- Diagrama de Atividades (fluxo geral)



Estado (simples)

Condição ou situação na vida de um objeto que satisfaz alguma condição, realiza alguma atividade ou espera um evento.



Estado



Estado inicial / Estado final

Estado inicial: Determina o início da modelagem dos estados de um elemento.



Estado final: Indica o final dos estados modelados para o elemento.





Elementos de um diagrama de estados

Transição: Movimento de um estado para outro estado. Representa um evento que causa uma mudança no estado de um objeto, levando a um novo estado.





Elementos de um diagrama de estados

Transição:

Ocorre da seguinte forma:

1. Um elemento está em um estado inicial;
2. Um evento ocorre;
3. Uma ação é realizada;
4. O elemento muda para um estado distinto.

Descrição





Elementos de um diagrama de estados

Evento: Incidente que leva os objetos a transacionar de um estado para outro. Ocorrência de um estímulo que pode disparar uma transição de estados. Pode ser interno ou externo.

Tipo de eventos: Sinal, Chamada, Temporização, Mudança



Elementos de um diagrama de estados

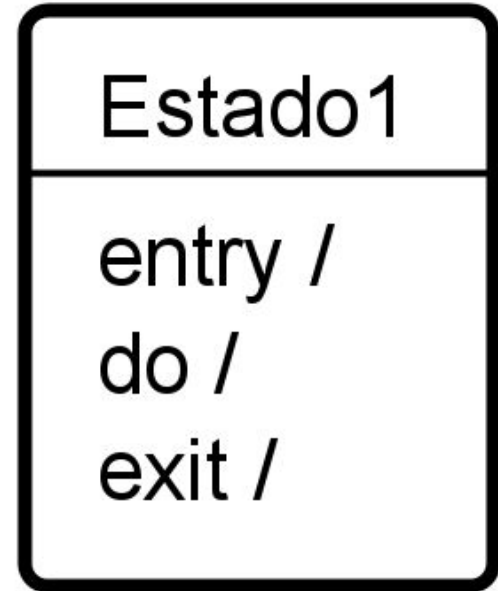
Ação: Execução atômica que se completa sem interrupção, resultando em uma alteração de estado.

Atividade: Execução atual não atômica em uma máquina de estados.



Estado - Comportamento interno

O comportamento interno de um estado representa ações que ocorrem enquanto o objeto está naquele estado específico, independentemente de transições externas.



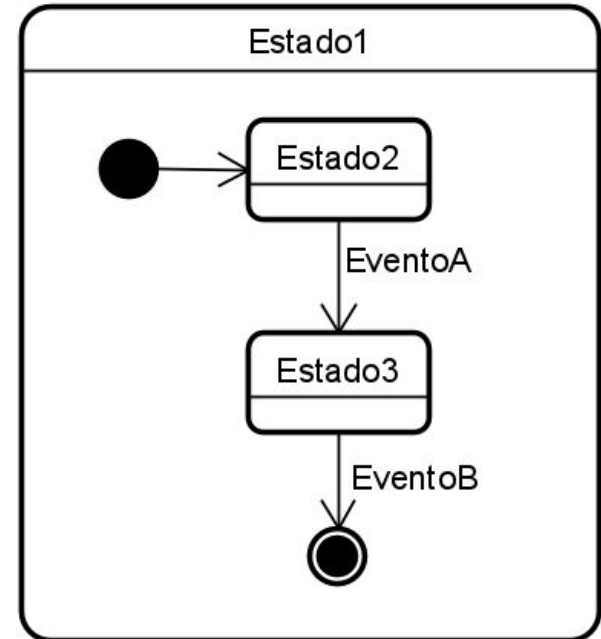


Comportamentos de Estado: Entry / Do / Exit.

Comando	Quando é executado?	Exemplo
entry	Ao entrar no estado	entry / exibirMensagemBoasVindas()
do	Enquanto o estado está ativo	do / monitorarSessao()
exit	Ao sair do estado	exit / limparVariaveis()

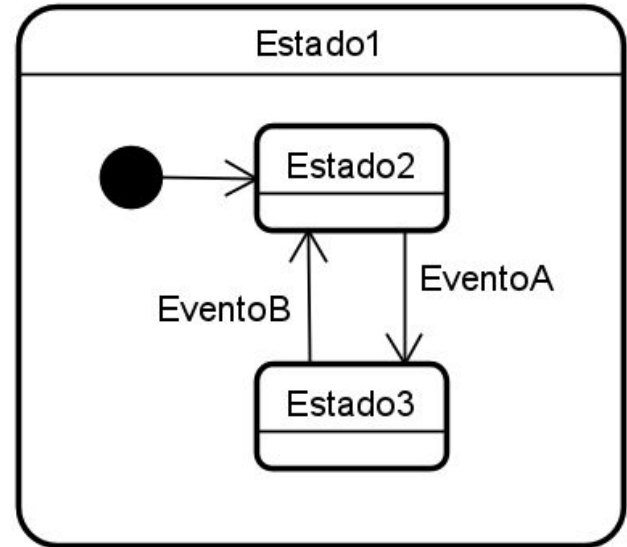
Elementos de um diagrama de estados

Estado composto: Estado que possui sub-estado.



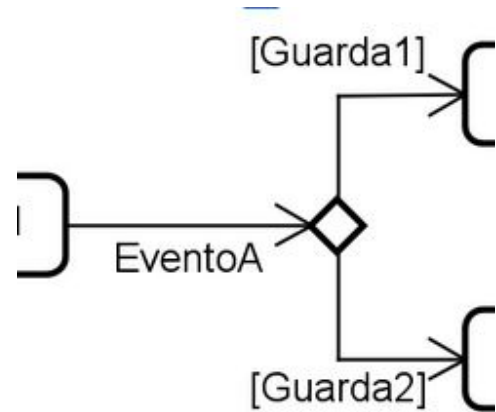
Elementos de um diagrama de estados

Estado composto: Estado que possui sub-estado.



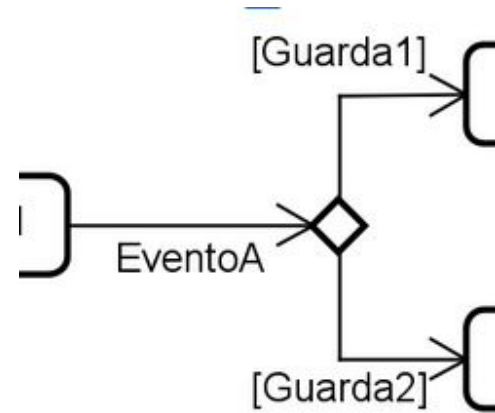
Elementos de um diagrama de estados

Pseudo Estado de escolha: Ponto na transição de estado de um objeto no qual uma decisão será tomada, baseada em uma condição. É um nó de decisão, condicionado por **condições de guarda**, para decidir qual o próximo estado a ser gerado para o objeto.



Elementos de um diagrama de estados

Condições de guarda: Condição avaliada após o disparo de um evento, que determina como ocorrerá a transição (pode haver múltiplas transições possíveis do mesmo estado com o mesmo disparo) mas só uma ocorrerá.





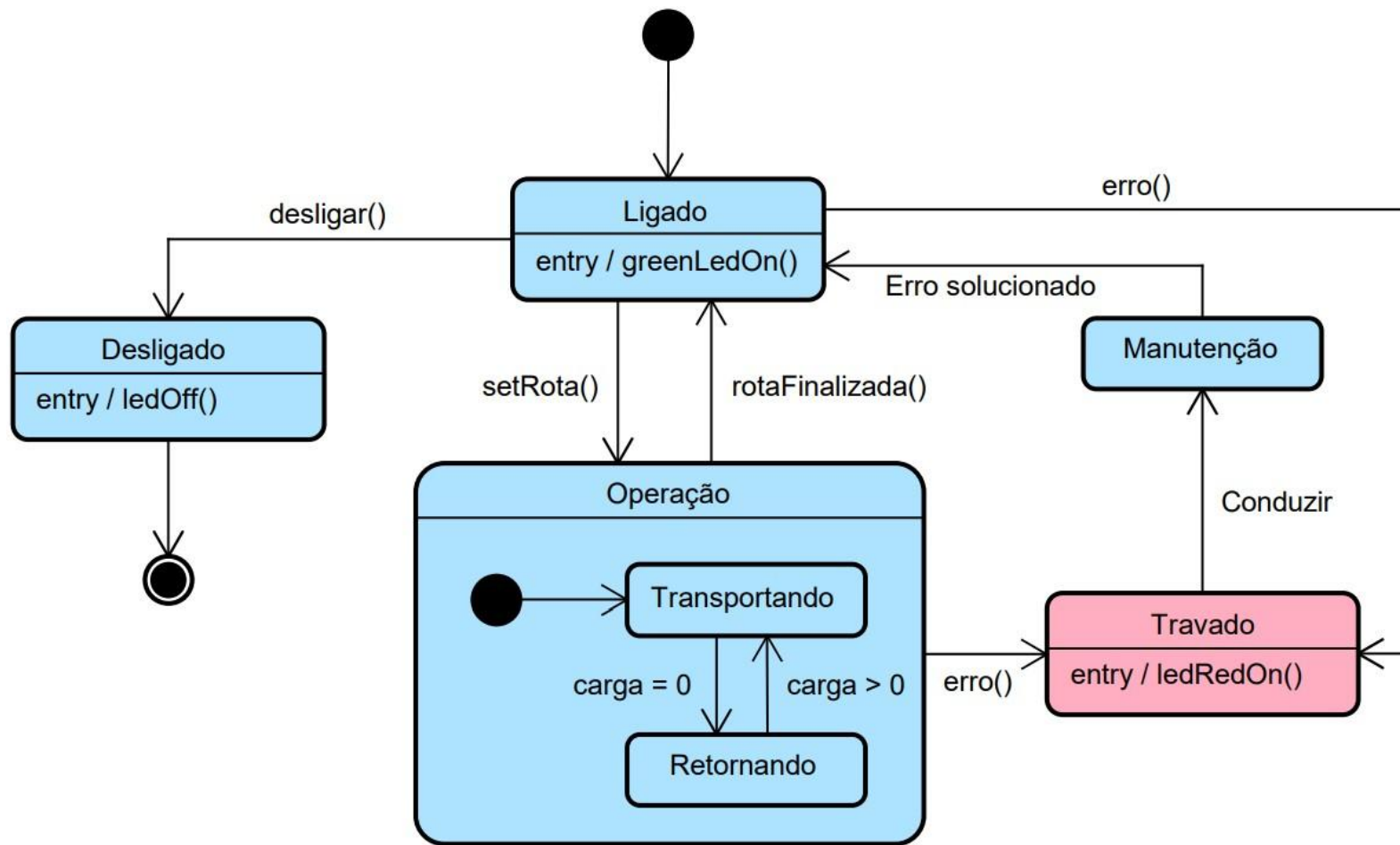
Como criar um diagrama de estados

- Determinando o estado inicial e o estado final
- Identificando todos os estados possíveis para o processo modelado
- Use setas ou linhas para destacar as transições de controle de um estado para outro, conectando origem e destino
- Rotule os eventos que disparam essas transições



Como criar um diagrama de estados

- Estabeleça condições de guarda para assegurar que as transições são apropriadas e relevantes. Uma condição de guarda força a verificação da transição contra uma condição antes de prosseguir.



Atividade Prática

Criar um diagrama de estados para o personagem principal de um jogo.





Prática 1

Pedido de e-commerce

Desafio: Modele os estados de um pedido do momento da criação até o desfecho.

Sugestão de estados: criado, pagamento, separação, expedição, entregue, cancelado.

Eventos possíveis: aprovação/reprovação do pagamento, falhas na entrega, tentativa novamente.



Prática 2

Drone de entrega

Desafio: Modele o ciclo de missão de um drone: antes, durante e após o voo.

Estados mínimos (opcional usar): aguardando missão, decolando, em rota, retorno, aterrissando, falha.

Eventos possíveis: missão atribuída, GPS ok, baixa bateria, vento forte, perda de sinal, pouso ok.



Bibliografia

BEZERRA, Eduardo. Princípios de análise e projeto de sistemas com UML. 2. ed. rev. atual. Rio de Janeiro, RJ: Campus, 2007. 369 p.

WAZLAWICK, Raul Sidnei. Análise e projeto de sistemas de informação orientados a objetos. 2. ed. rev. e atual. Rio de Janeiro: Elsevier, 2011. 330 p. (Série SBC, Sociedade Brasileira de computação) ISBN 978-85-352-3916-4

SOMMERVILLE, Ian. Engenharia de software, 10ª ed., 2019. (Biblioteca Digital)

BOOCH, Grady; RUMBAUGH, James; JACOBSON, Ivar. UML: guia do usuário. 2. ed. Rio de Janeiro, RJ: Ed. Campus, 2006. 474 p.

LARMAN, Craig. Utilizando UML e padrões: uma introdução à análise e ao projeto orientados a objetos e ao desenvolvimento iterativo. 3. ed., reimpr. Porto Alegre: Bookman, 2007. 695 p. ISBN 978-85-60031-52-8

SOFTWARE GUIDES. **State Machine Diagrams - UML 2 Tutorial**. UML-Diagrams.org, [20--]. Disponível em: <https://www.uml-diagrams.org/state-machine-diagrams.html>.