



INSTITUTO DE CIÊNCIAS EXATAS

DEPARTAMENTO DE COMPUTAÇÃO

# UML – Diagrama de Transição de Estados (DTE)

Eduardo Kinder Almentero  
[ekalmentero@gmail.com](mailto:ekalmentero@gmail.com)

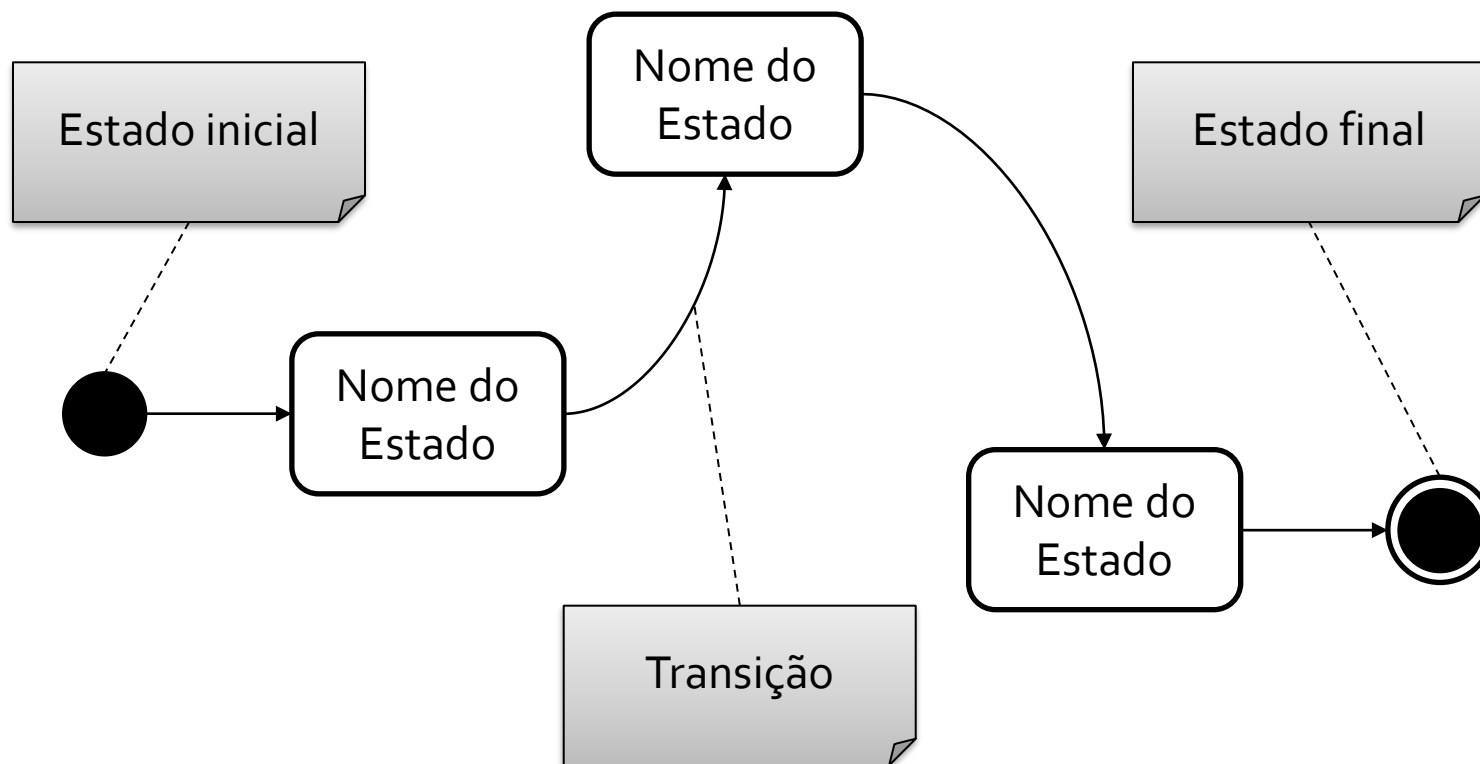
# Introdução

- É um diagrama comportamental
  - Enfatizam o que deve acontecer no sistema que está sendo modelado.
  - Como ilustram o comportamento de um software, são muito utilizados para descrever as funcionalidades, ou comportamentos específicos .

# Introdução

- O diagrama mais antigo que faz parte da UML
  - Foi criado nos anos 60;
- Ferramenta útil para mostrar o **ciclo de vida de um objeto**;
- Em um sistema real, **somente poucas classes** demandam o uso de um DTE.
  - Classes críticas;
  - Classes que se comportam de forma diferente em função de transições de estados;
  - Estados relevantes para compreensão do funcionamento do sistema;

# Notação Básica



# Estados

- Os estados **são momentos no ciclo de vida de um objeto;**
  - São determinados pelo valor que um ou mais de seus atributos assume.
- Esses estados determinam **que ações podem ser feitas sobre o objeto;**
- Exemplo
  - Um **ventilador simples** pode ter os estados **desligado e ligado;**
  - Um **ventilador mais elaborado** pode ter os estados **desligado, ventilando fraco, ventilando médio e ventilando forte.**

# Notação Gráfica: Estado Inicial e Final

- Estado inicial
  - Ponto de início do diagrama
  - Não aceita transições de entrada



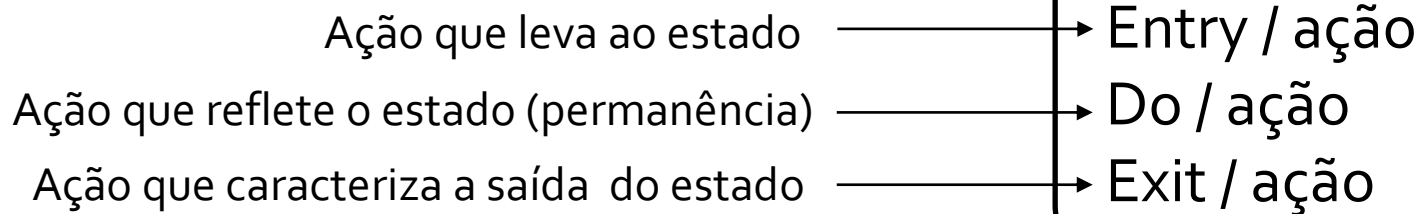
- Estado final
  - Ponto de término do diagrama
  - Não aceita transições de saída



# Notação Gráfica: Estados

Um estado caracteriza um determinado instante (momento) do ciclo de vida do objeto

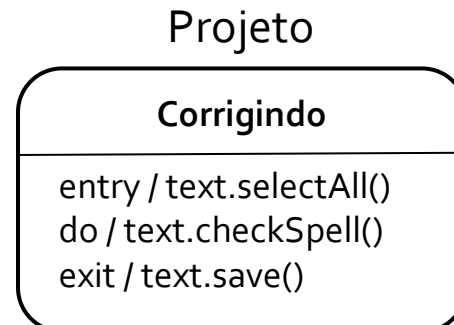
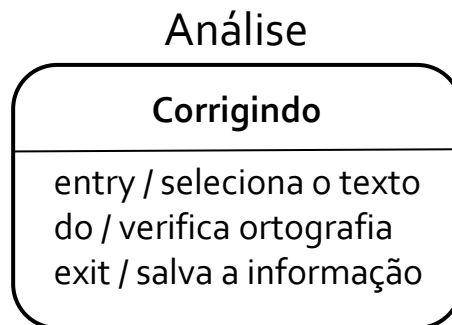
É comum suprimir as informações internas do estado quando estão claras a partir do seu nome e transições.



É usual que o nome do estado seja um substantivo ou um verbo (gerúndio) acompanhado de um substantivo.

# Efeitos

- É possível abordar os efeitos do a partir **da perspectiva de requisitos (análise) e técnica (projeto)**
- Em análise (requisitos)
  - Os efeitos de entrada, execução e de saída são descritos em linguagem natural
- Em projeto (aspectos técnicos considerados)
  - Os efeitos de entrada, execução e de saída são transformados em operações



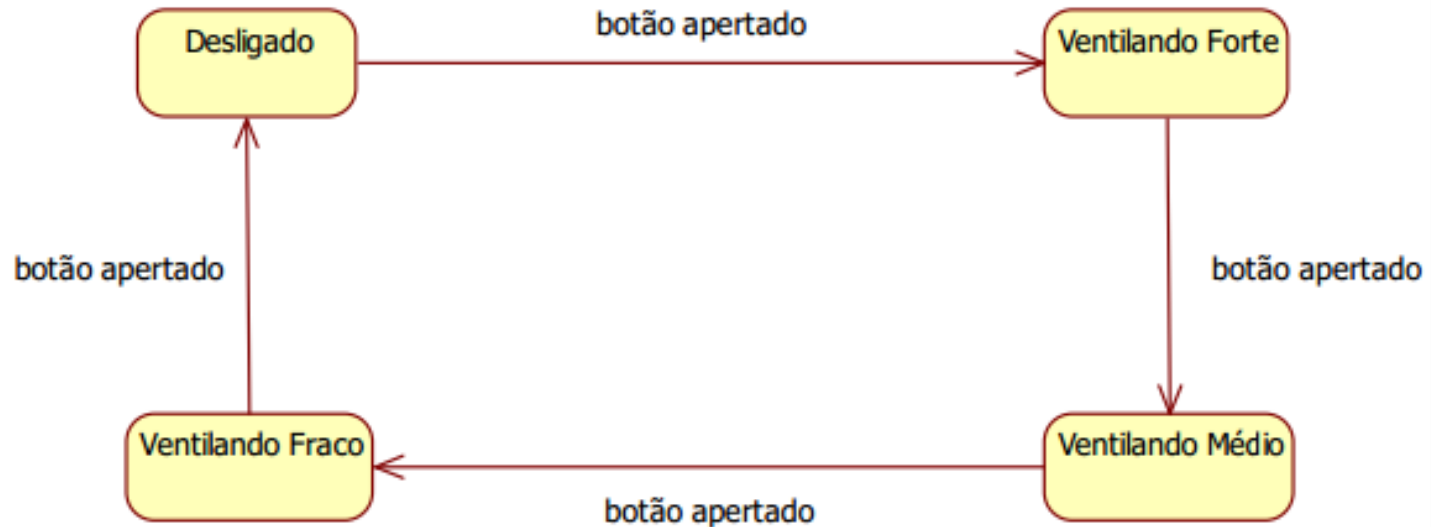


# Transições

- As transições determinam a **troca de estados em função de um determinado evento**;
- Exemplo
  - A partir do estado desligado, caso o botão seja apertado, o ventilador vai para o estado ventilando forte
  - A partir do estado ventilando forte, caso o botão seja apertado, o ventilador vai para o estado ventilando médio
  - A partir do estado ventilando médio, caso o botão seja apertado, o ventilador vai para o estado ventilando fraco
  - A partir do estado ventilando fraco, caso o botão seja apertado, o ventilador vai para o estado desligado

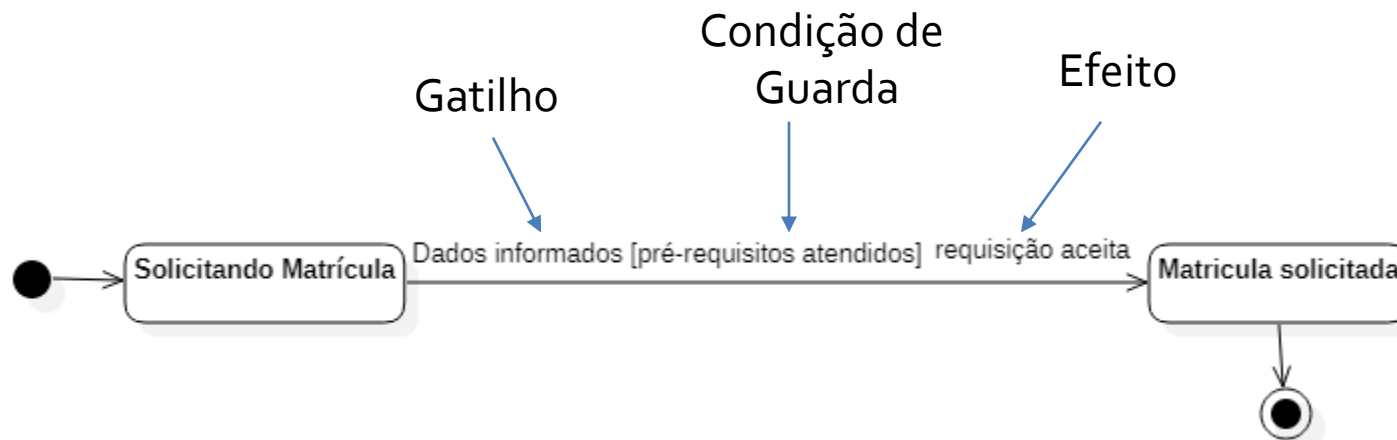
# Transições

- Exemplo simples: ventilador



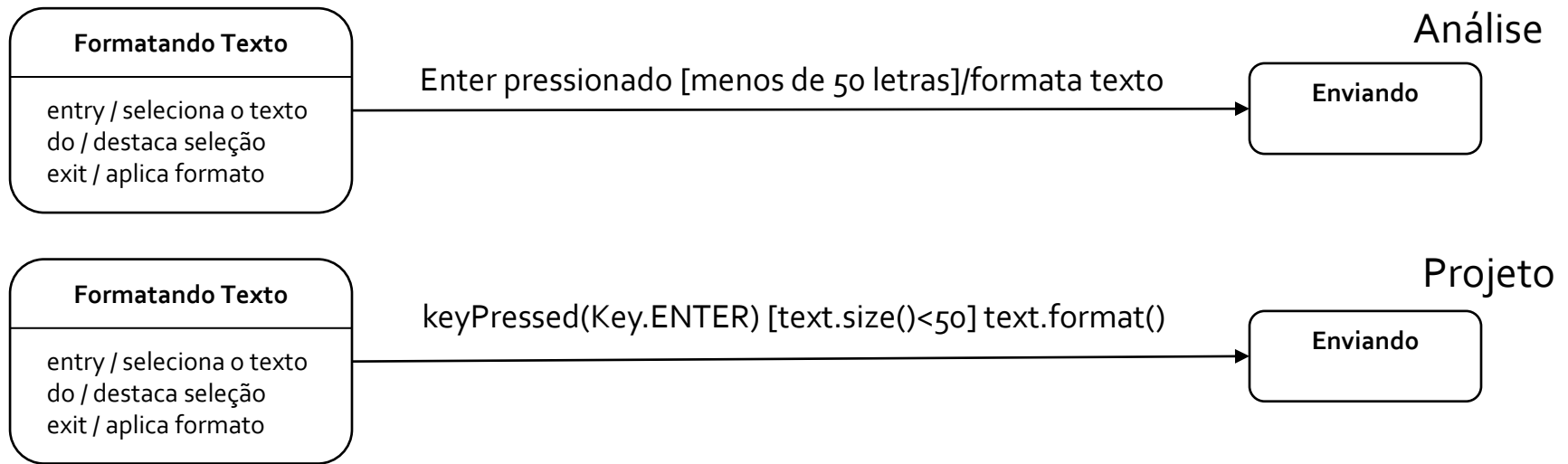
# Transições

- As transições são representadas por linhas, que podem conter (versão completa):
  - Gatilho (*trigger*)
  - Condição de guarda (*guard*)
  - Efeito (*effect*)
- Sintaxe: GATILHO [ CONDIÇÃO ] / EFEITO



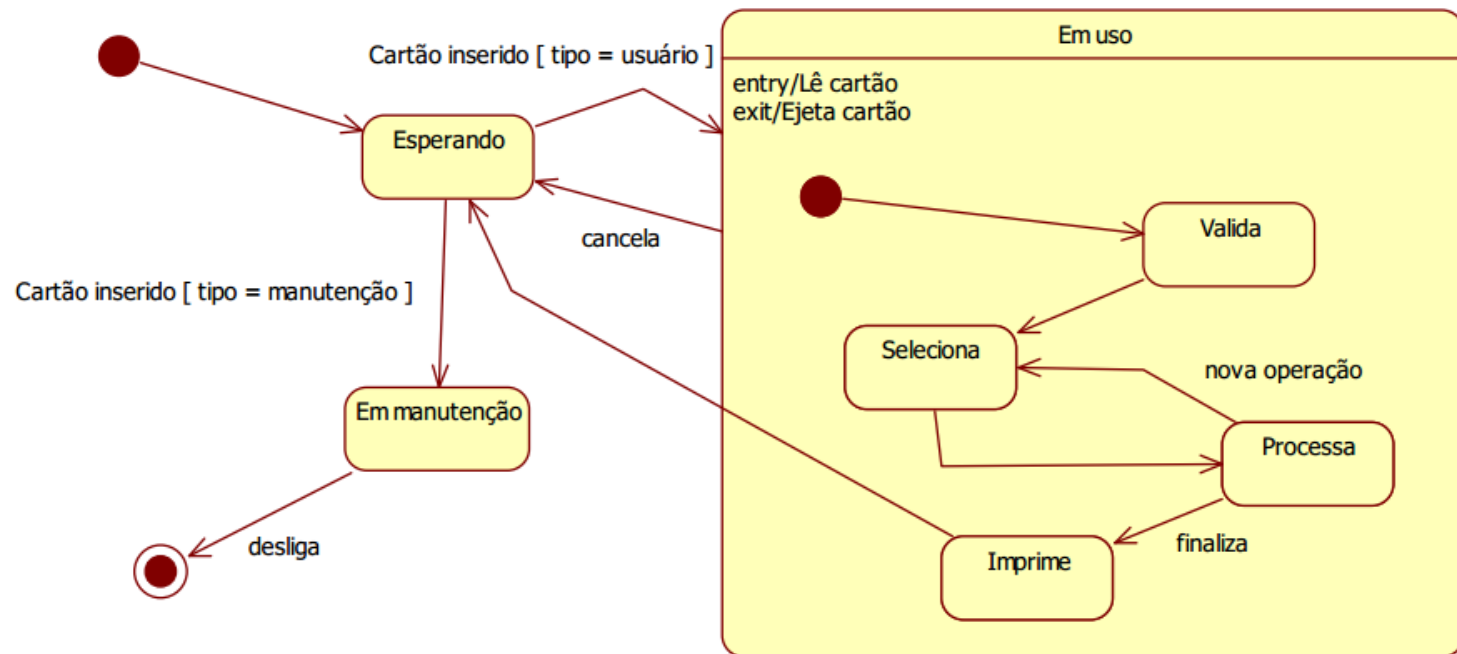
# Transições

- Em análise
  - Utilização de linguagem natural para gatilhos, condição de guarda e efeitos
- Em projeto
  - Gatilhos transformados em eventos
  - Condições de guarda transformadas em expressões booleanas
  - Efeitos transformados em operações



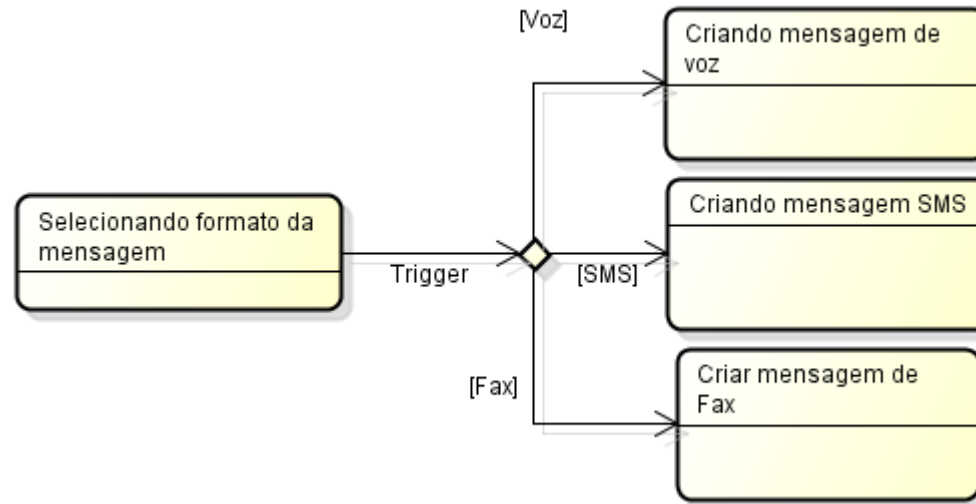
# Estados compostos

- Para viabilizar uma melhor organização do diagrama, podem ser criados estados compostos
- Estados compostos permitem a descrição de um diagrama interno de transição de estados



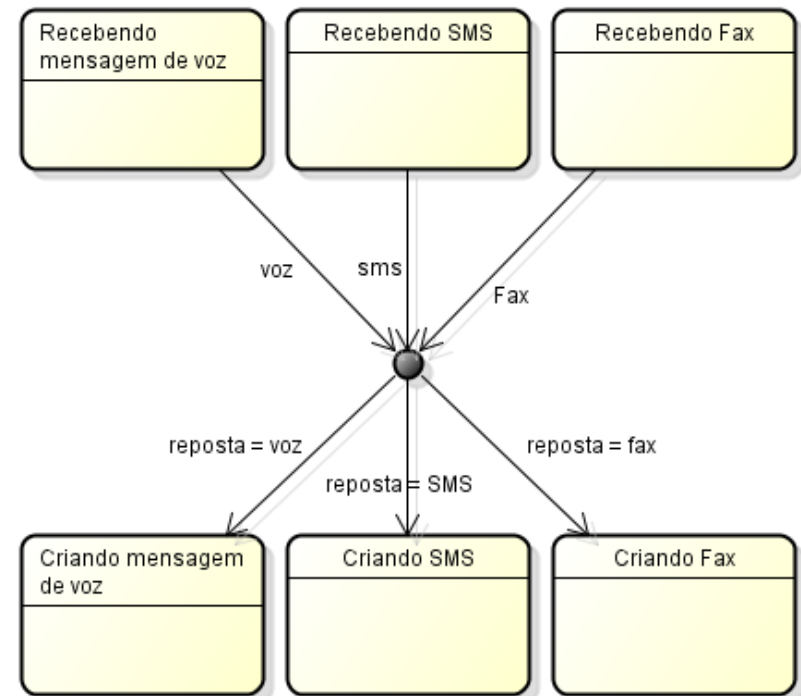
# Escolha

- É exibida como um losango, com uma transação chegando e duas ou mais saindo;
- Representa um desvio dinâmico, ou seja, quando **só é possível determinar o caminho em tempo de execução.**



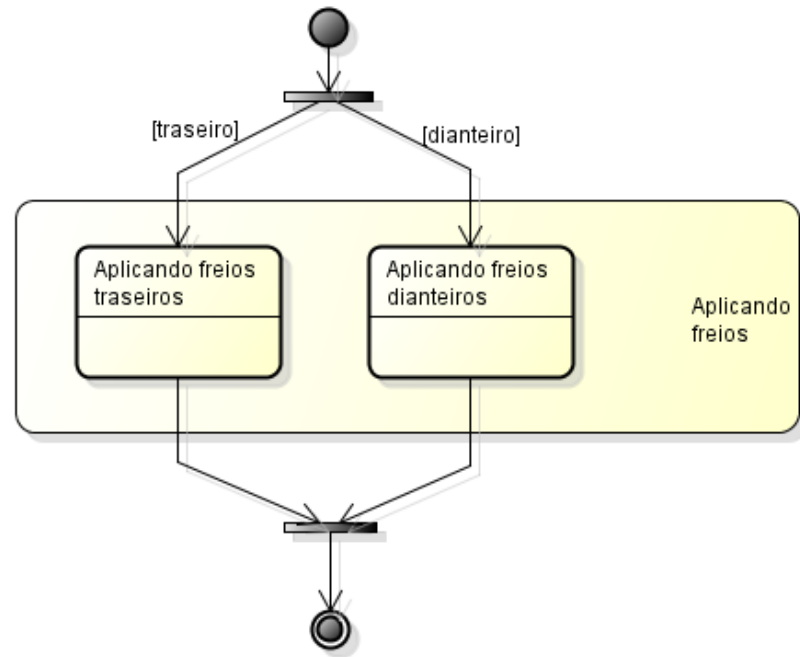
# Ponto de junção

- São utilizados para **agrupar múltiplas transições**.
- Representam desvio condicional estático, ou seja, **o caminho é determinado em tempo de projeto**.
  - A escolha representa um desvio dinâmico.



# Concorrência

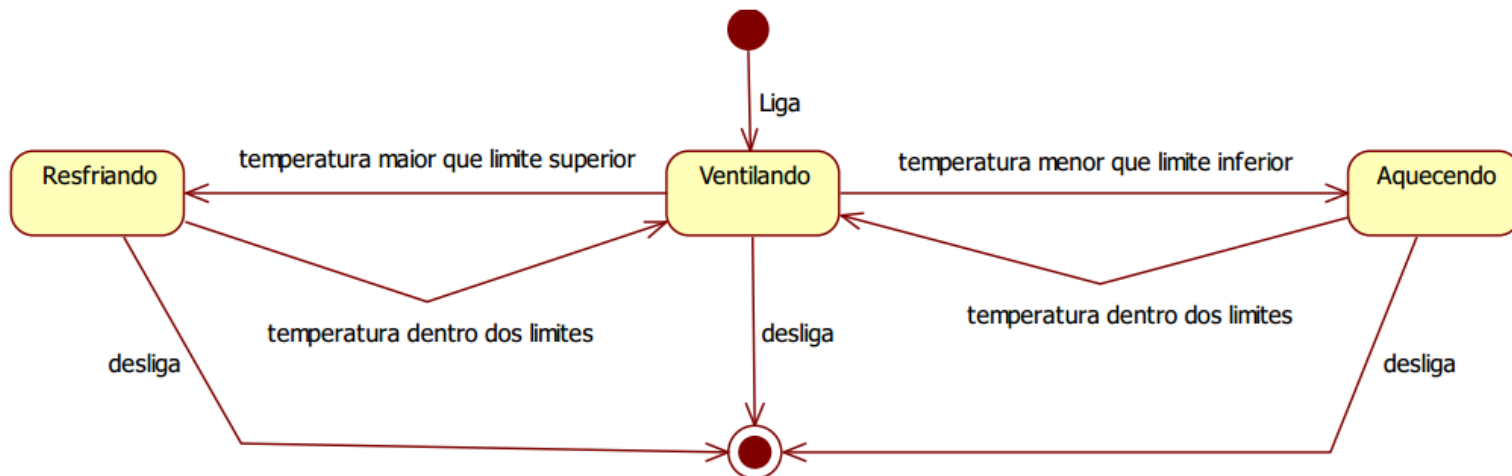
- Um estado pode ser dividido em duas regiões, contendo sub estados que executam concorrentemente.





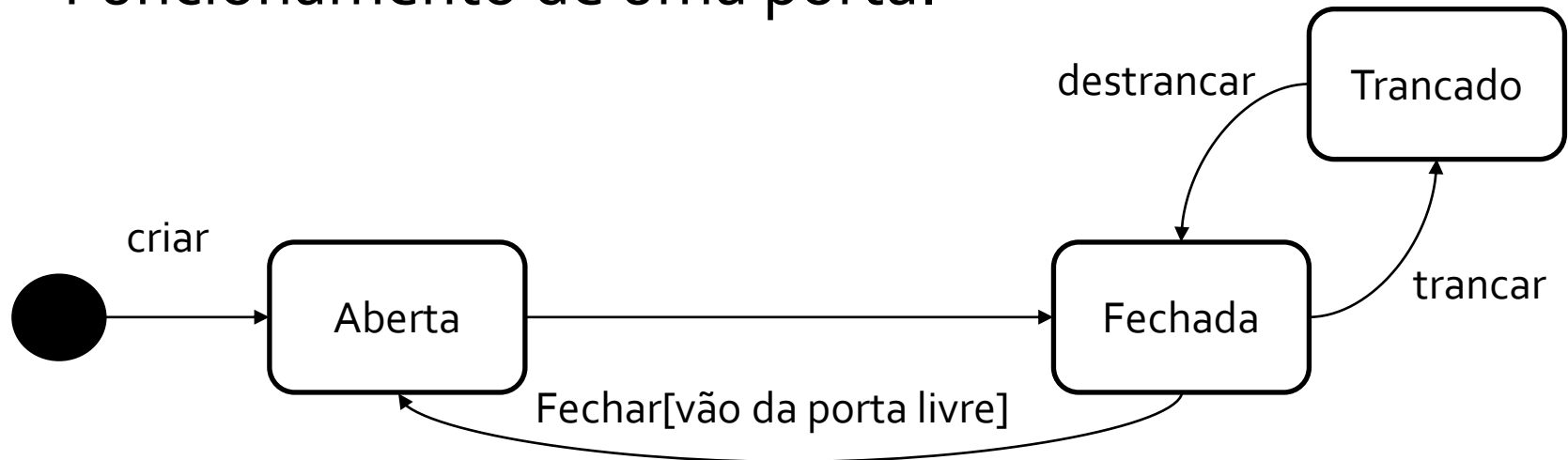
# Observações

- Faça inicialmente um diagrama simples, e complique somente se for necessário
- O que aparece em quase todo diagrama
  - Estados com nome
  - Transições com gatilho
- Exemplo simples no nível de análise para um condicionador de ar:



# Exemplo

- Funcionamento de uma porta.



- A porta pode estar em dos três estados: “aberta”, “fechada” ou “trancada”.
- O objeto pode responder aos eventos: fechar, abrir, trancar e destrancar.
- Nem todos os eventos são válidos em todos os estados.

# Exercício

- Matrícula
  - na Universidade
  - em disciplina
- Pedidos em e-commerce tradicional

# Ferramentas de Modelagem

- StarUML (avaliação por tempo indefinido)
  - <https://staruml.io/>
- Draw.io
  - <https://app.diagrams.net/>
  - Web

# Referências

- Boock, G. and Rumbaugh, J. The Unified Modeling Language User Guide . Addison-Wesley, 1999
- Arlow, J. and Neustadt, I. UML 2 and the Unified Process: Practical Object-Oriented Analysis and Design, 2nd Edition, The Addison-Wesley Object Technology Series, 2005.
- Rumbaugh, J.; Jacobson, I. and Booch , G. The Unified Modeling Language Reference Manual, 2nd Edition, The Addison-Wesley Object Technology Series, 2004.
- Boock, G.; Rumbaugh, J. and Jacobson, I; Unified Modeling Language User Guide, 2nd Edition, The Addison-Wesley Object Technology Series, 2005.
- Jacobson, I; Boock, G. and Rumbaugh, J., Unified Software Development Process, Addison-Wesley, Janeiro 1999.
- Larman, C. Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design Prentice-Hall, New Jersey - USA, 1997



INSTITUTO DE CIÊNCIAS EXATAS

DEPARTAMENTO DE COMPUTAÇÃO

Perguntas?