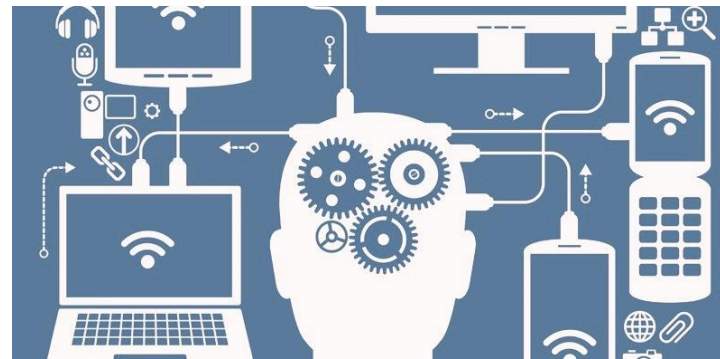


# Engenharia de Software 2



*Prof. Esp. João Paulo S. Araújo*

# Modelagem UML

## *Diagrama de Estados*

# Introdução

- *Objetos* do mundo real se encontram em **estados** particulares a cada momento.
  - Uma jarra está *cheia* de água
  - Uma pessoa está *cansada*.
- Da mesma forma, cada *objeto* participante de um sistema de software orientado a objetos se encontra em um **estado** particular.
- Um *objeto* muda de **estado** quando acontece algum **evento** interno ou externo ao sistema.

# Introdução

- Durante a **transição** de um estado para outro, um objeto realiza determinadas ações dentro do sistema.
- Quando um objeto transita de um estado para outro, significa que o sistema no qual ele está inserido também está mudando de estado.

# Diagrama de Estados

- Modela os **comportamentos** possíveis dos **objetos** de uma classe ao longo de seu tempo de vida (ciclo de vida de um objeto)
  - *Eventos, transições, estados entre eventos.*
- O Diagrama de Estados descreve:
  - *Todos os estados possíveis de um objeto*
  - *Como o estado de um objeto muda a partir de eventos.*

# Diagramas de Estados

- Também conhecido como:
  - *Diagrama de Transição de Estados (DTE) ou;*
  - *Diagrama de Máquina de Estados (State Machine)*
- *É um exemplo de estrutura dinâmica na UML.*

# Estado (*State*) - Conceito

- Representa uma *situação estável de um objeto* que se prolonga durante um *intervalo de tempo*, durante o qual *os atributos não sofrem qualquer alteração de valor*, nem o objeto sofre estímulos externos;
- Cada *objeto* está num *estado particular* em cada momento;
- O estado é caracterizado pelos conceitos de duração e estabilidade.

# Diagrama de Estados - *Conceito*

- É uma técnica para descrever o comportamento de um objeto.
  - *Descreve todos os estados possíveis que um objeto pode assumir e o modo como o estado do objeto se altera em resultado de eventos que atingem o objeto.*
- Pode ser utilizado para representar os estados de um Caso de Uso ou mesmo de um Subsistema ou Sistema Completo.



# Exemplo de *Estados* típicos em um sistema

- Para logar em um sistema:
  - *Aguardando* a introdução de uma senha.
- Em um sistema de fabricação de cerveja:
  - *Misturando* ingredientes
  - *Aquecendo* uma mistura.
- Em um sistema de motorização:
  - *Acelerando* um motor.
- Em um sistema de uma distribuidora de combustível:
  - *Enchendo* o tanque.

# Outros estados típicos de um sistema

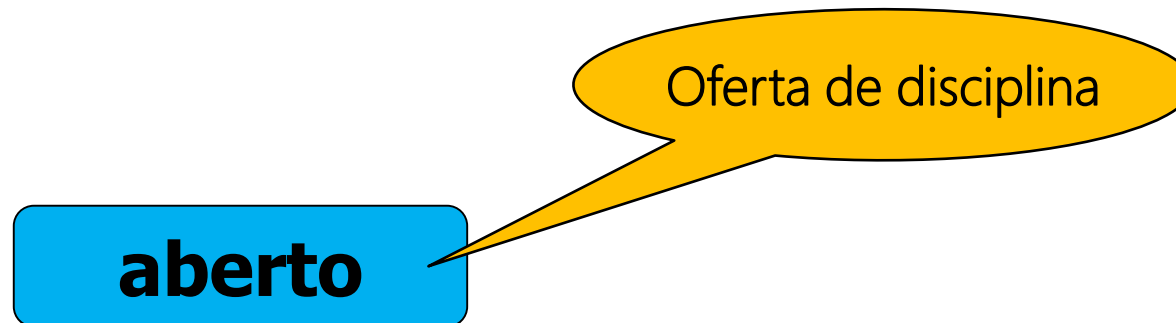
- **Aguardando** o próximo comando;
- **Aguardando** dados em um instrumento;
- **Verificando** dados;
- **Ocioso** – sem fazer nada.

# Diagramas de Estados – *Principais elementos*

- Estados;
- Transições;
- Eventos;
- Condições de guarda;
- Ações;
- Pontos de junção.

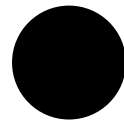
# Diagramas de Estados - *Estados*

- Um estado de um objeto pode ser caracterizado pelo **valor** de um ou mais **atributos** do objeto:
  - *Exemplo: Um objeto de uma disciplina ofertada pode estar **aberto** (se há vaga) ou **fechado** (se não há mais vaga). Nesse caso, o estado é definido de acordo com o número de alunos matriculados.*
- A situação de uma conta bancária passa para o **vermelho** quando o seu saldo fica negativo.

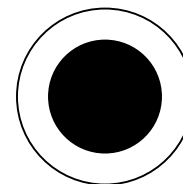


# Diagramas de Estados - *Elementos*

- Estado Inicial
  - Indica o estado de um objeto quando ele é criado;
  - Só pode haver um estado inicial em um diagrama.
- Estado Final
  - Indica o fim do ciclo de vida de um objeto;
  - É opcional e pode haver mais de um estado final em um diagrama.



***Estado Inicial***



***Estado Final***

# Diagramas de Estados - *Transições*

- Os estados estão associados a outros pelas transições;
- Quando uma transição entre estados ocorre, diz-se que a transição foi disparada;
- Representado por uma linha conectando os estados, com uma seta apontando para um dos estados;
- Uma transição pode ser rotulada com uma expressão da seguinte forma:

evento (lista-parâmetros) / [guarda] / ação



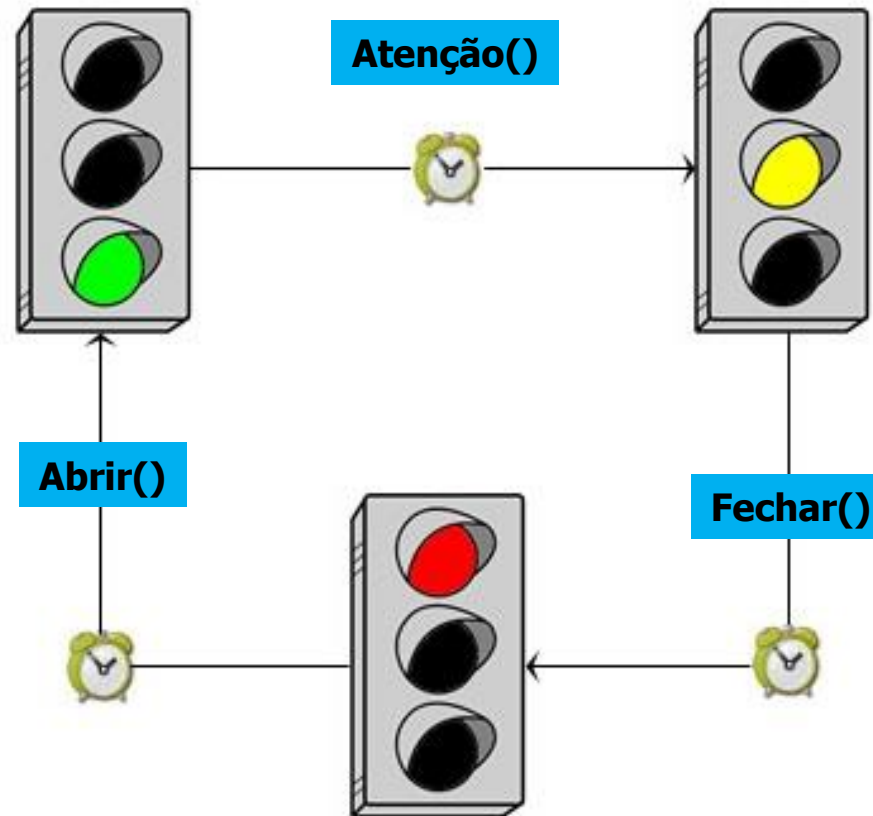
# Diagramas de Estados - *Eventos*

- *Um evento é algo que acontece em algum ponto no tempo e que pode modificar o estado de um objeto.*
- Uma transição possui um evento associado;
- A ocorrência de um evento provoca a transição entre estados de objetos de alguma classe pertencente ao sistema.
- Exemplos de Eventos:
  - Finalizar()
  - Pagar()
  - Devolver()
  - Interromper()



# Diagramas de Estados - *Eventos*

## Exemplo





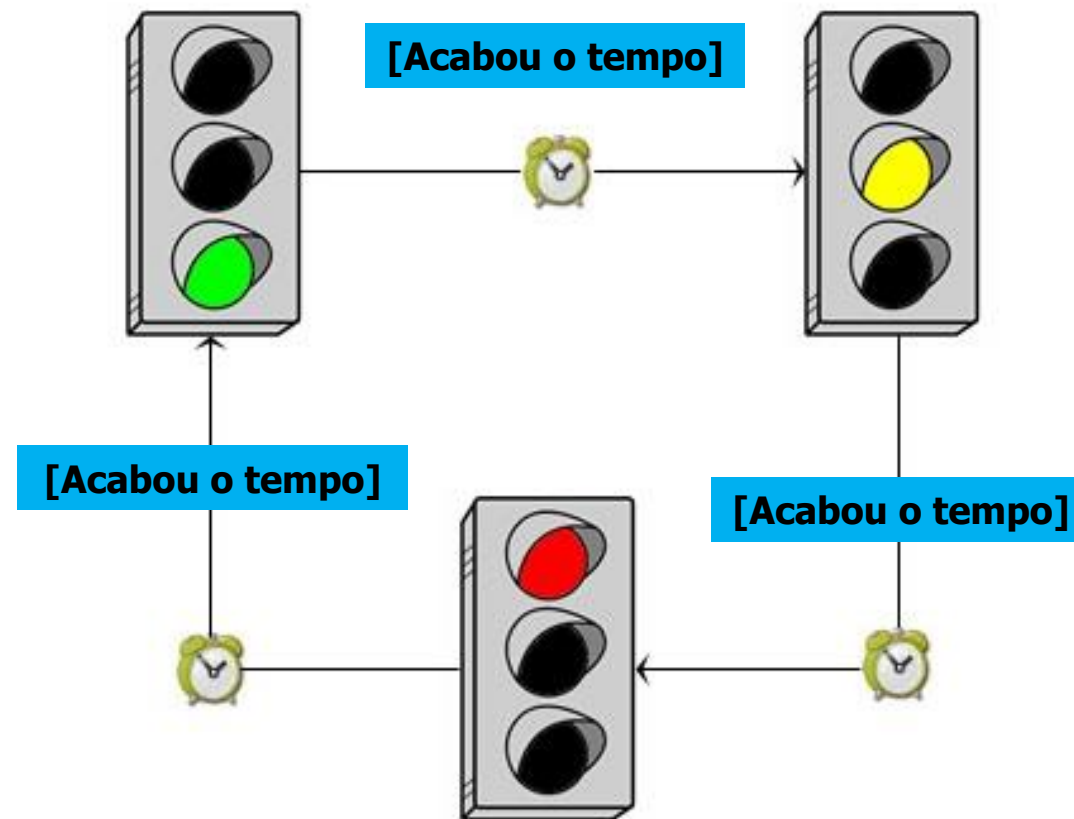
# Diagramas de Estados – *Condição de Guarda*

- É uma expressão de valor lógico que condiciona o disparo de uma transição;
- A transição correspondente é disparada se, e somente se, o evento associado ocorre e a condição de guarda é verdadeira.
  - *Uma transição que não possui condição de guarda é sempre disparada quando o evento ocorre.*
  - *A condição de guarda pode ser definida utilizando-se parâmetros passados no evento e também atributos.*



# Diagramas de Estados – *Condição de Guarda*

## Exemplo



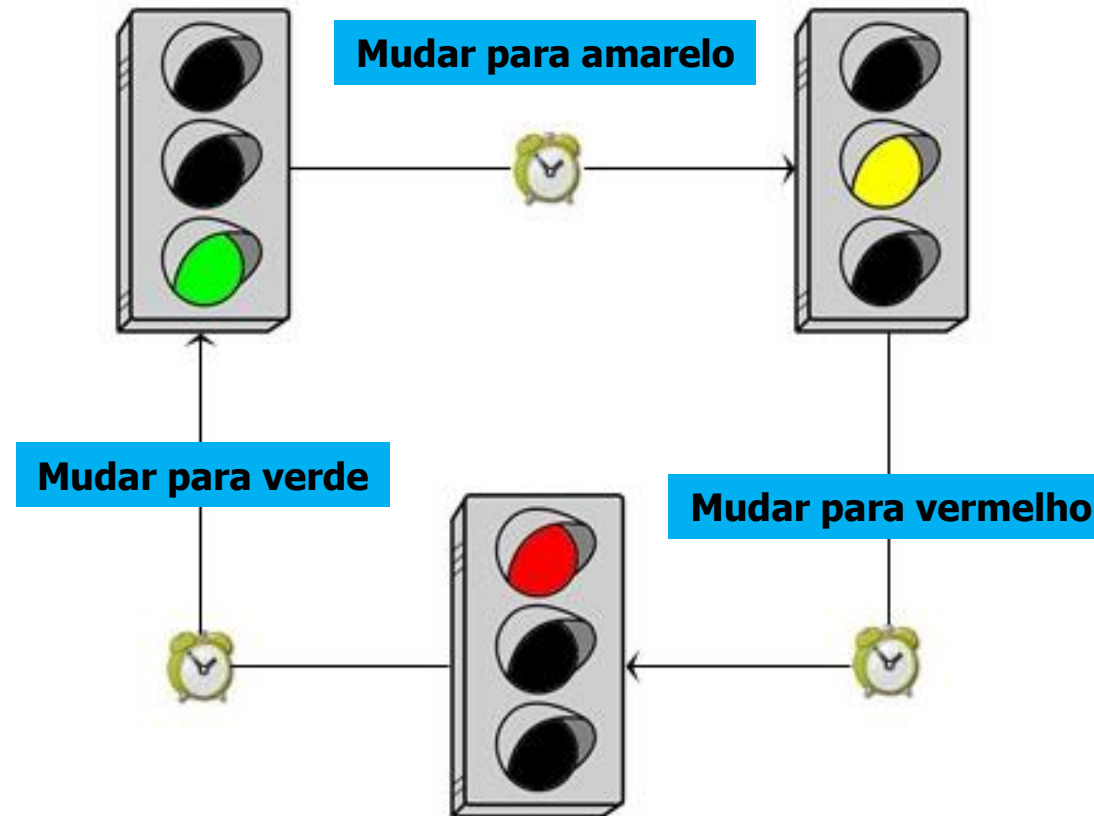
# Diagramas de Estados – Ações

- Ao transitar de um estado para outro, um objeto pode realizar *uma* ou *mais* ações.
- Uma ação é uma expressão definida em termo dos *atributos*, *operações da classe* ou dos *parâmetros do evento*.
- A ação associada a uma transição é executada se e somente se a transição for disparada.



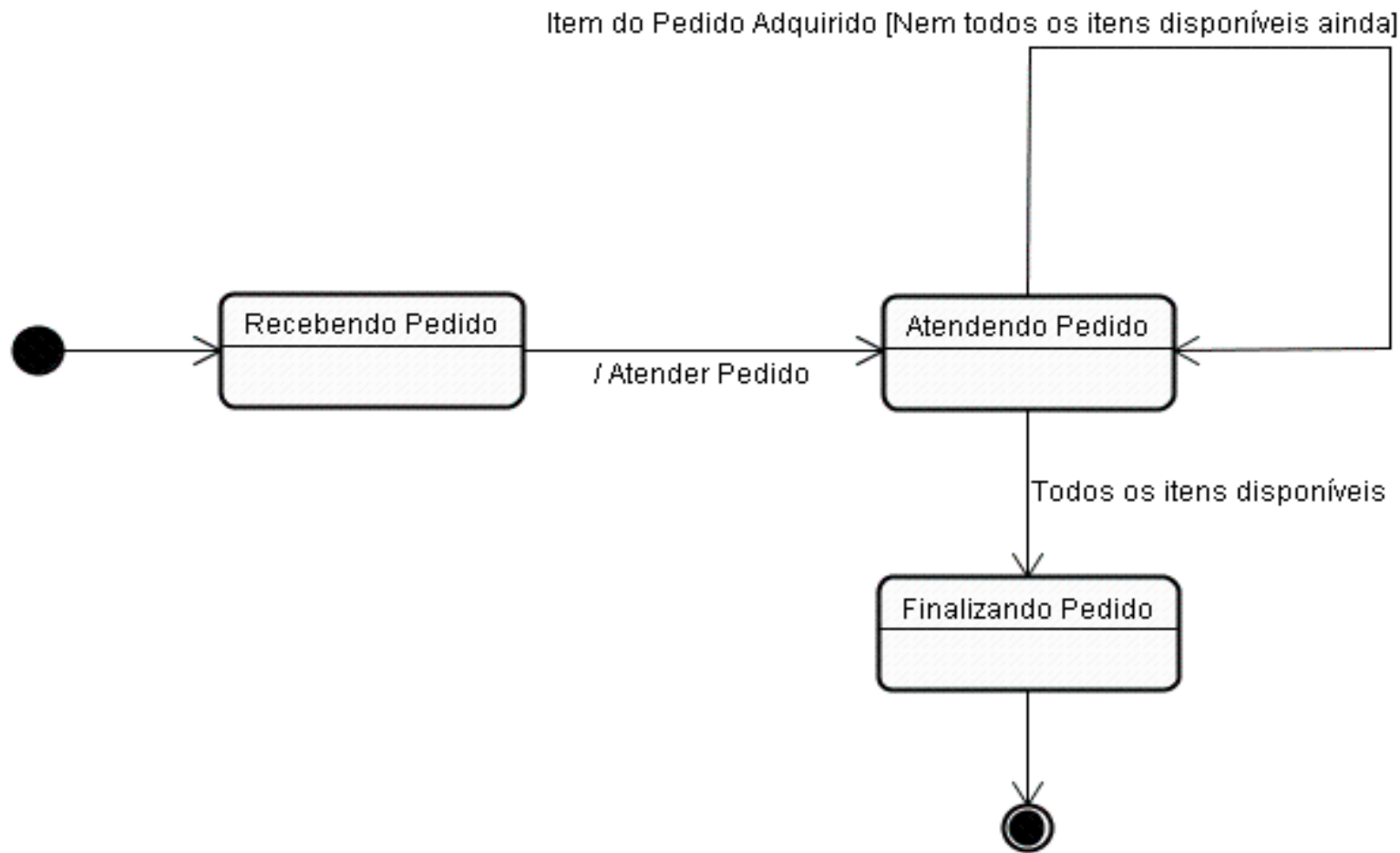
# Diagramas de Estados – *Ações*

## Exemplo



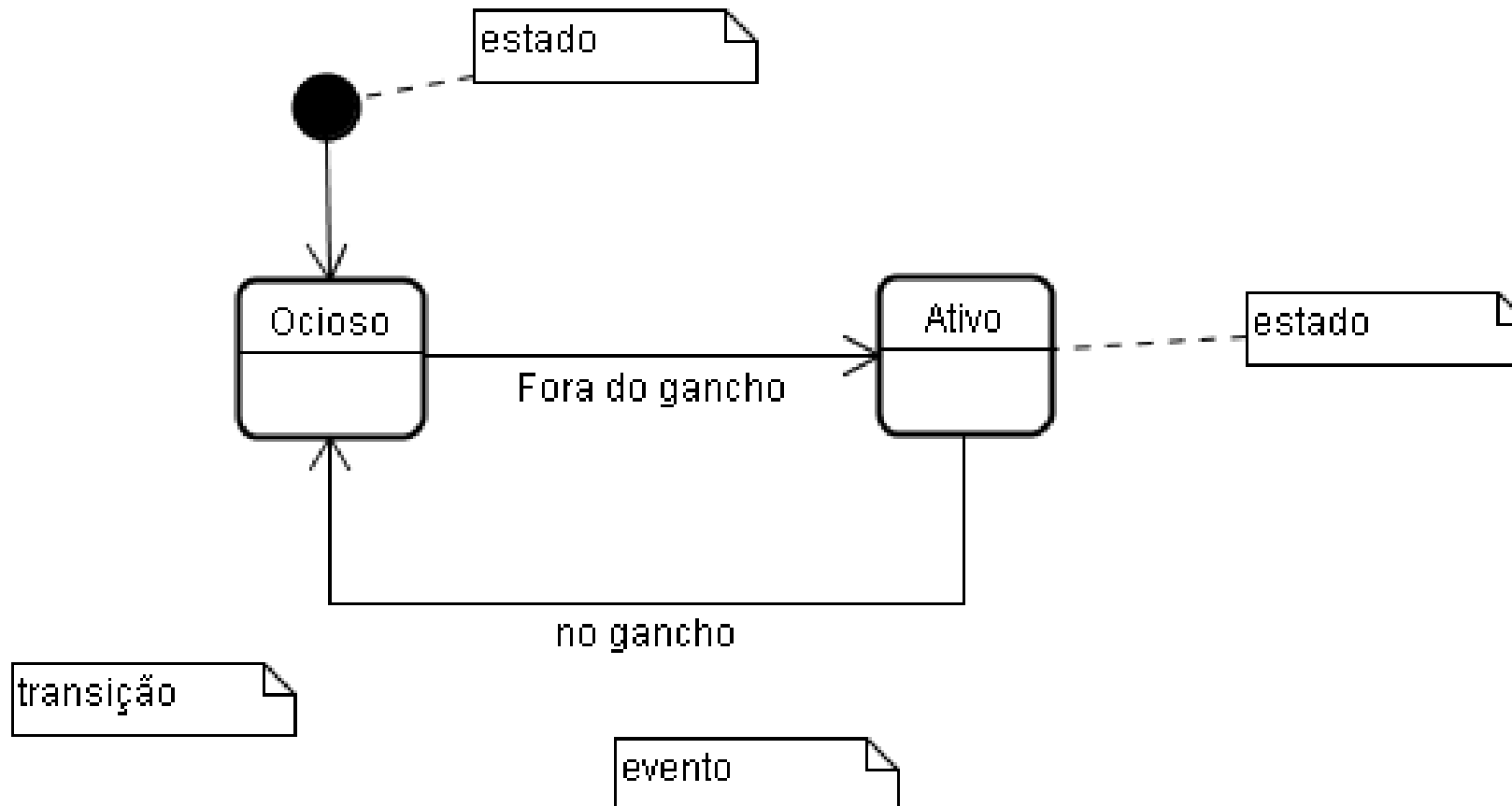
# Diagramas de Estados – *Exemplo*

- Diagrama de Estados para um objeto **Pedido**



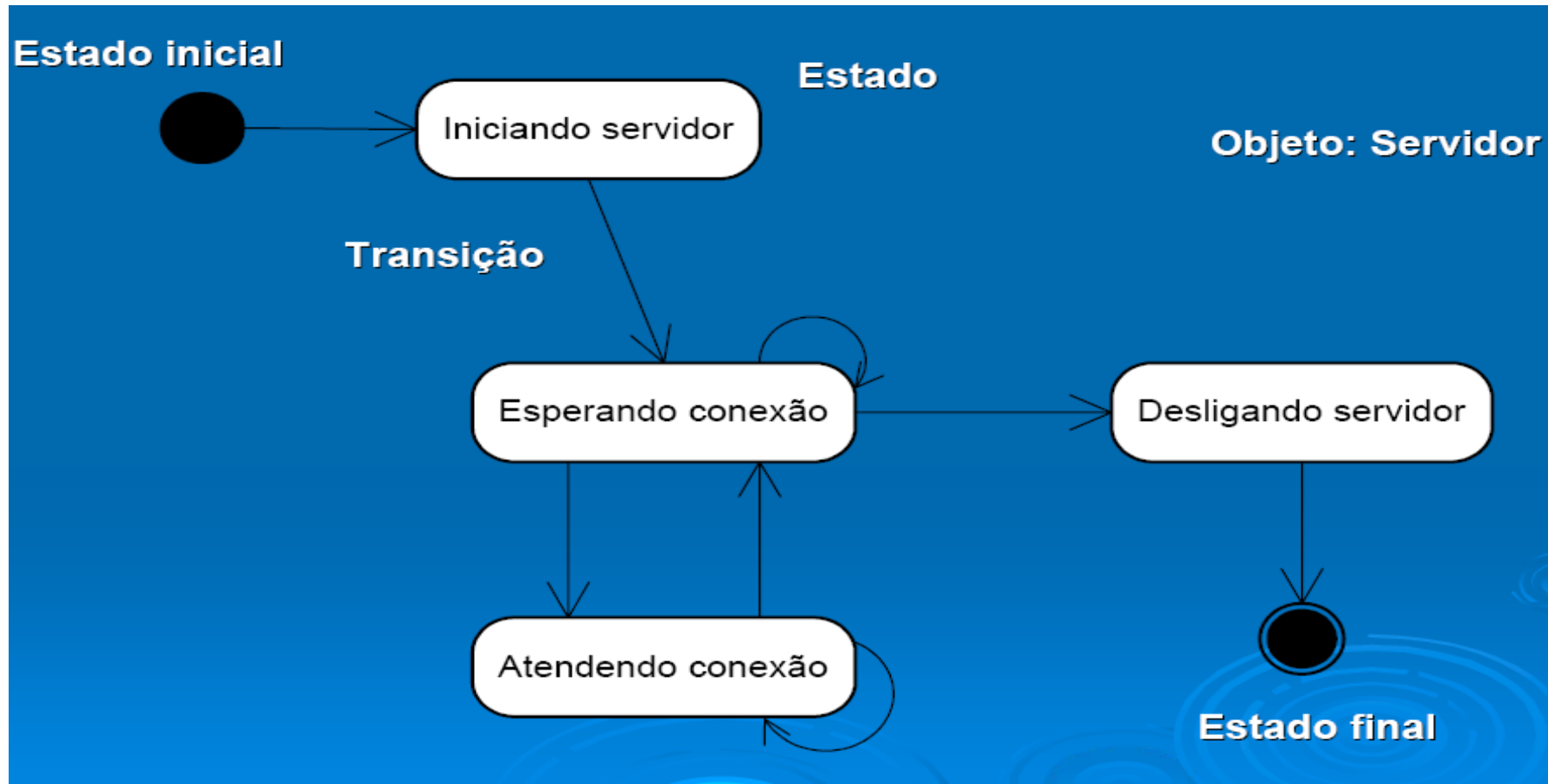
# Diagramas de Estados – *Exemplo*

- Diagrama de Estados de um telefone fixo...



# Diagramas de Estados – *Exemplo*

- Diagrama de Estados de um servidor...



# Diagramas de Estados – *Pontos de Decisão*

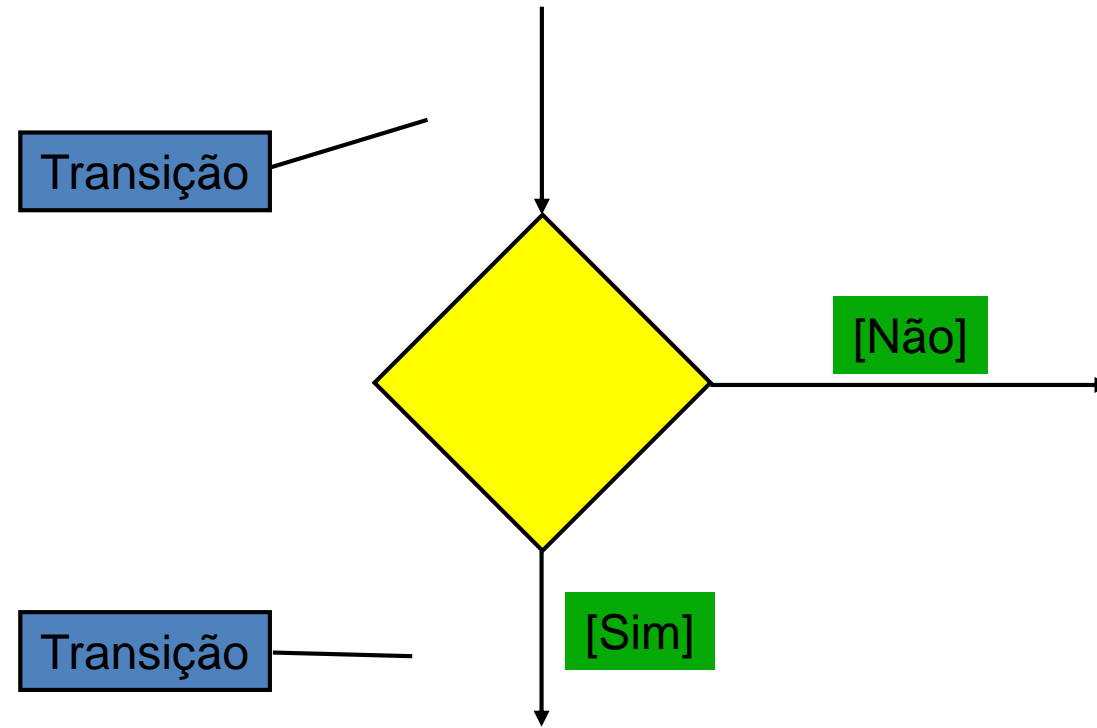
- Ponto de Decisão (ou ponto de ramificação)
  - Decisão é um recurso utilizado para controlar desvios no fluxo de controle de um diagrama de atividade.
  - É composto de condições booleanas e cada condição, quando satisfeita, dispara uma transição correspondente.
  - Cada opção é identificada por meio de uma condição guarda.
  - Possui uma única transição de entrada e várias transições de saída.
    - *Para cada transição de saída, há uma condição de guarda associada.*



# Diagramas de Estados – *Pontos de Decisão*

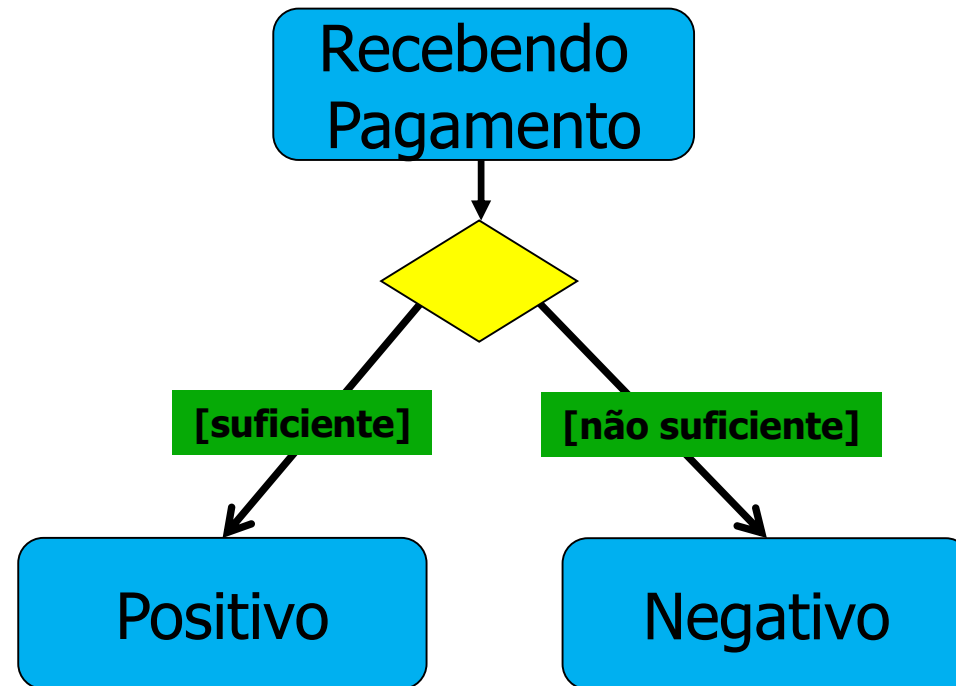
## Notação Gráfica

O losango do diagrama de Estado é um ícone de decisão, assim como nos fluxogramas ou diagrama de atividades (UML).



# Diagramas de Estados – *Pontos de Decisão*

## Exemplo



# Diagramas de Estados – *Pontos de Decisão*

- Cada condição deve ser mutuamente **exclusiva**, de modo que **somente uma opção** seja possível em qualquer ponto de decisão.
- Essa construção está relacionada a instruções **case** ou estruturas **if-then-else**.

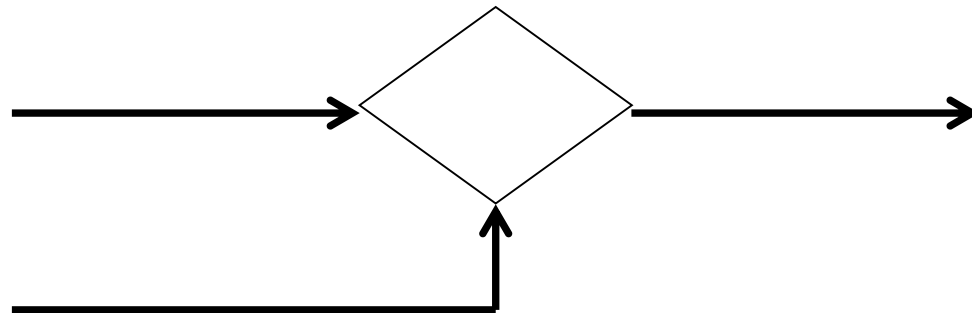
# Diagramas de Estados – *Pontos de União*

- Ponto de União (ou ponto de convergência)
  - Consiste no ponto onde dois ou mais caminhos alternativos se juntam e continuam em um único caminho.
    - *Ou seja, existe mais de uma transição de entrada e apenas uma transição de saída.*
  - Reúne diversas transições que, direta ou indiretamente, têm um ponto de ramificação em comum.

# Diagramas de Estados – *Pontos de União*

## Notação Gráfica

- O ícone de losango também é usado para modelar um ponto de união, o local onde dois caminhos alternativos se juntam e continuam como um.



# Diagramas de Estados – *Barras de Sincronização*

- É utilizada quando da ocorrência de **estados paralelos** causados por **transições concorrentes**.
- Uma **transição** que começa numa barra de sincronização **somente** é executada quando **TODAS as transições** que chegam nesta barra ocorrerem.
- Quando uma **transição** chega a uma barra de sincronização, **as transições** que partem desta barra ocorrem **simultaneamente** e **independentemente**.
  - *Ou seja, dois ou mais fluxos (transições) são executados simultaneamente.*

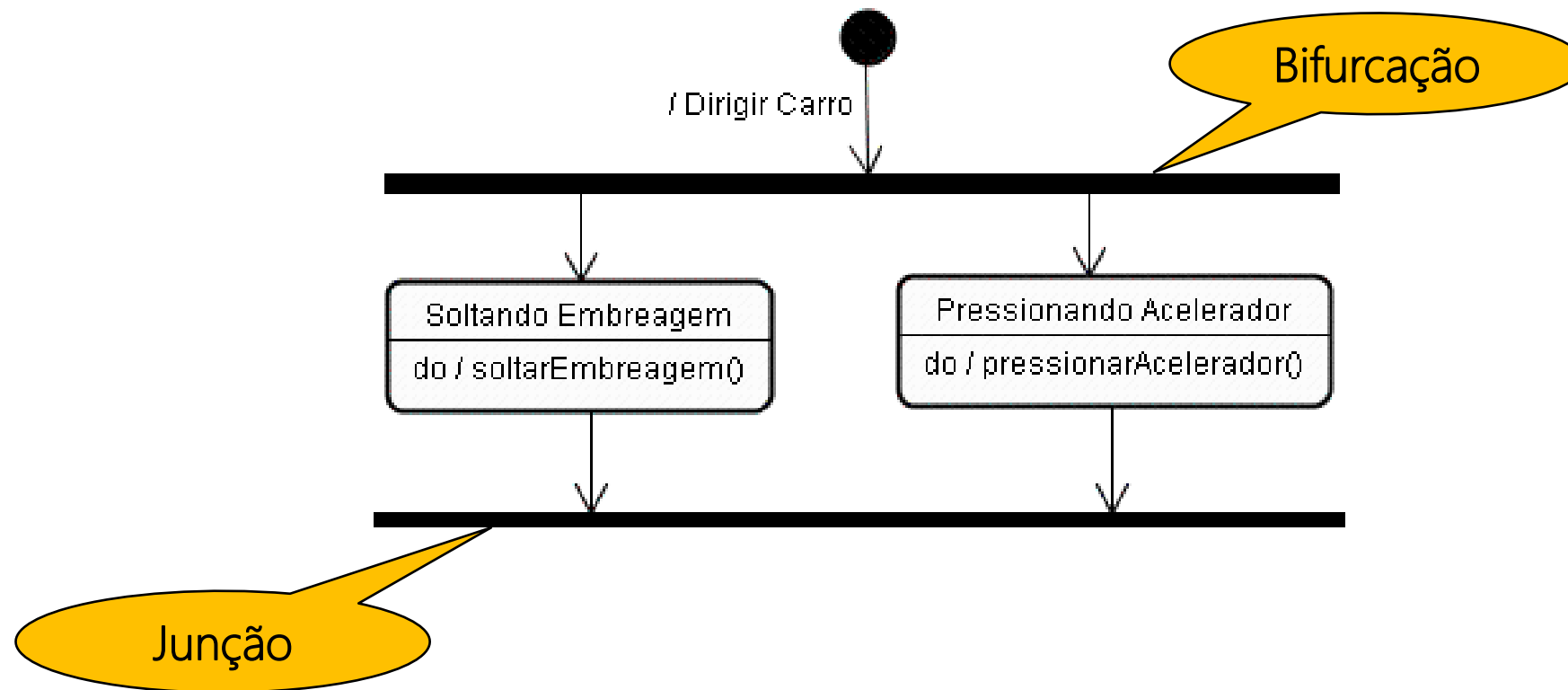
# Diagramas de Estados – *Barras de Sincronização*

Existe dois tipos de barras de Sincronização:

1. Barra de bifurcação (concorrência): recebe **uma transição de entrada**, e **cria dois ou mais fluxos (transições) de controle paralelos**.
  - Cada fluxo é executado independentemente e em paralelo com os demais.
2. Barra de junção (sincronismo): recebe **duas ou mais transições de entrada** e **une os fluxos em um único fluxo**.

# Diagramas de Estados – *Barras de Sincronização*

## Notação Gráfica





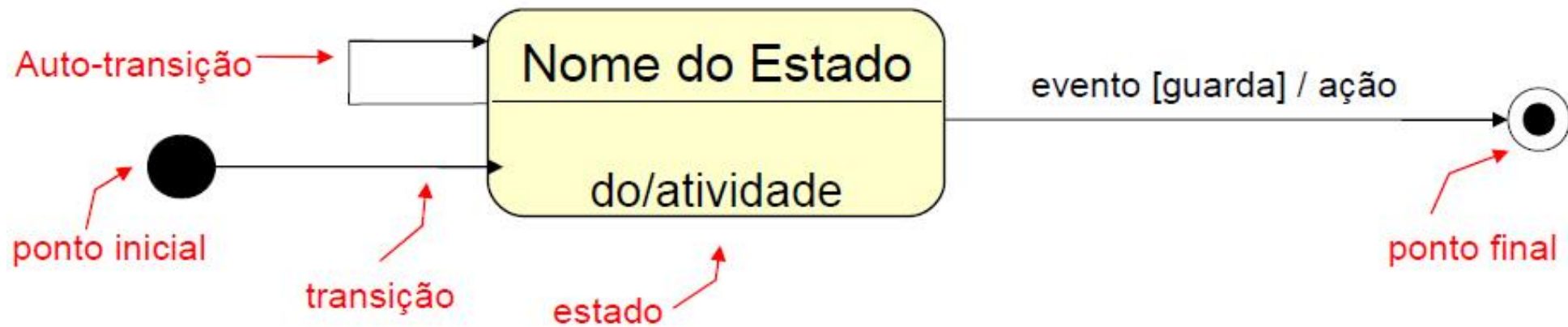
# Diagramas de Estados

## Verificando um Diagrama de Estados

- *Todos os estados podem ser atingidos?*
- *Todos os estados tem saída?*
- *Foram definidos todos os estados possíveis?*
- *Em cada estado o sistema reage adequadamente a todas as condições possíveis?*

# Diagramas de Estados

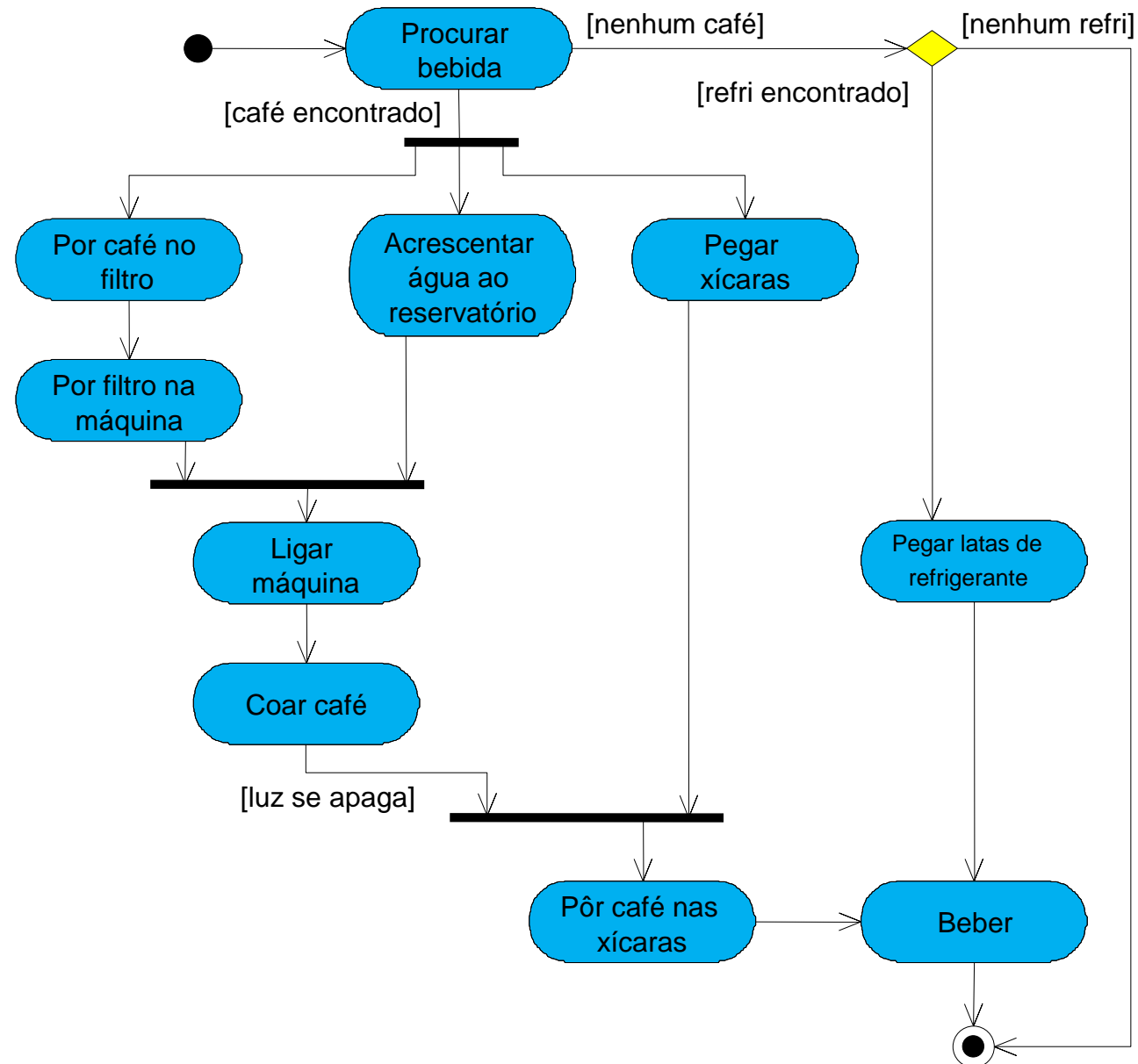
## *Resumo*



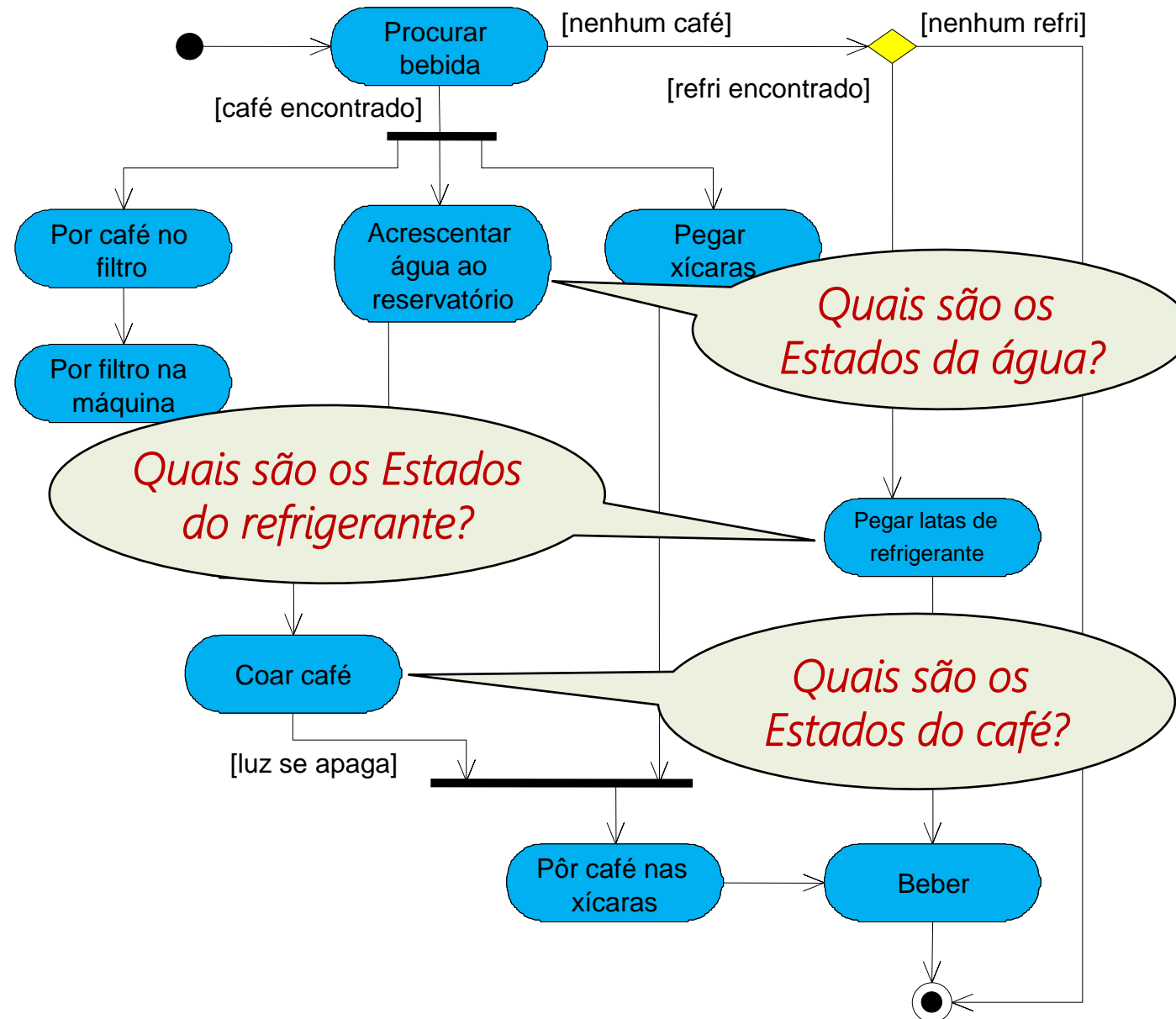
## *Exemplo 1 – Obter Estados a partir de Atividades*

- Preparar/Pegar Bebida
  - Definir os estados da água
  - Definir os estados do refrigerante
  - Definir os estados do café.

# Exemplo 1 – Diagrama de Atividades

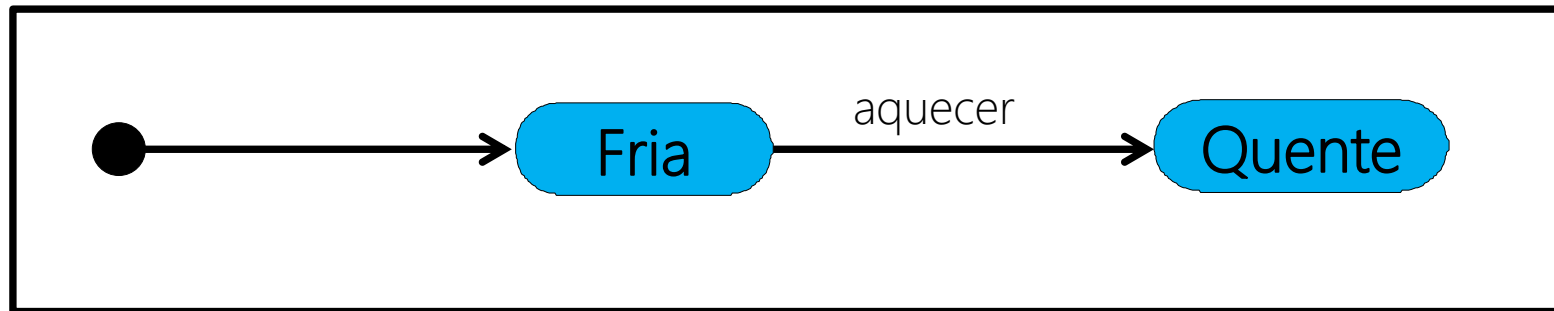


# Exemplo 1 – Diagrama de Atividades



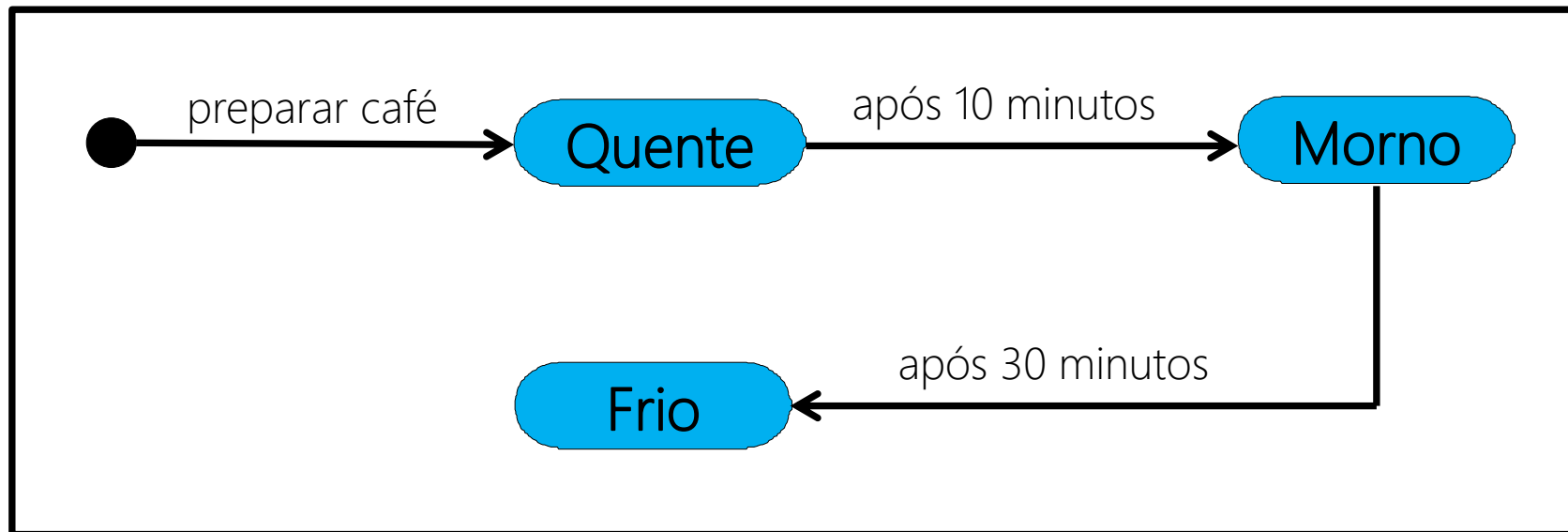
# Exemplo 1 – Diagrama de Estados

## Estados da água



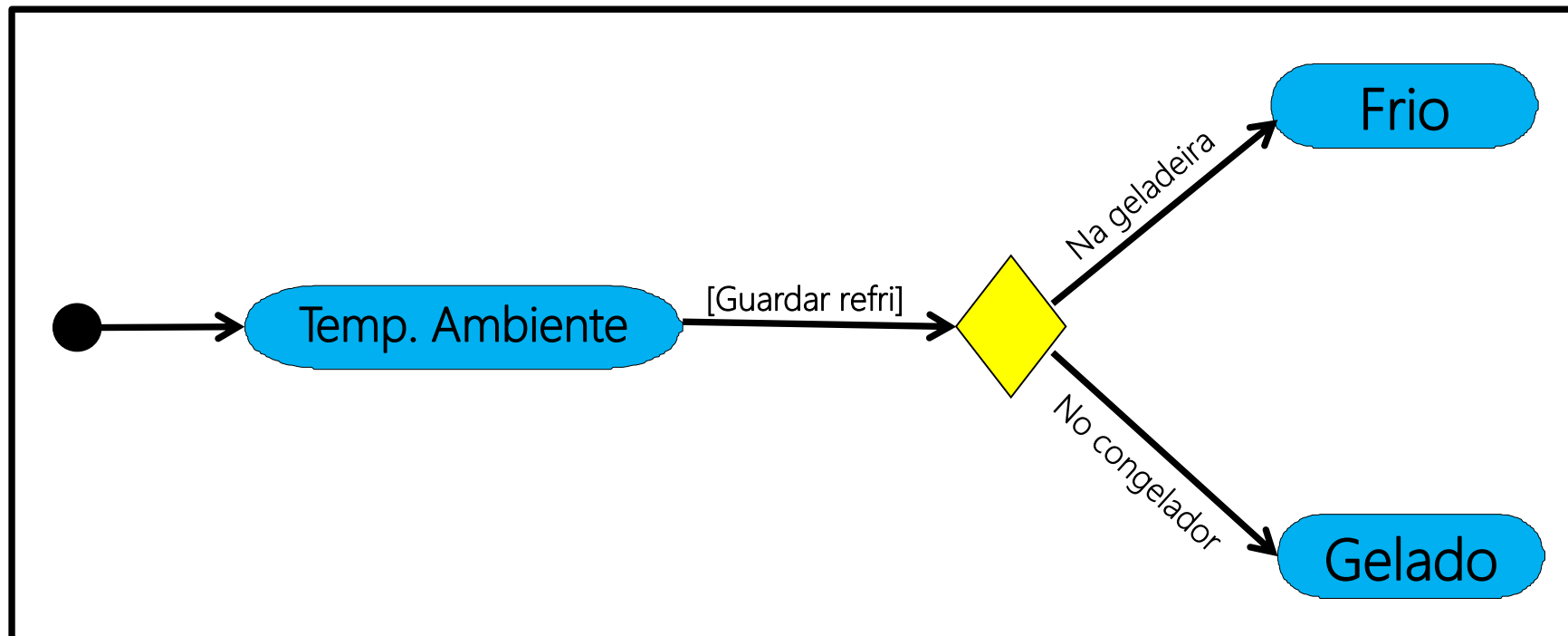
# Exemplo 1 – Diagrama de Estados

## Estados do café



# Exemplo 1 – Diagrama de Estados

## Estados do refrigerante





## *Diagrama de Estados – Exercício*

- 1) Identifique os objetos e os estados de um sistema que utiliza um sensor de luminosidade que conforme a iluminação detectada acende ou apaga uma lâmpada;
  - *Caso detecte iluminação apaga a lâmpada. Caso contrário acende a lâmpada.*
- ❖ *Após identificar os objetos e seus estados possíveis crie um diagrama de estados para representa-los;*

## Diagrama de Estados – Exercício

- 2) Crie um Diagrama de Estados de um objeto de pedido de venda para o modelo da loja de roupas da atividade anterior.
- *Considere as etapas envolvidas na Venda e que o objeto pedido pode ter vários estados:*
    - *Aguardando verificação de estoque do produto;*
    - *Aguardando Pagamento;*
    - *Cancelado;*
    - *Finalizado;*
    - *Entregue;*
- ❖ *Note que:*
- *Cada produto na venda corresponde a um item do Pedido;*
  - *A entrega só é autorizada após o pagamento ser confirmado;*
  - *Neste caso teremos que representar também algumas condições e transições de um estado para outro;*