



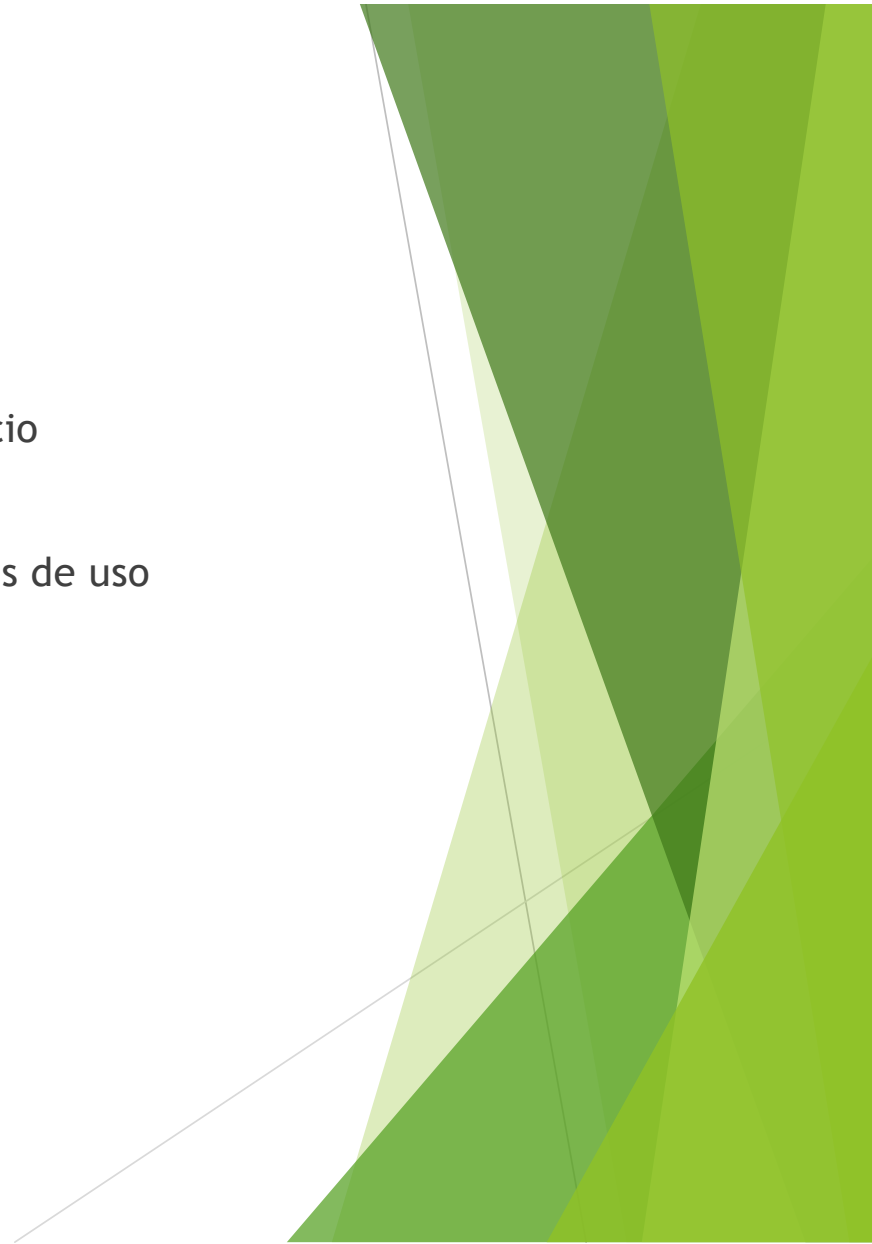
# Análise e Desenvolvimento de Software

2022/2023 - Diagramas de Atividades  
Instituto Politécnico do Cávado e do Ave

Pedro Emanuel Cardoso de Sousa  
pesousa@ipca.pt

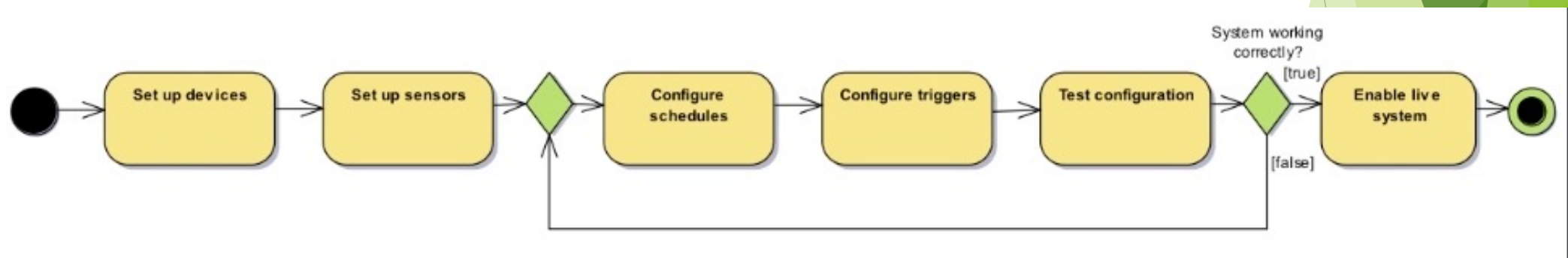
# Propósito

- ▶ Disponibilizar uma visão de alto nível dos processos de negócio
- ▶ Modelar os processos de negócio
- ▶ Analisar as atividades que fazem parte da execução dos casos de uso
- ▶ Documentar o algoritmo e um possível método de execução



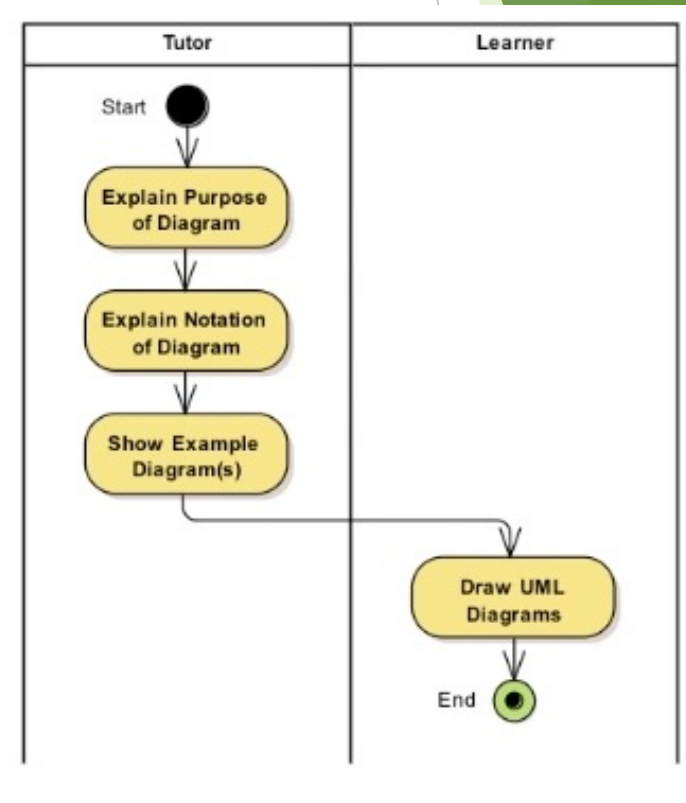
# Processos de negócio

- ▶ Modelar os processos de negócio de alto nível
- ▶ Disponibilizar um contexto para implementar passos com os casos de uso



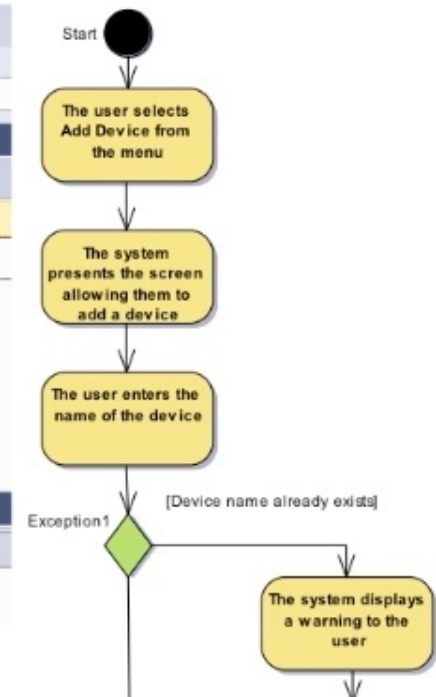
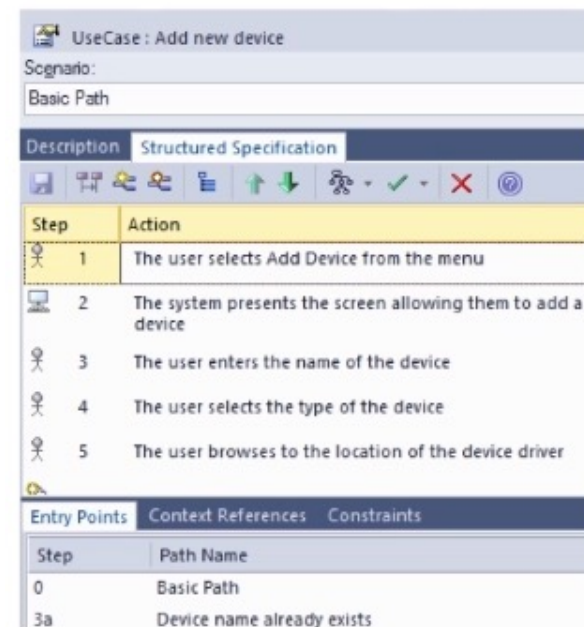
# Modelar os fluxos de trabalho

- ▶ Demonstrar aos participantes de uma atividade onde estes são responsáveis por interagir
- ▶ Os participantes podem ser:
  - ▶ Pessoas
  - ▶ Unidades organizacionais
  - ▶ Sistemas



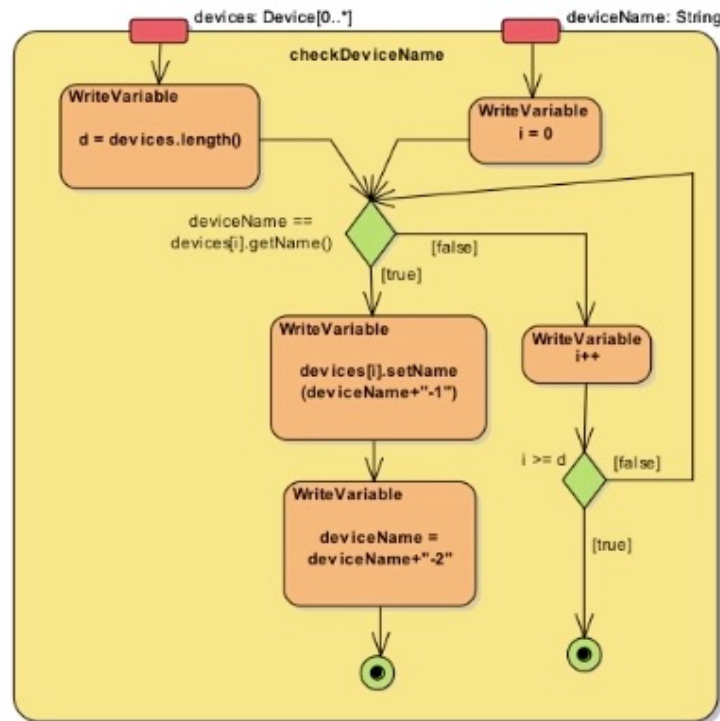
# Analizando atividades num caso de uso

- Os passos num caso de uso podem ser visualizados num diagrama de atividades
- Algumas ferramentas geram os diagramas de atividades a partir dos casos de uso



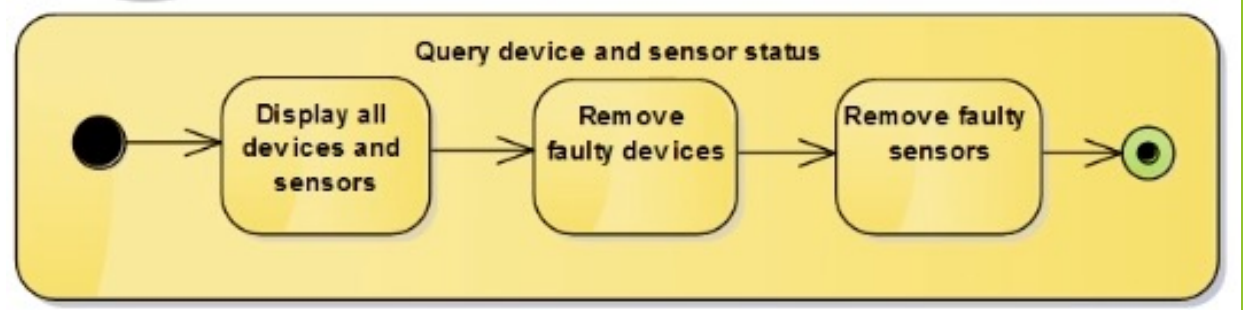
# Especificar um algoritmo

- Um diagrama de atividades pode representar um algoritmo através de um fluxo ou notação



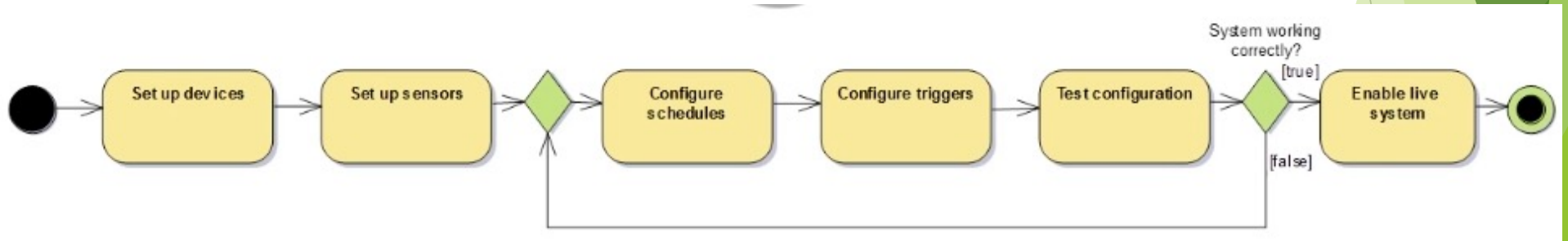
# Notação nos Diagramas de Atividades

- ▶ Fluxo de trabalho (workflow)
- ▶ Atividade
- ▶ Nós iniciais e finais
- ▶ Ações e atividades
- ▶ Controlo de fluxos



# Fluxo de trabalho (workflow)

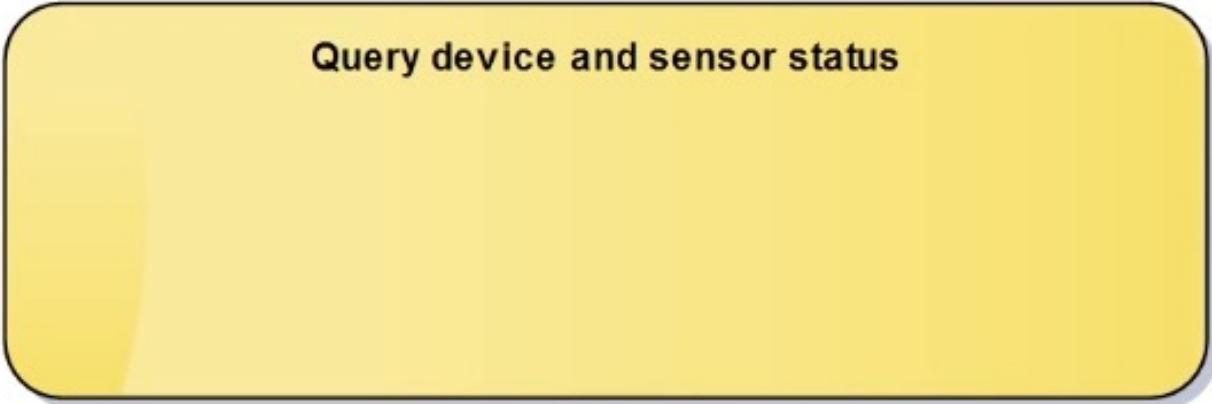
- ▶ Um diagrama de atividades é utilizado para modelar um fluxo de trabalho, através de ações ligadas, que podem ser com as próprias
- ▶ Temos nós de controlo, execução e objetos





# Atividade

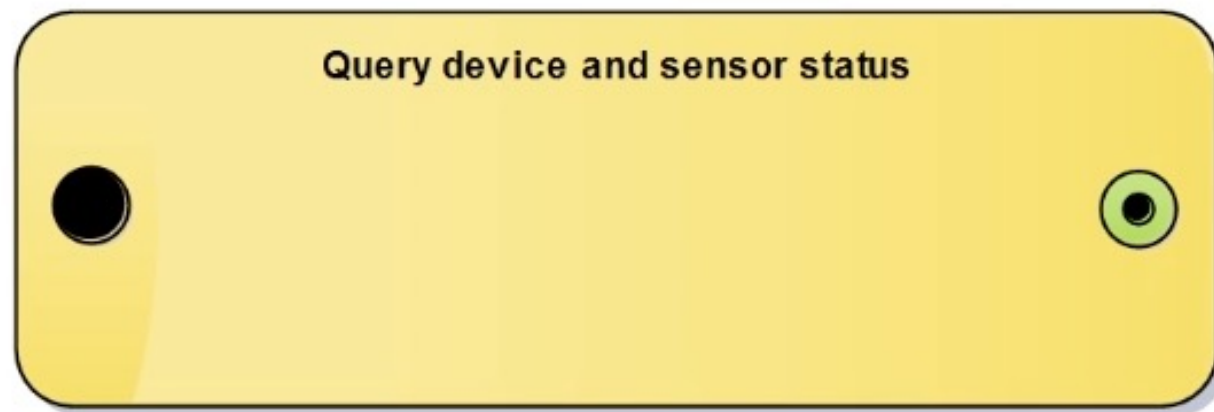
- ▶ Retângulo de cantos redondos com o nome na parte superior
- ▶ Podem estar implícitos ou esquecidos no diagrama



**Query device and sensor status**

# Nós iniciais e finais

- ▶ Inicial → círculo escuro e sólido
- ▶ Final → círculo com outro círculo mais pequeno no interior

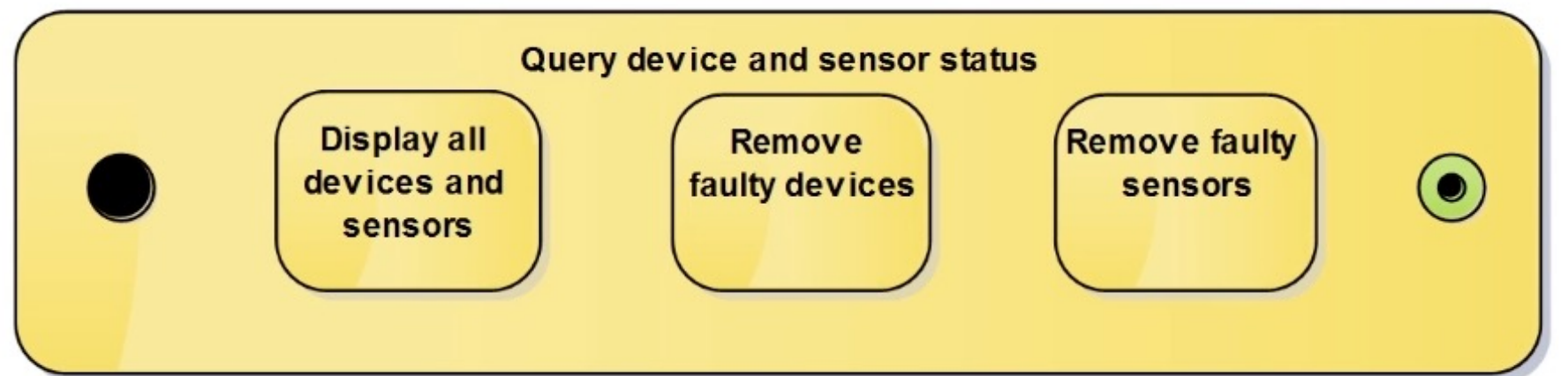


# Ações e Atividades

- ▶ Em retângulos com cantos redondos
- ▶ Com o nome a descrever a ação ou atividade
- ▶ As ações são os elementos que representam comportamento
- ▶ Ações são nós de execução
- ▶ Tecnicamente:
  - ▶ As atividades apenas são apresentadas nos diagramas de atividades, como parte do fluxo de trabalho, se estas forem estruturadas

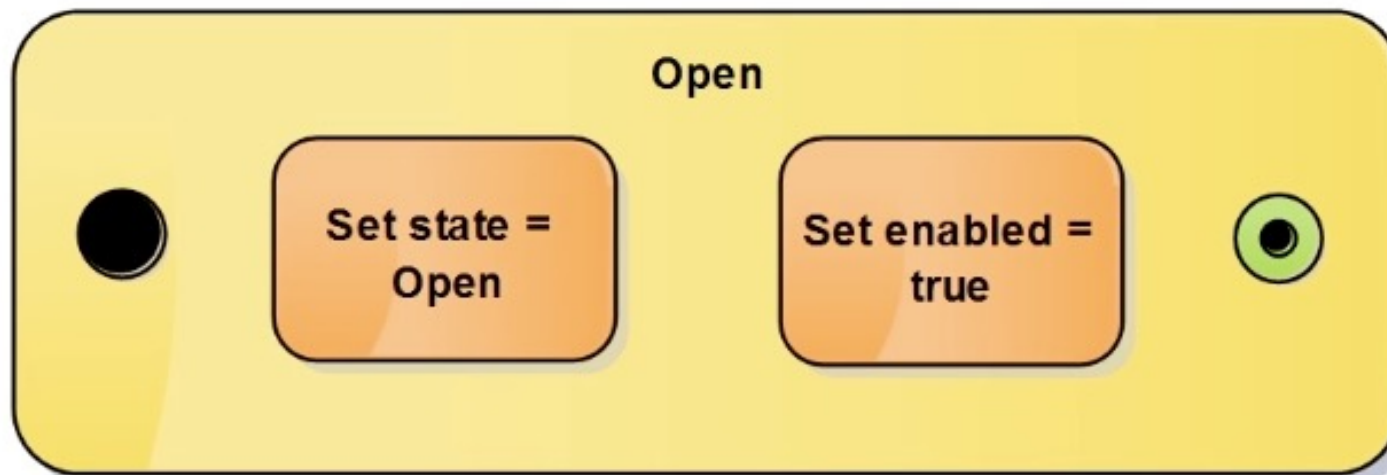
# Ações e Atividades

- ▶ Usar atividades para representar comportamentos de grande escala
- ▶ Processos de negócio ou atividades dos casos de uso



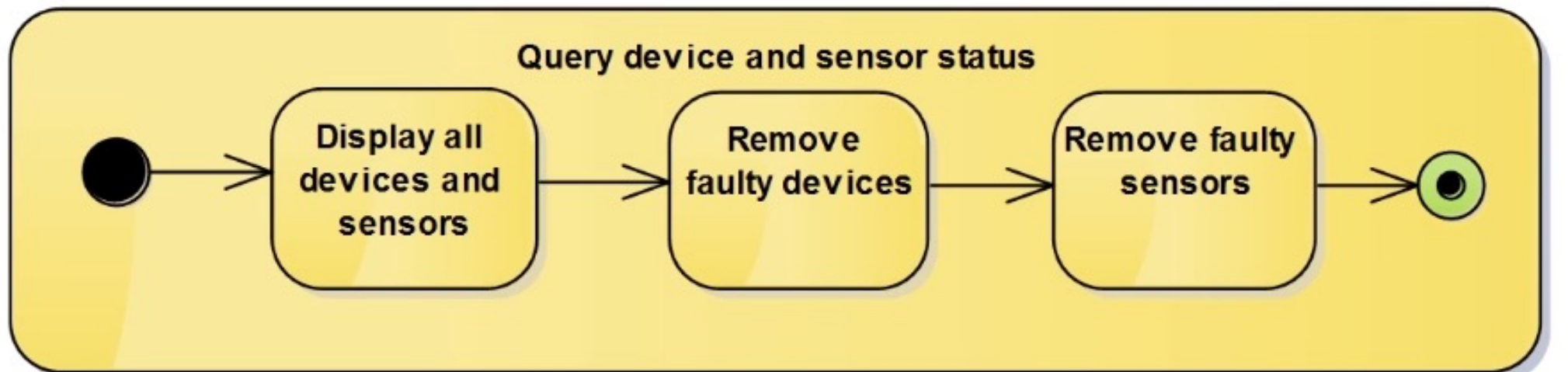
# Ações e Atividades

- ▶ Usar as ações para representar comportamentos de baixo nível
- ▶ Especificação de operações



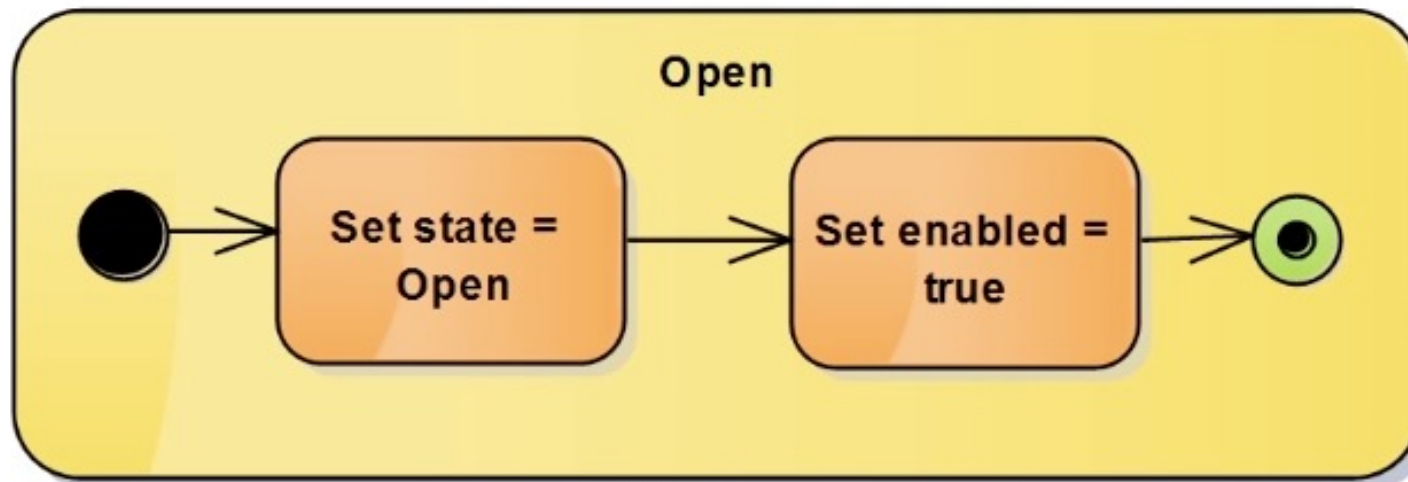
# Setas de controlo de fluxo

- ▶ As setas ligam as ações
- ▶ Setas com pontas abertas



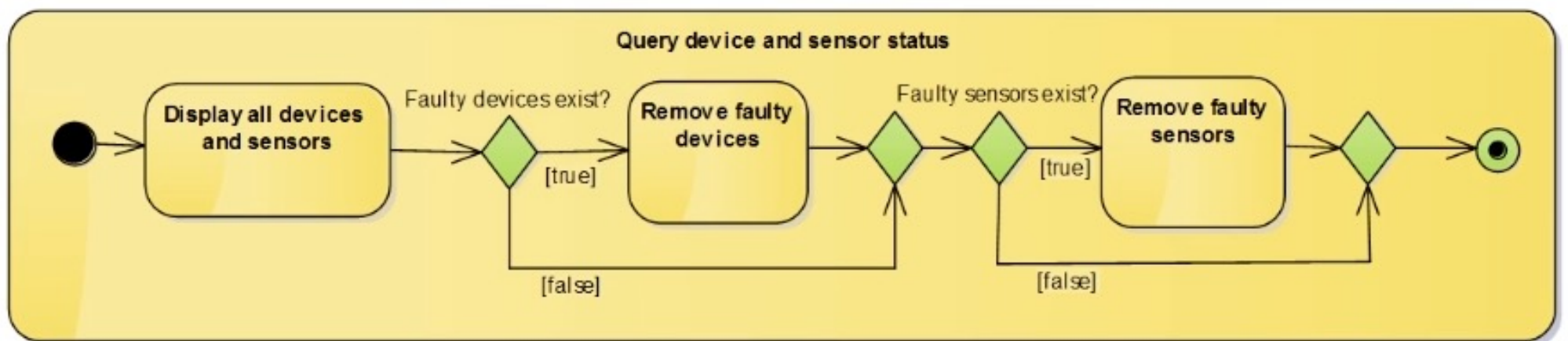
# Fluxos de controlo

- A mesma abordagem que tivemos para as atividades, no caso das ações



# Decisões nos Diagramas de Atividades

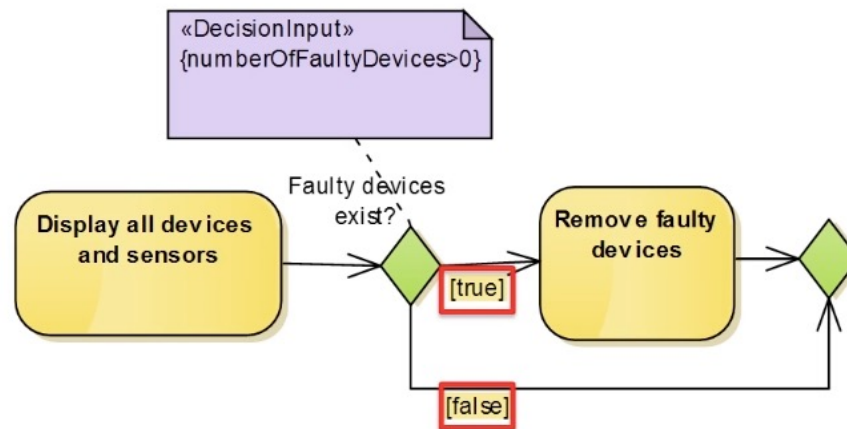
- ▶ Decisões e uniões
- ▶ Condições
- ▶ Quando não usar nós de decisões





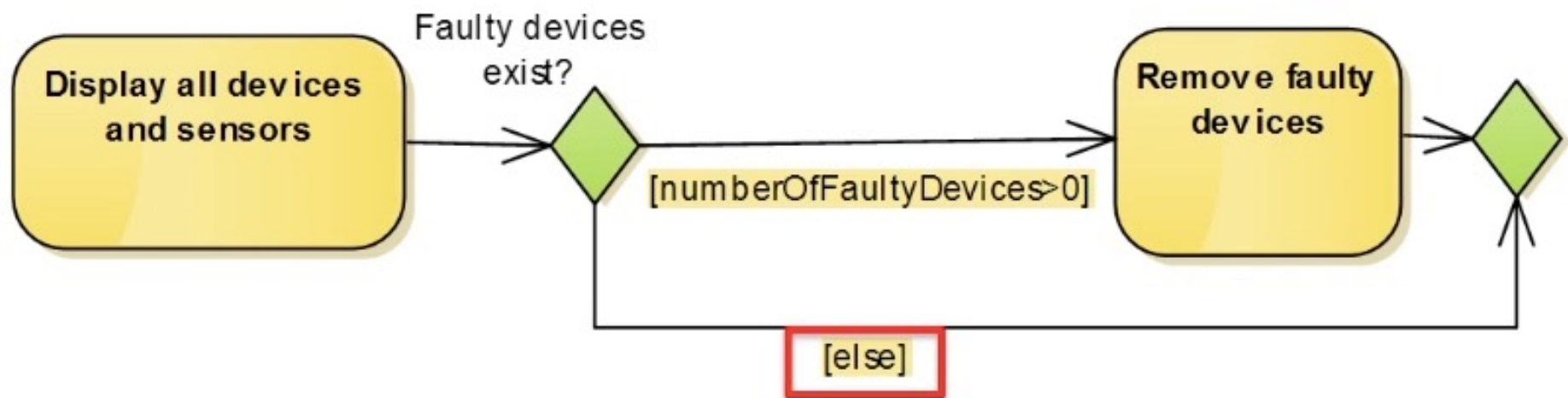
# Decisões e Uniões

- ▶ Em alguns pontos o fluxo pode divergir, mediante decisões baseadas em condições
- ▶ As decisões e uniões são apresentados como diamantes
- ▶ A decisão normalmente tem a condição próxima de si



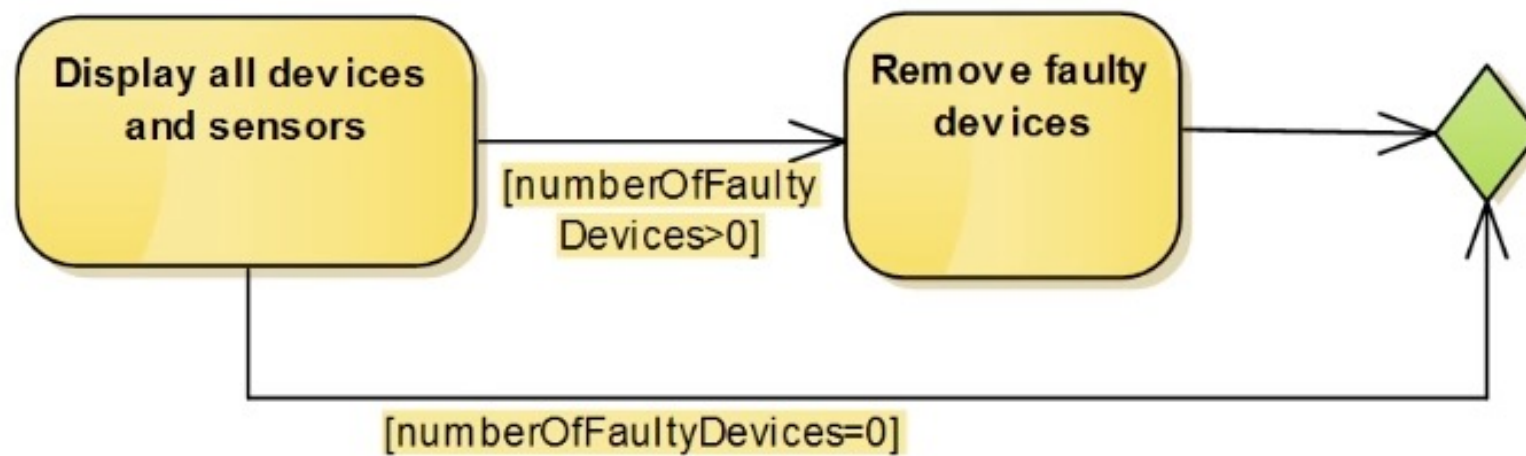
# Controlo de Fluxos

- Usar a chave “else”



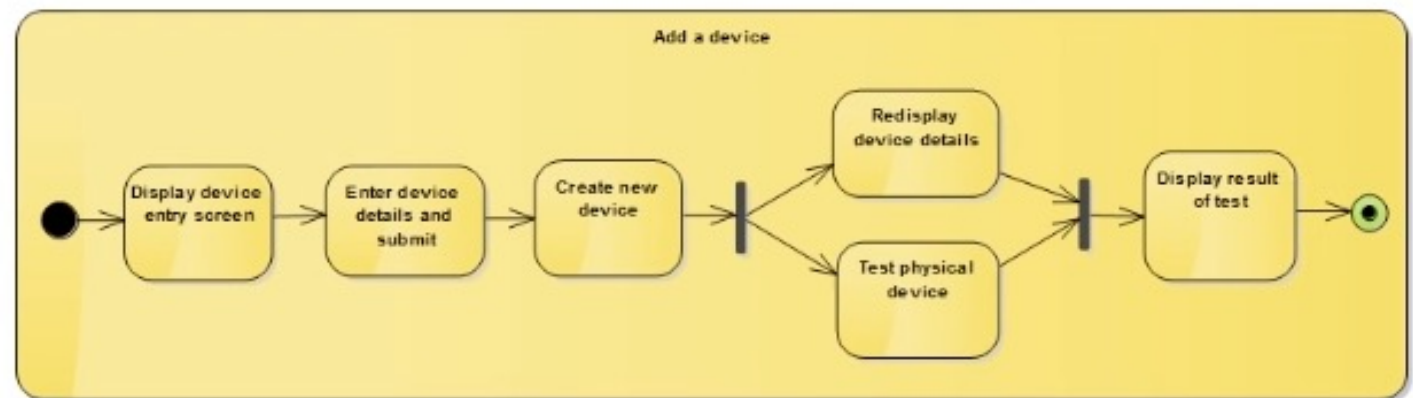
# Não usar nós de decisão

- ▶ Etiquetas na saída a partir da atividade ou ação
- ▶ Necessita de uma união algures
- ▶ Mais difícil de ler



# Fluxos Paralelos

- ▶ Fluxos paralelos
- ▶ Bifurcações e Uniões
- ▶ Sem utilizar bifurcações e uniões
- ▶ Token
- ▶ Nó final de fluxo

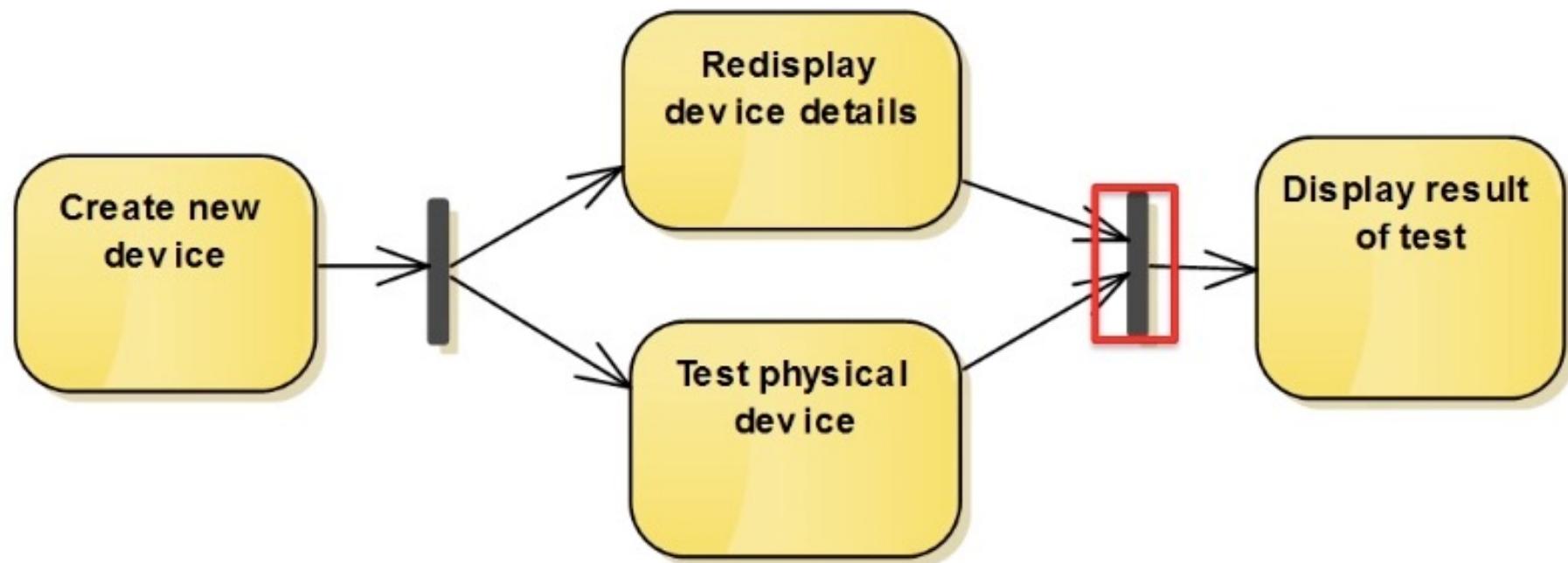


# Fluxos Paralelos

- ▶ Existem atividades que podem decorrer em paralelo
- ▶ Em sistemas de computadores as atividades podem decorrer em paralelo
- ▶ Estas são representadas em fluxos paralelos
- ▶ O fluxo bifurca no nó de bifurcação e une-se no nó de união
- ▶ Ambos são representados por barras no nosso modelo

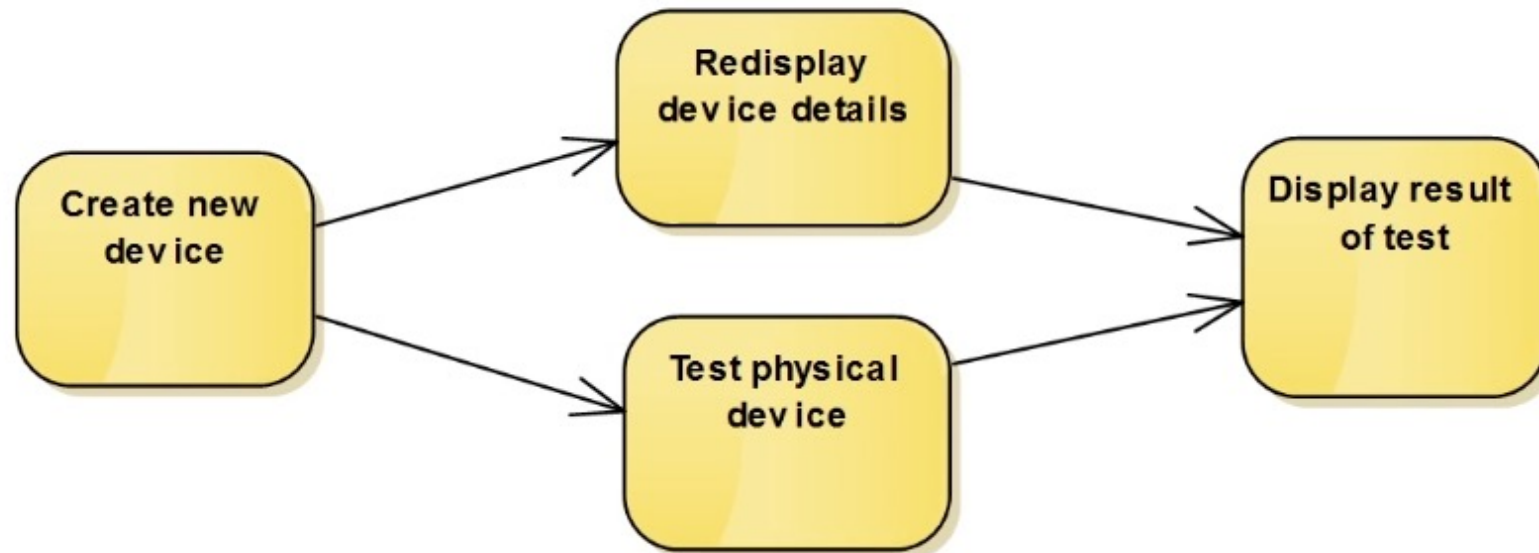


## Bifurcações e uniões

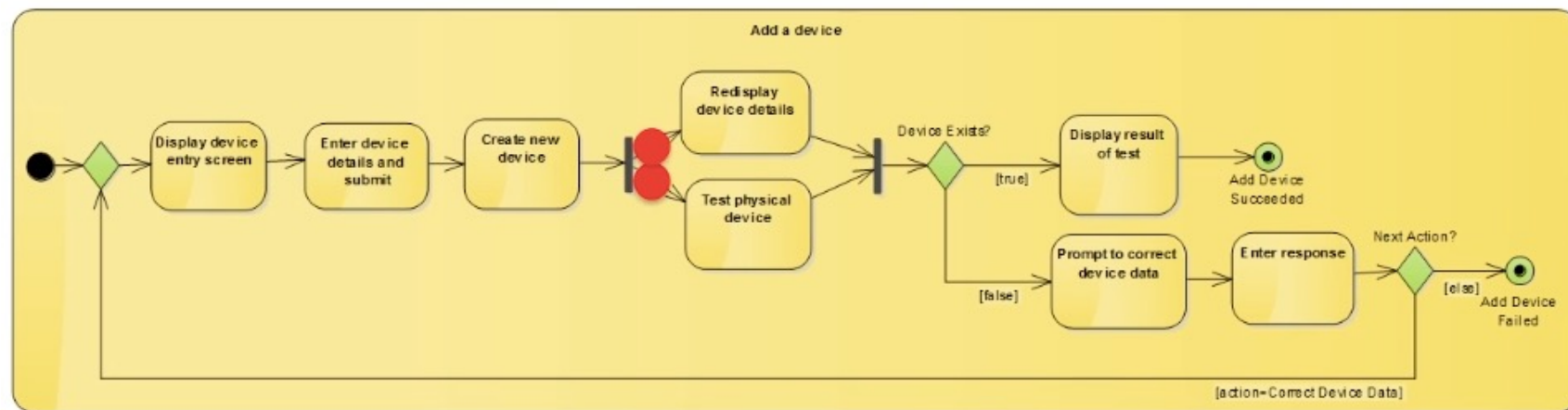
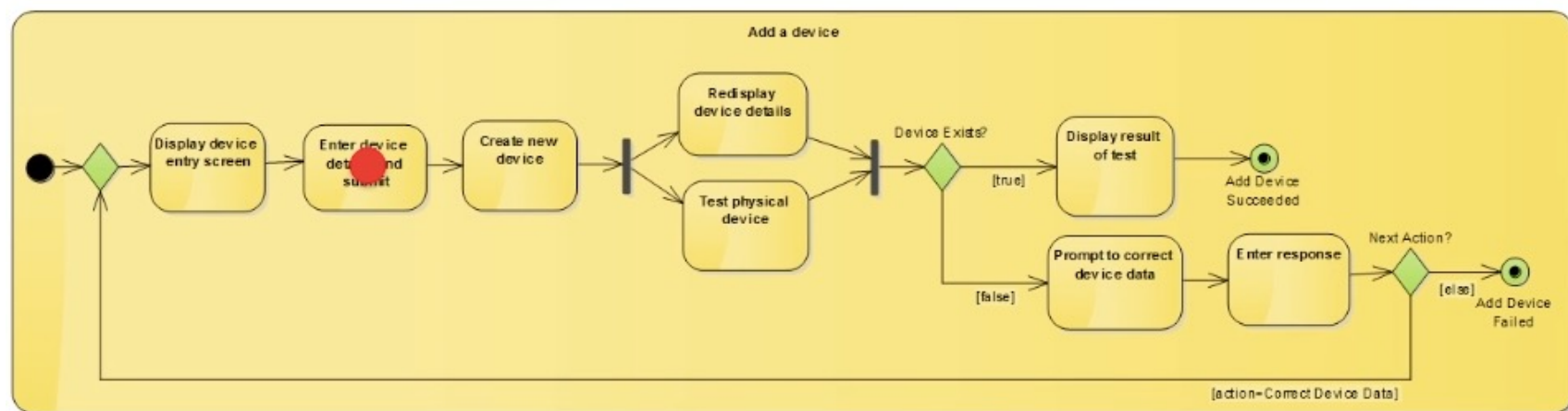


# Não utilizar bifurcações e uniões

- Bifurcações e uniões implícitas



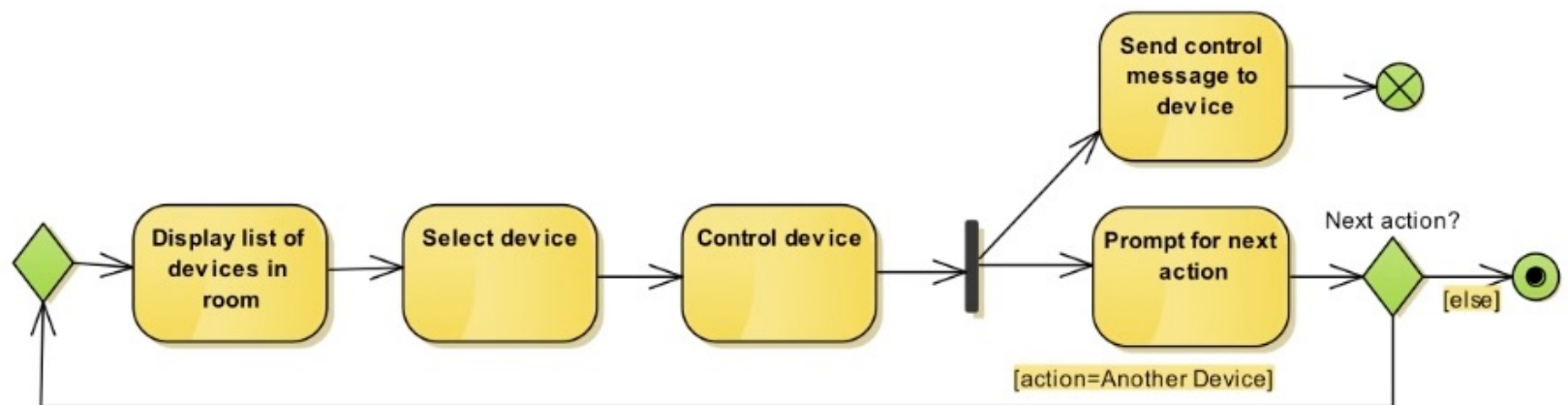
# Passagem do Token





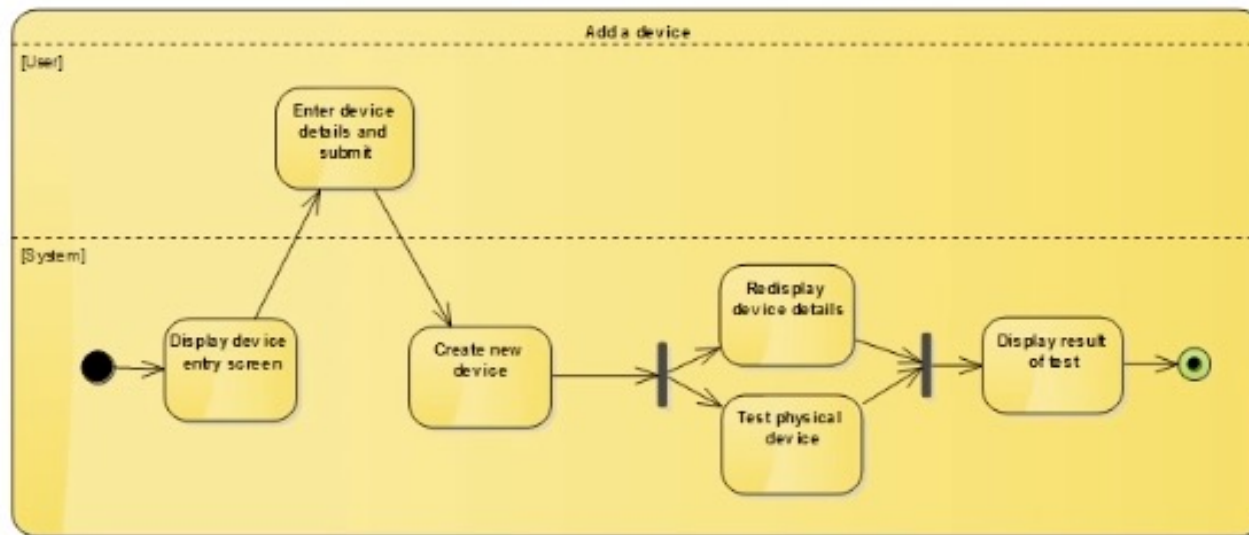
# Nó final de fluxo

- ▶ Existem cenários onde o fluxo chega ao fim sem terminar todas as atividades, isto acontece em atividades paralelas
- ▶ Nó final de fluxo
- ▶ É um círculo com um X



# Swimlanes

- ▶ Apresenta responsabilidades
- ▶ Divisão de um diagrama de atividades
- ▶ Swimlanes

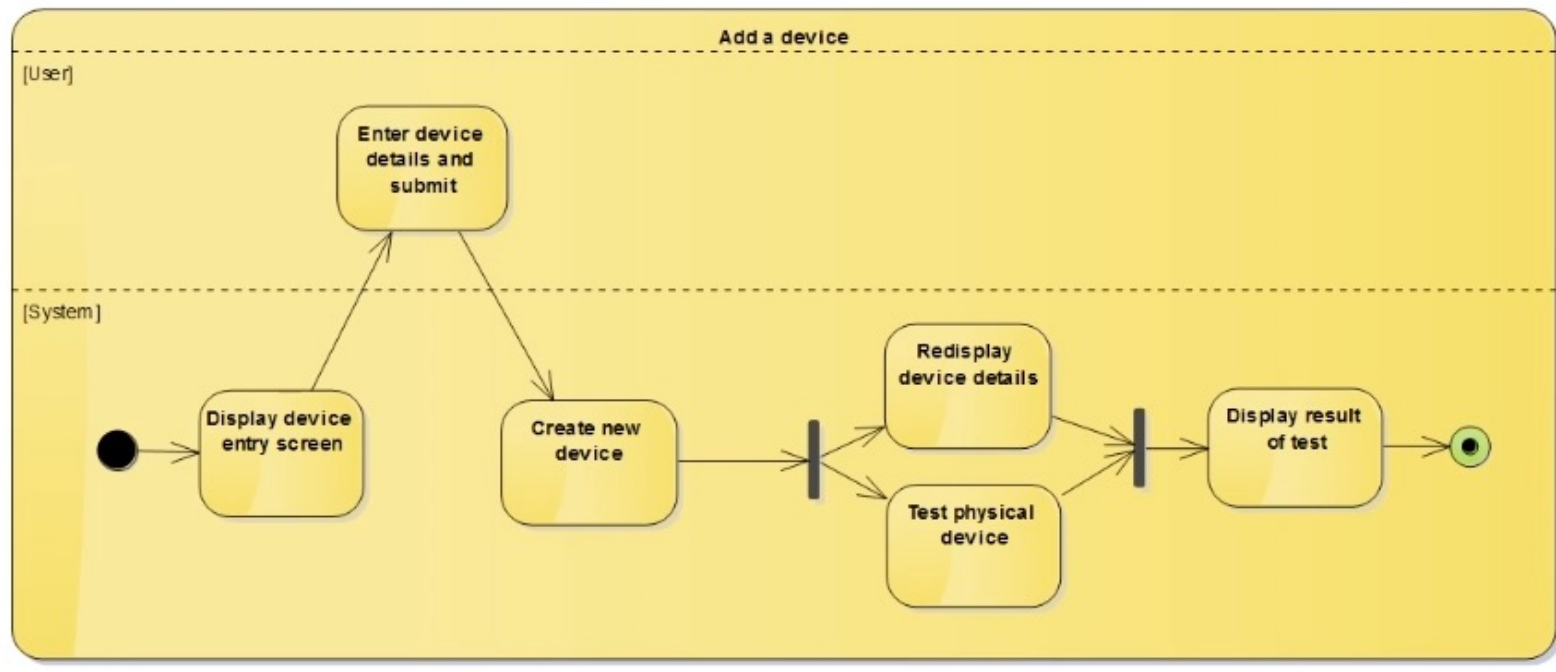


# Apresentação de responsabilidades

- ▶ Em processos de negócio, pessoas, papéis, departamentos ou organizações podem ser responsáveis por ações ou atividades
- ▶ Em sistemas de computadores, classes, sistemas ou componentes podem ser responsabilidades ações ou atividades
- ▶ Podemos demonstrar isso nos diagramas de atividades com Swimlanes

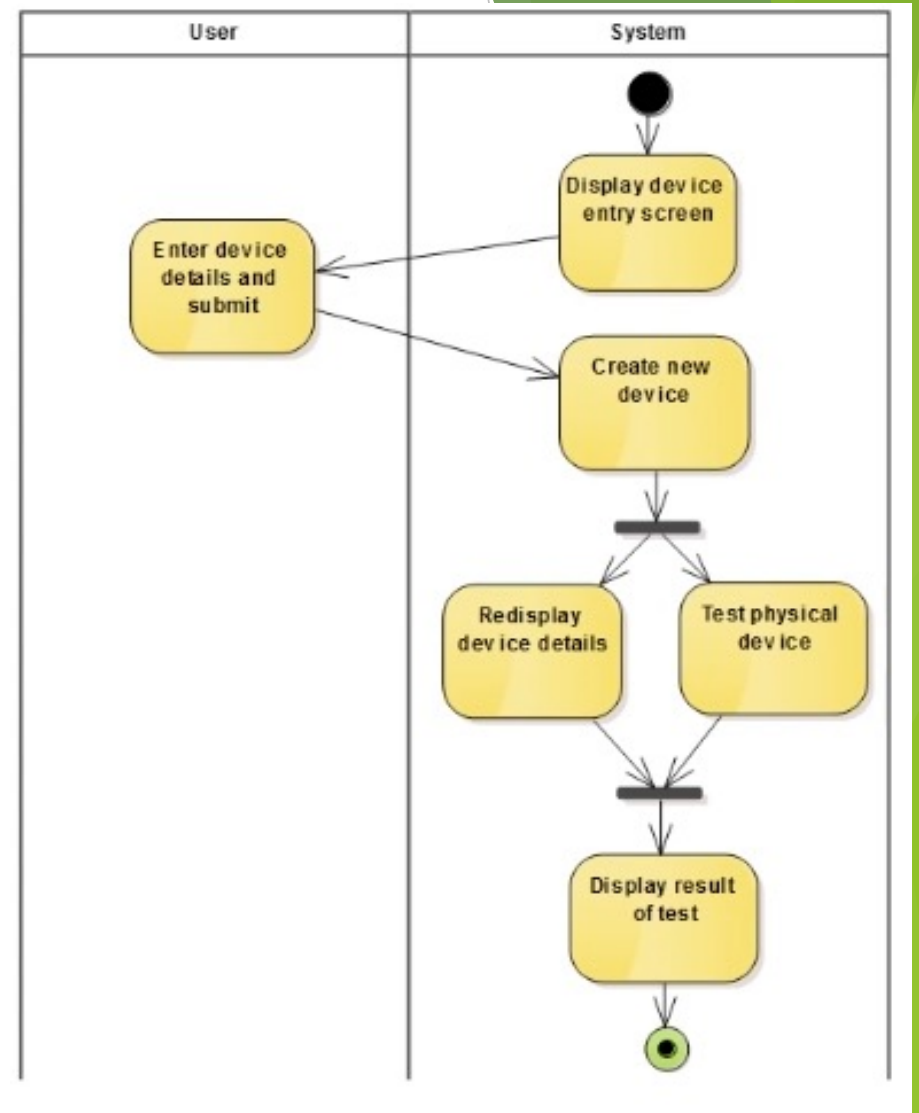


## Divisão de um diagrama



# Swimlanes

- São apresentadas na vertical ou horizontal
- O nome é apresentado no topo ou na esquerda



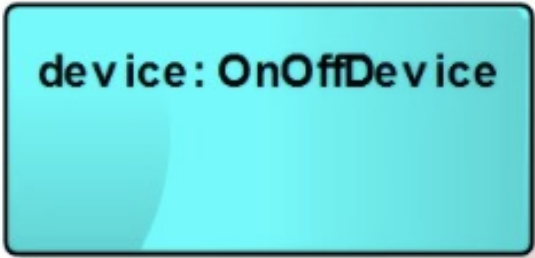
# Objetos e fluxos de objetos

- ▶ Representação de objetos em diagramas
- ▶ Fluxos de objetos
- ▶ Alfinetes em ações



# Representação de objetos

- ▶ Ações têm influência em objetos e isto pode ser representado em diagramas de atividades
- ▶ Representação:
  - ▶ `<name> : <classifier>`

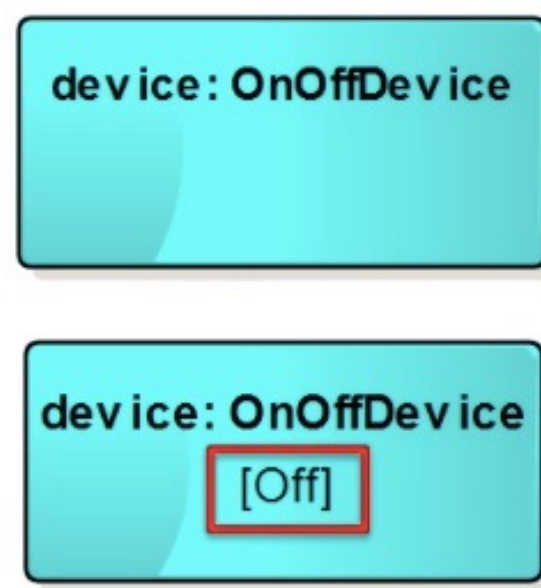


**device: OnOffDevice**

The image shows a light blue rectangular box with a black border and rounded corners. Inside the box, the text 'device: OnOffDevice' is written in a bold, black, sans-serif font. The box is positioned on the right side of the slide, overlapping a large, abstract green geometric shape that serves as a background element. A thin white line extends from the bottom right corner of the box towards the bottom right of the slide.

# Representação de objetos

- ▶ Ações têm influência em objetos e isto pode ser representado em diagramas de atividades
- ▶ Representação:
  - ▶ <name> : <classifier>
- ▶ O estado do objeto também pode ser representado



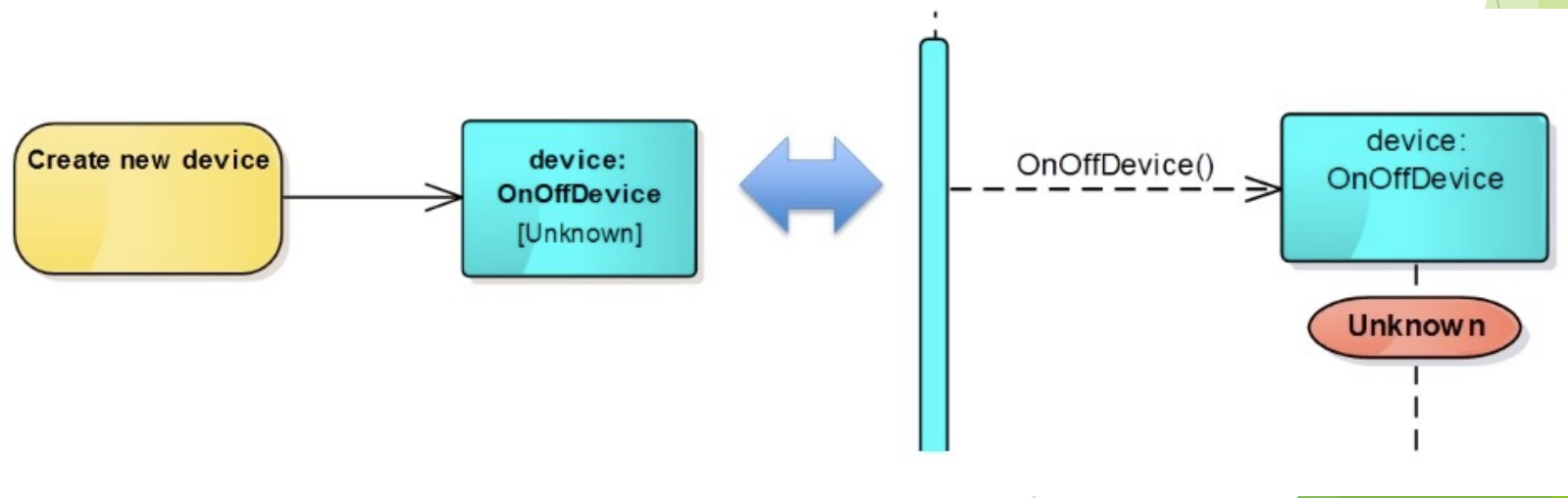
**device: OnOffDevice**

**device: OnOffDevice**  
[Off]



# Representação de objetos

- Atualizações de estados têm o mesmo efeito que as interações



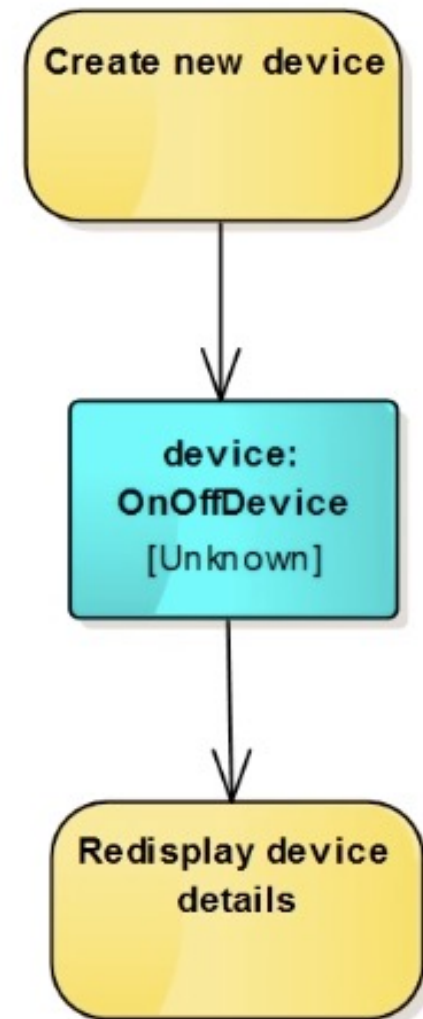
# Fluxo de objetos

- ▶ Fluxos de objetos ligam estes a atividades e ações
- ▶ Também liga os objetos a outros elementos como buffers, datastores, que armazenam objetos de forma temporária ou permanente;



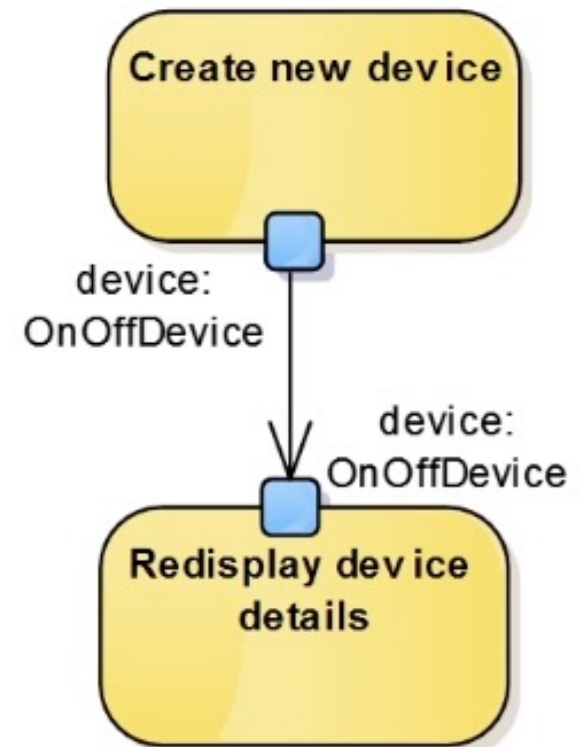
# Fluxo de objetos

- ▶ O fluxo de um diagrama de atividades pode passar por objetos
- ▶ São representados pela mesma notação



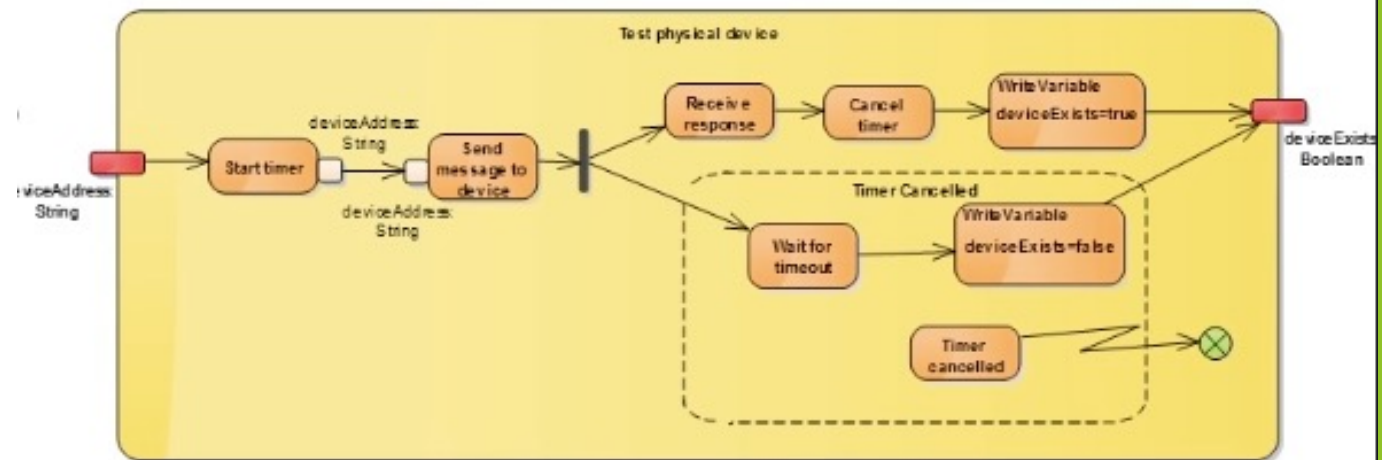
# PINGS

- ▶ Quer um objeto seja de origem ou destino, representamos via um quadrado pequeno, no fluxo de objetos;
- ▶ O nome do objeto e o classificador são representados junto do mesmo, separados por :



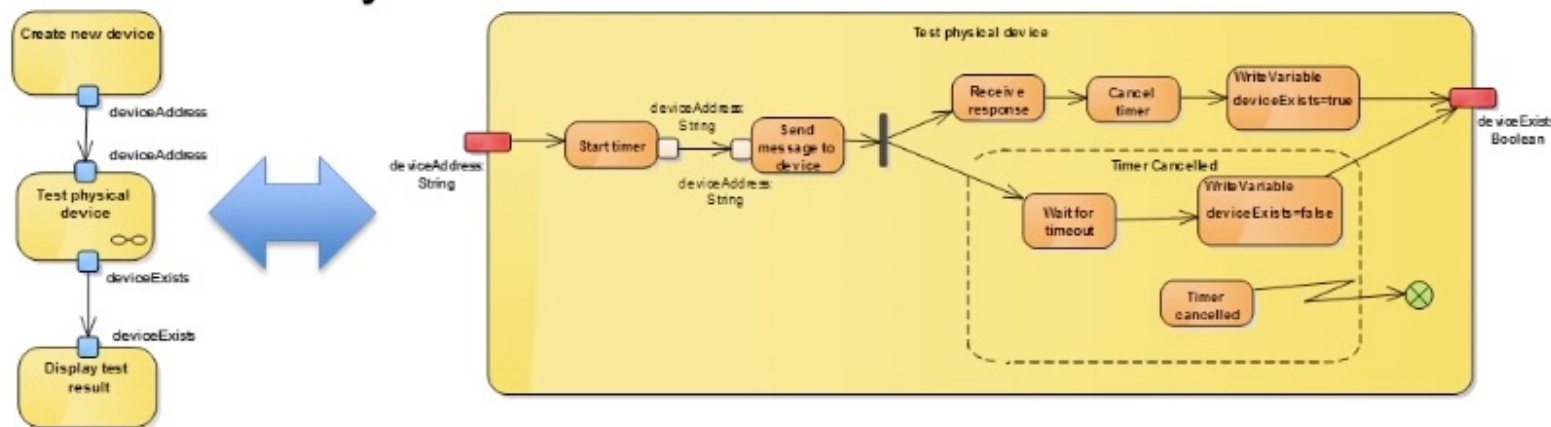
# Estruturação de atividades

- Decomposição de atividades
- Parâmetros de atividades
- Ações em atividades
- Regiões



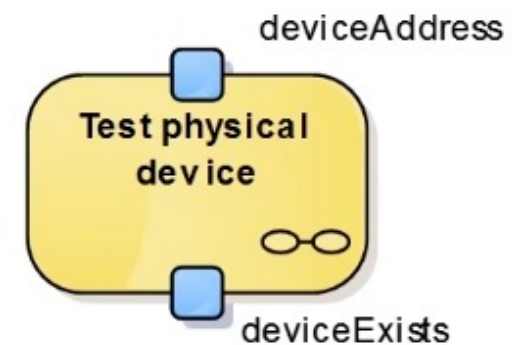
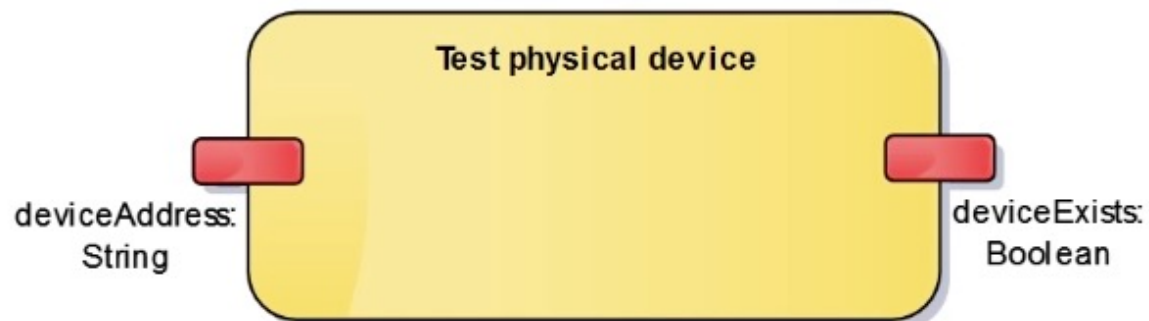
# Decomposição de atividades

- Podemos apresentar uma atividade em formato de alto nível, num diagrama de atividades;
- Detalhar a mesma atividade em outro diagrama de atividades;



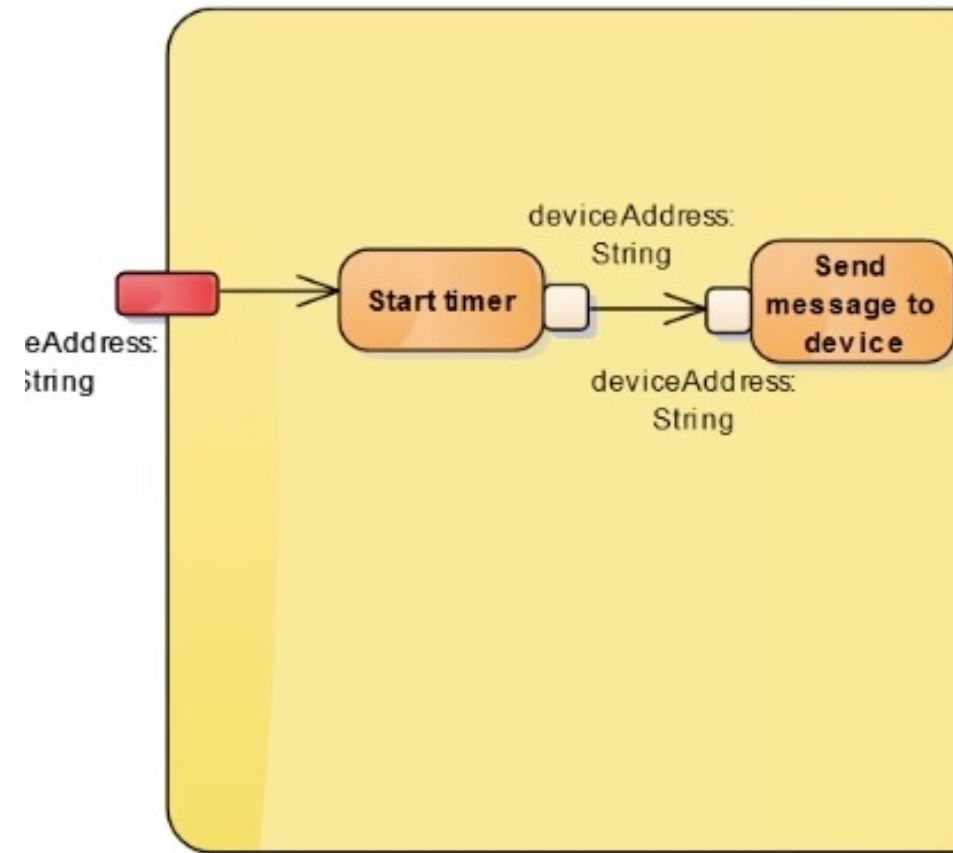
# Parâmetros de atividades

- ▶ Aparecem na extremidade da atividade;
- ▶ Representam os valores de I/O da atividade;



# Ações em atividades

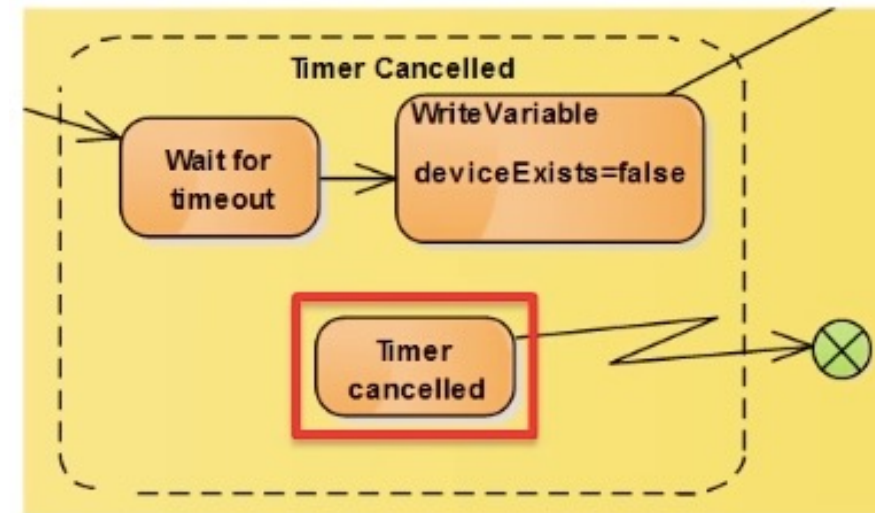
- O nível mais baixo que precisamos de representar, são as ações de atividades





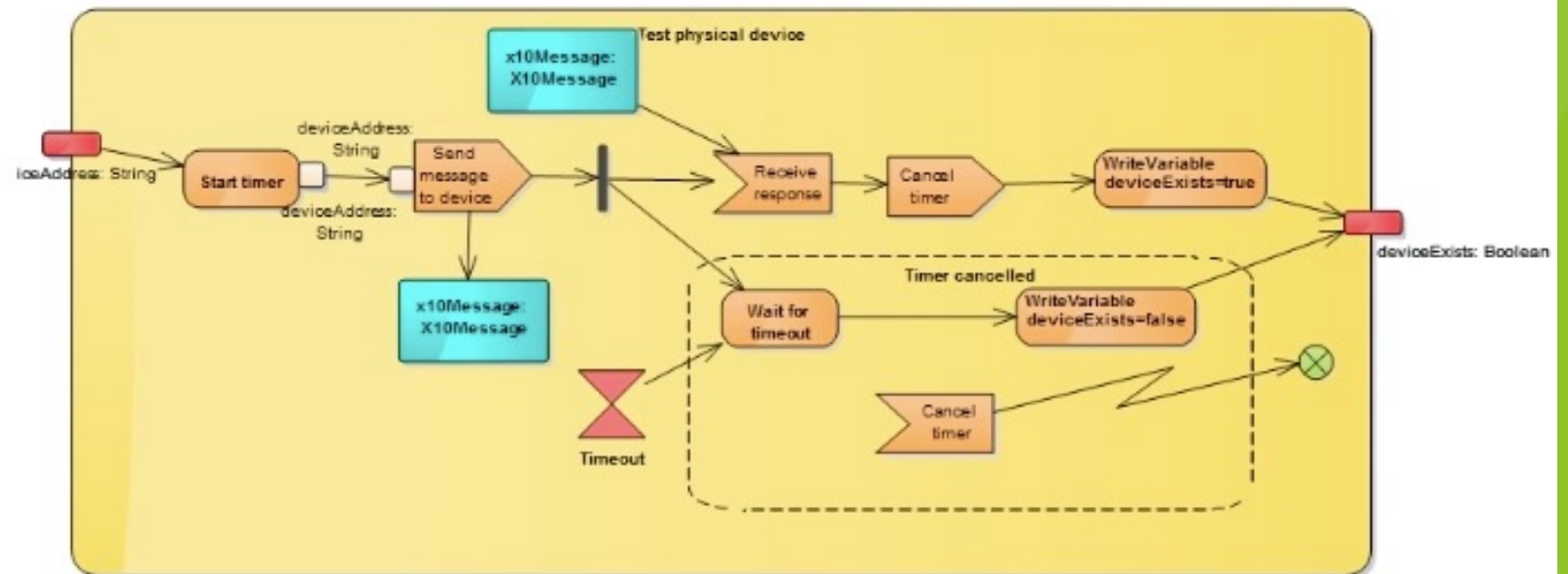
# Regiões

- ▶ Regiões “interrompidas” agrupam ações que podem ser terminadas por um evento
- ▶ São representadas por um retângulo de cantos redondos com linha tracejada
- ▶ Deve incluir o evento/ação que pode causar a interrupção
- ▶ O fluxo de saída é definido por uma seta zigzag



# Enviar, receber e temporizadores

- ▶ Ações de enviar e receber
- ▶ Sinais
- ▶ Temporizadores



# Ações de enviar e receber

- ▶ Enviar → `SendSignalAction`
- ▶ Utilizado para demonstrar o envio de sinais
- ▶ Receber → `AcceptEventAction`
- ▶ Utilizado para demonstrar a recepção de sinais



Send  
message  
to device


Receive  
response

# Sinais

- ▶ Enviar → `SendSignalAction`
- ▶ Receber → `AcceptEventAction`
- ▶ Objeto em que o seu classificar é um sinal



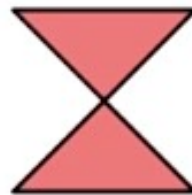
**x10Message:**  
**X10Message**



**«signal»**  
**X10Message**

# Temporizadores

- ▶ Timer → AcceptTimeEventAction
- ▶ Utilizado para demonstrar eventos despoletados por temporizadores



**Timeout**

