

# Metodologia de modelagem

## Etapa 3

**Prof. Murillo G. Carneiro**  
**FACOM/UFU**

Material baseado nos slides disponibilizados pelo Prof. Ricardo Pereira e Silva (UFSC)

# Objetivo

- Apresentar a terceira etapa da metodologia de modelagem
  - Etapa 3 – Refinamento de casos de uso

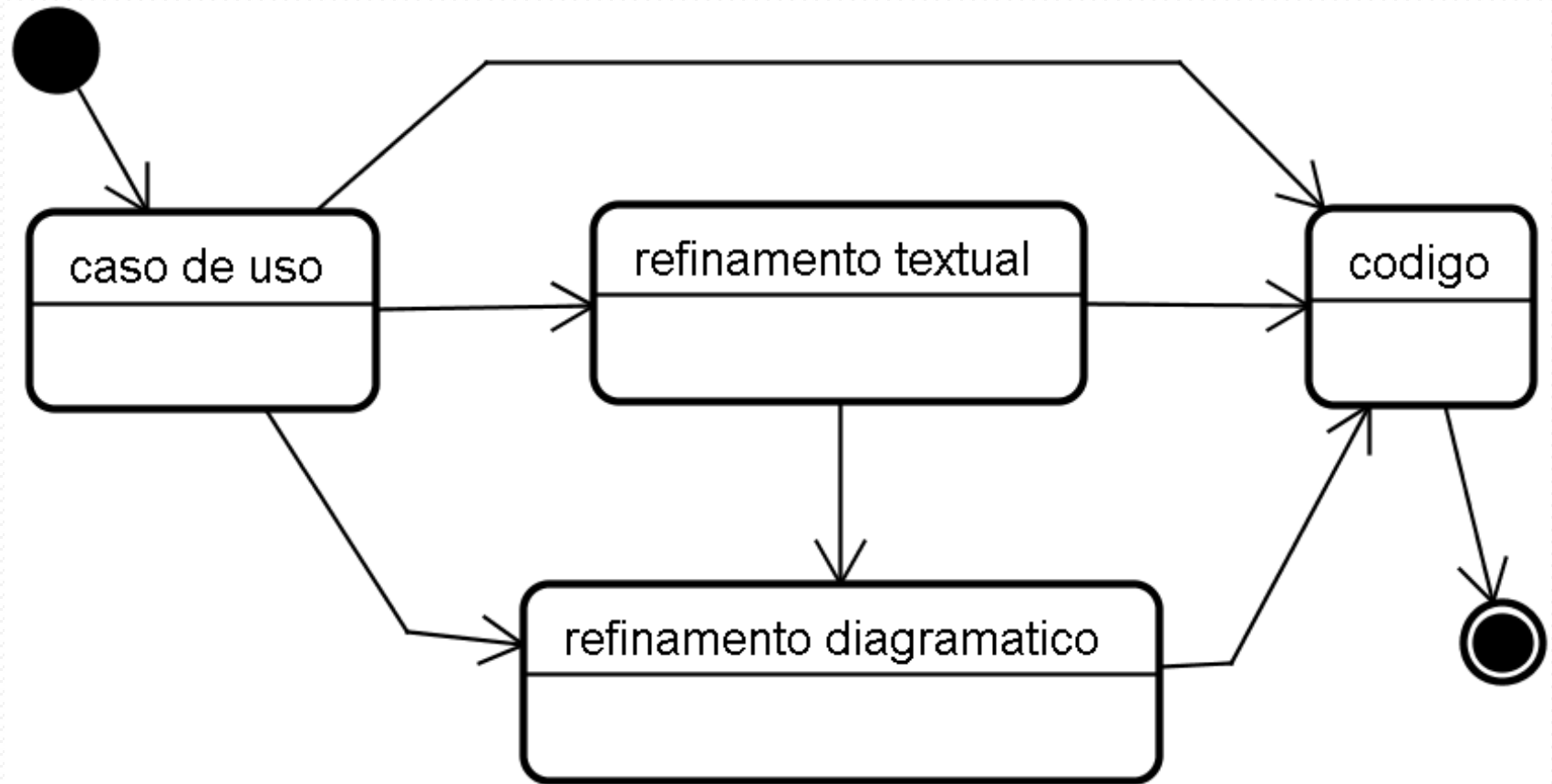
# Diagramas UML usados

- Diagrama de atividades
- Diagrama de comunicação
- Diagrama de sequência

## Etapa 3 – Refinamento de casos de uso

- Casos de uso estabelecem O QUE o software faz
- O refinamento de casos de uso vai estabelecer COMO faz
  - Apresentação das alternativas de refinamento
  - Refinamento com diagrama de atividades
  - Refinamento com diagrama de comunicação
  - Refinamento com diagrama de sequência

# Alternativas para refinamiento de casos de uso



## Primeira possibilidade → do caso de uso, direto ao código

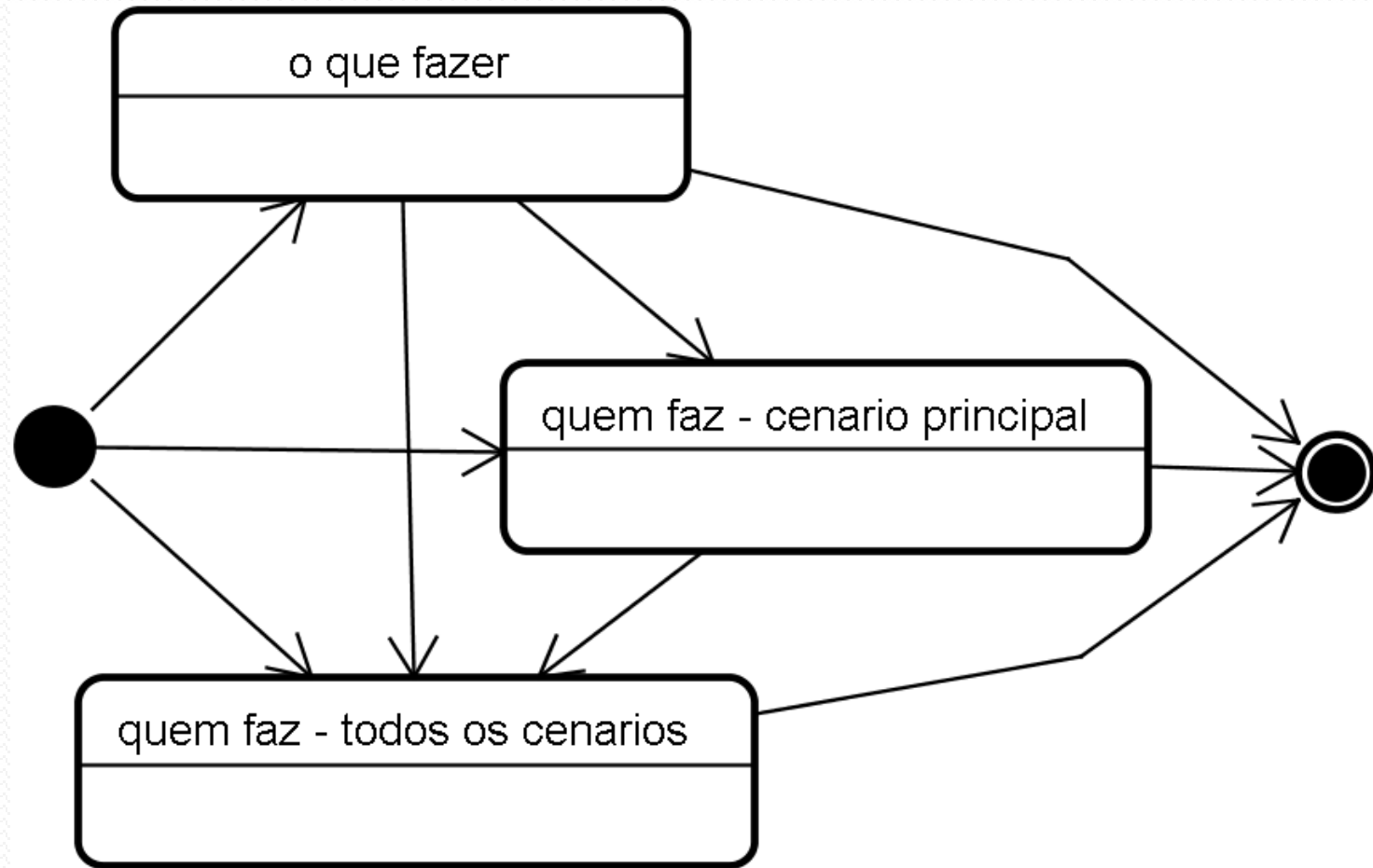
- Só é viável para sistemas de baixa complexidade
- Não faz uso da alternativa de refinamento gradual, possibilitada por UML

# Segunda possibilidade → refinamento textual

- Não faz uso do potencial expressivo de UML
- Texto com seções predefinidas (estruturado) ou não



# Terceira possibilidade → refinamento diagramático

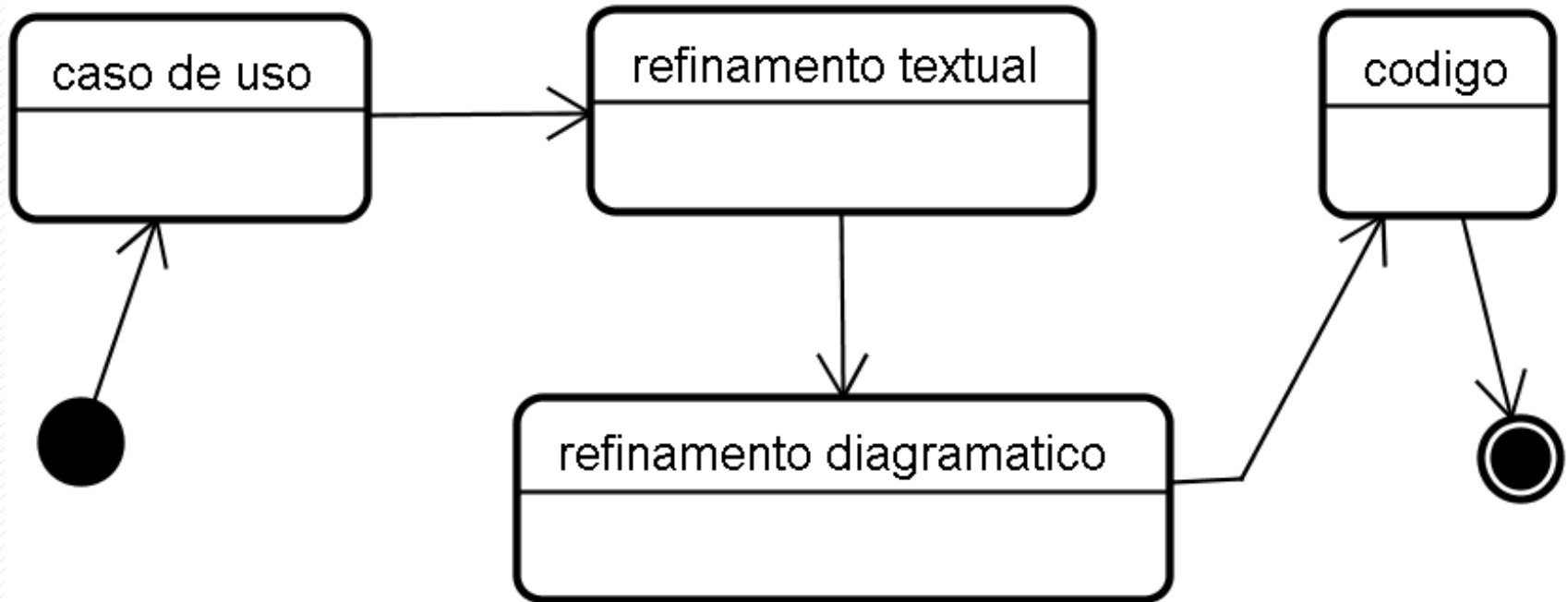




# Terceira possibilidade → refinamento diagramático

- Sem adotar descrição textual
  - Apenas com os diagramas de UML
  - Modelagem dinâmica de sistema
- Três estágios
  - Possível passar por um, por dois ou pelos três
  - Depende da complexidade do caso de uso tratado

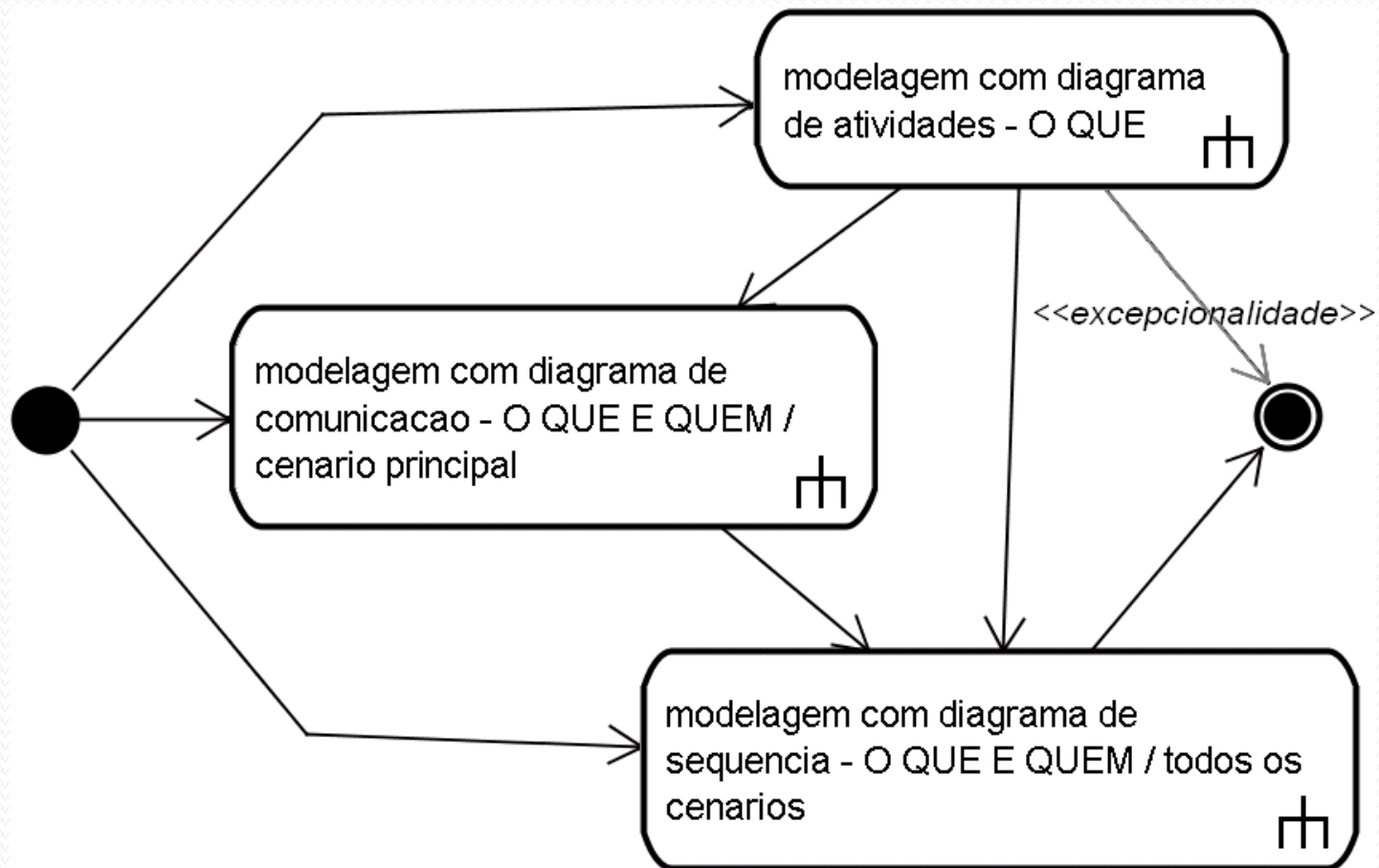
# Quarta possibilidade → refinamentos textual e diagramático



# O processo de refinamento dos casos de uso

- Adotada a terceira possibilidade → refinamento diagramático
  - O que fazer → diagrama de atividades
  - Quem faz (cenário principal) → diagrama de comunicação
  - Quem faz (todos os cenários) → diagrama de sequência

# O processo de refinamento dos casos de uso

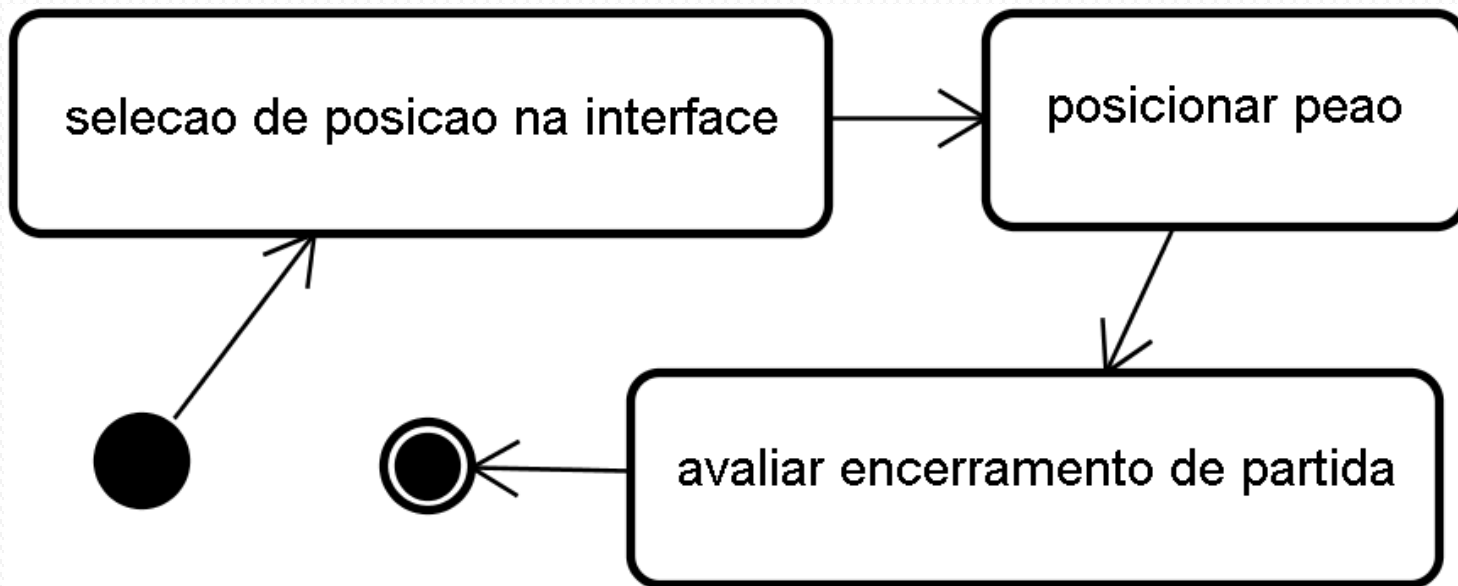


# O que fazer: modelagem com diagrama de atividades

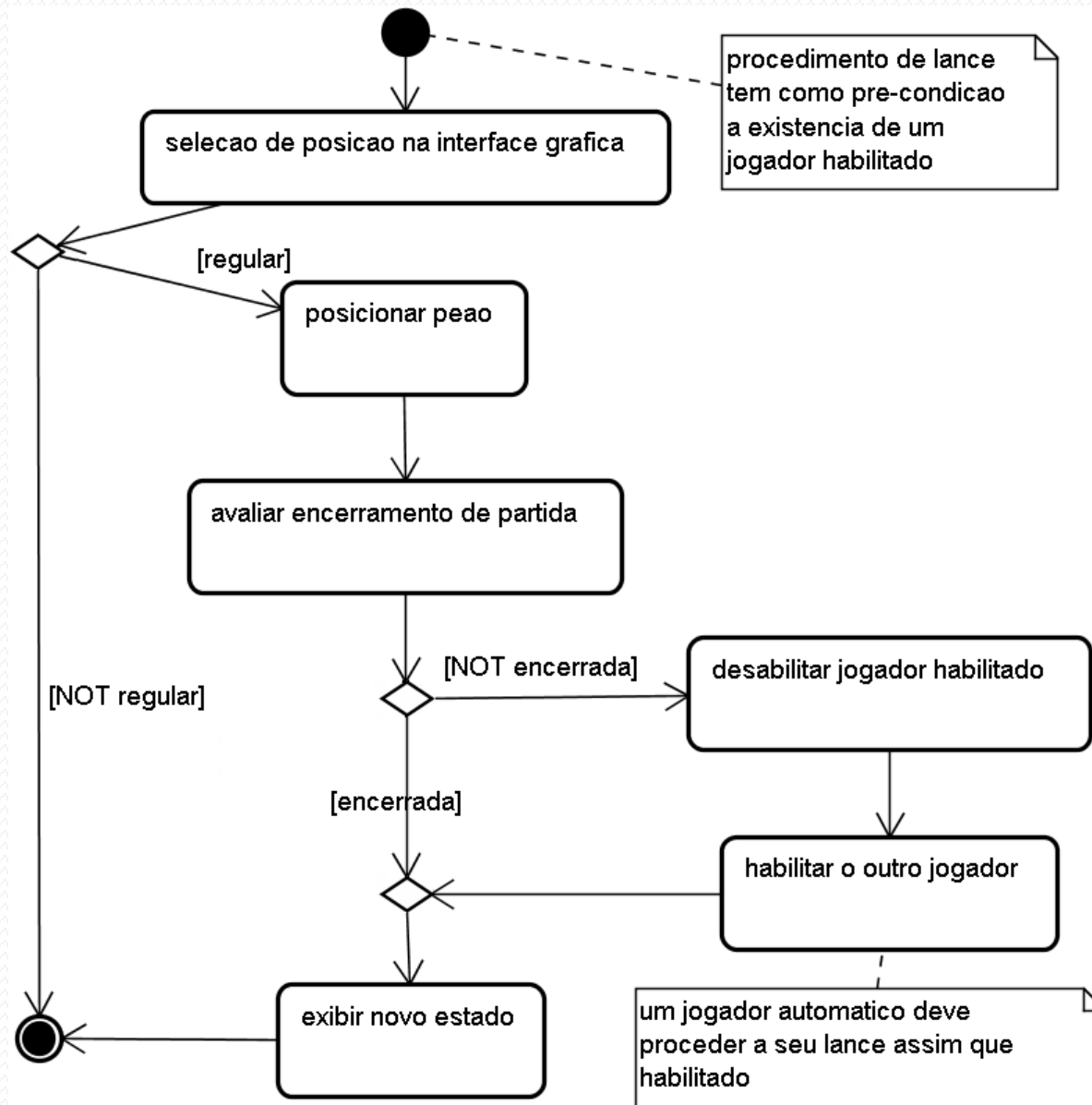
- Primeira modelagem em diagrama de atividades
  - Foco no cenário principal (quando “tudo dá certo”)
  - Abstrair as outras possibilidades (outros cenários)

# O que fazer: modelagem com diagrama de atividades

- Exemplo Jogo-da-velha (“procedimento de lance”)



# Detalhamento da primeira modelagem em diagrama de atividades (procedimento de lance, do Jogo-da-velha)



# Exemplo: questões que surgem no aprofundamento da modelagem

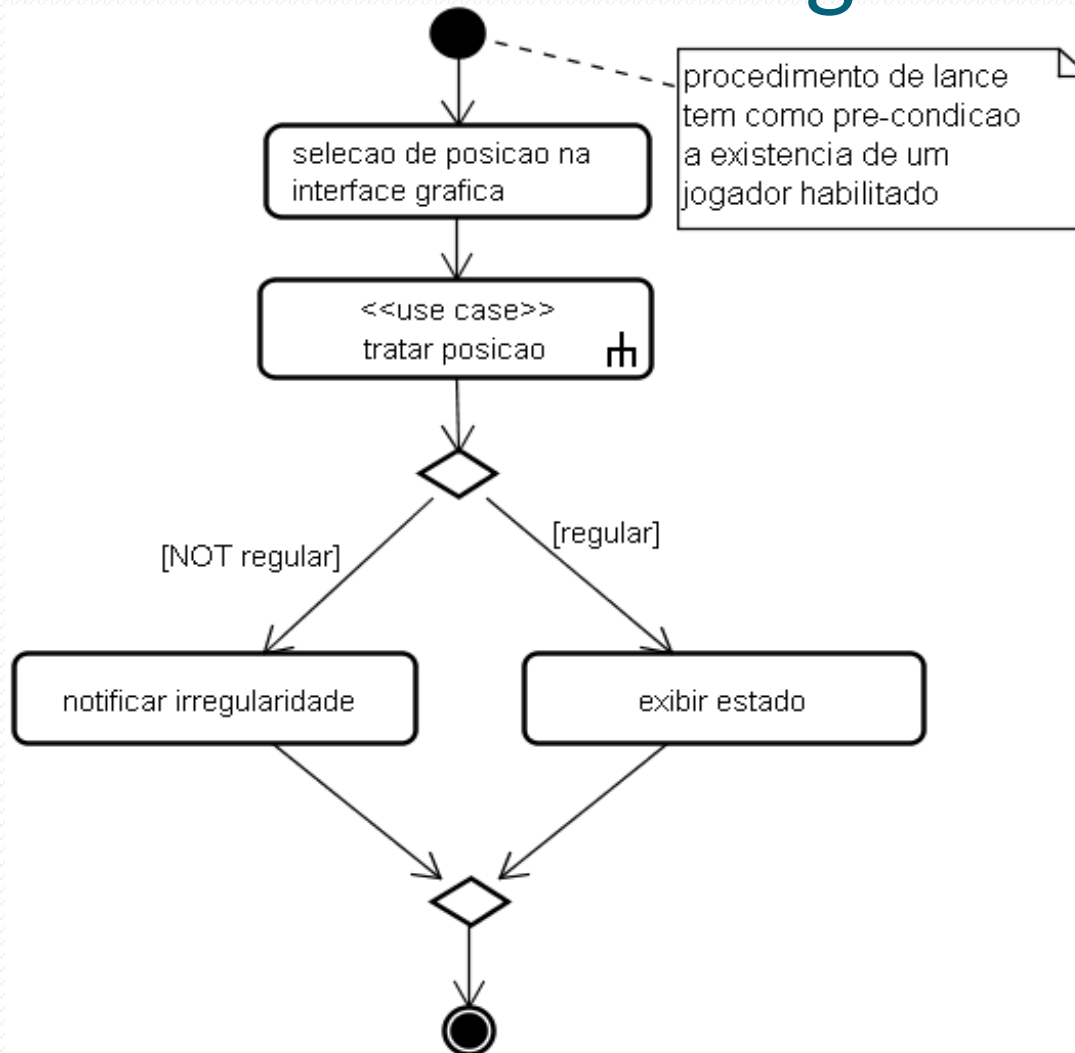
- O caso de uso prevê a ação de procedimento de lance sobre a interface, mas como a instância de JogadorAutomatico atua?
  - Decisão de projeto → retorna seu lance quando habilitado (procedimento “habilitar o outro jogador”)
- Precisa avaliar a precondição?



# Exemplo: tratamento de problemas identificados na modelagem

- Caso “habilitar jogador” habilite instância de JogadorAutomatico
  - Ele definirá uma opção de lance (não como ação na interface )
  - Após, é preciso “*posicionar peao*” e os procedimentos posteriores
- Conclusão: da forma como está modelado, parte de “procedimento de lance” pode ser repetido
  - Decisão de projeto → transformar a repetição em novo caso de uso

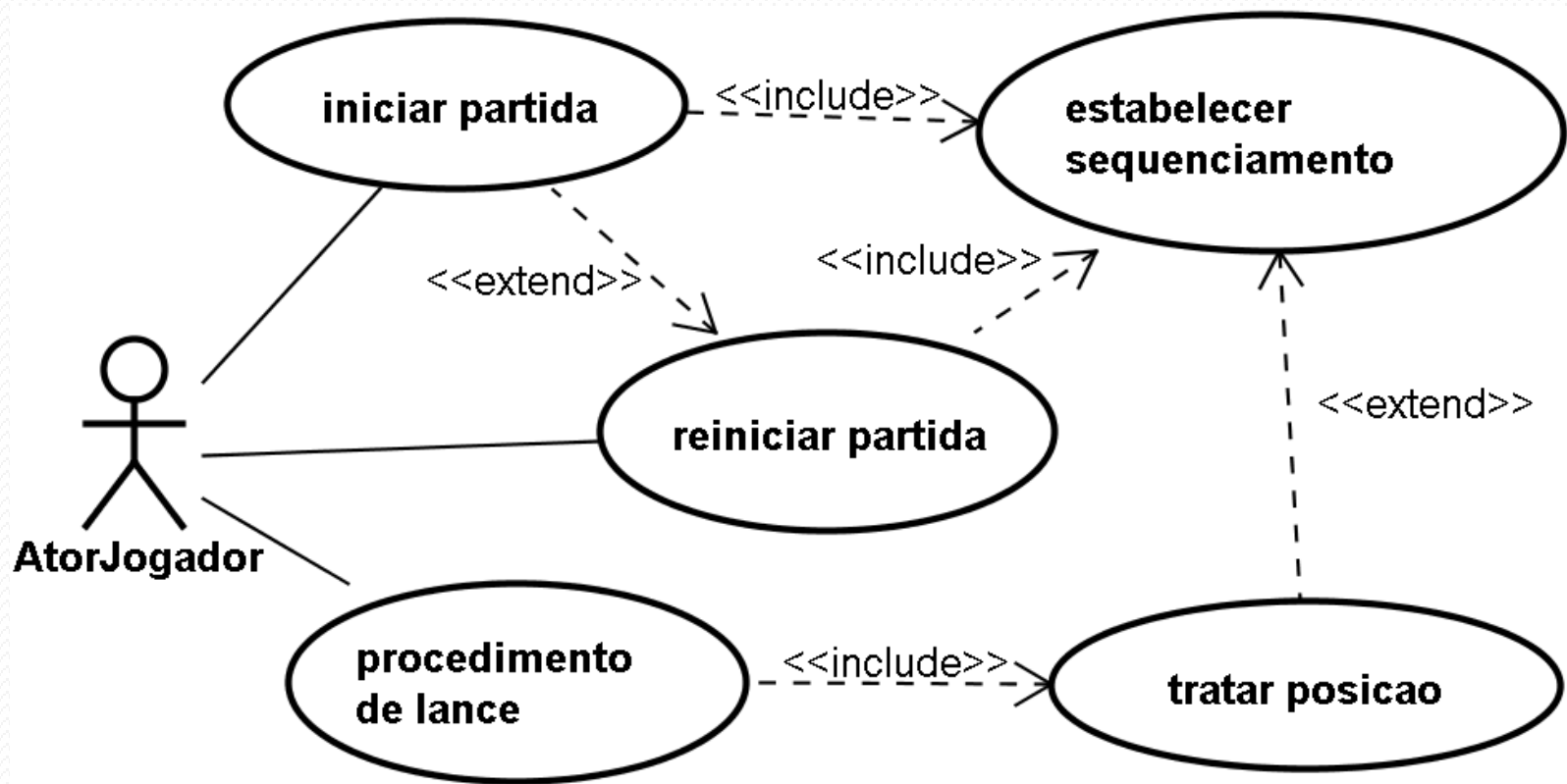
# Exemplo: tratamento de problemas identificados na modelagem



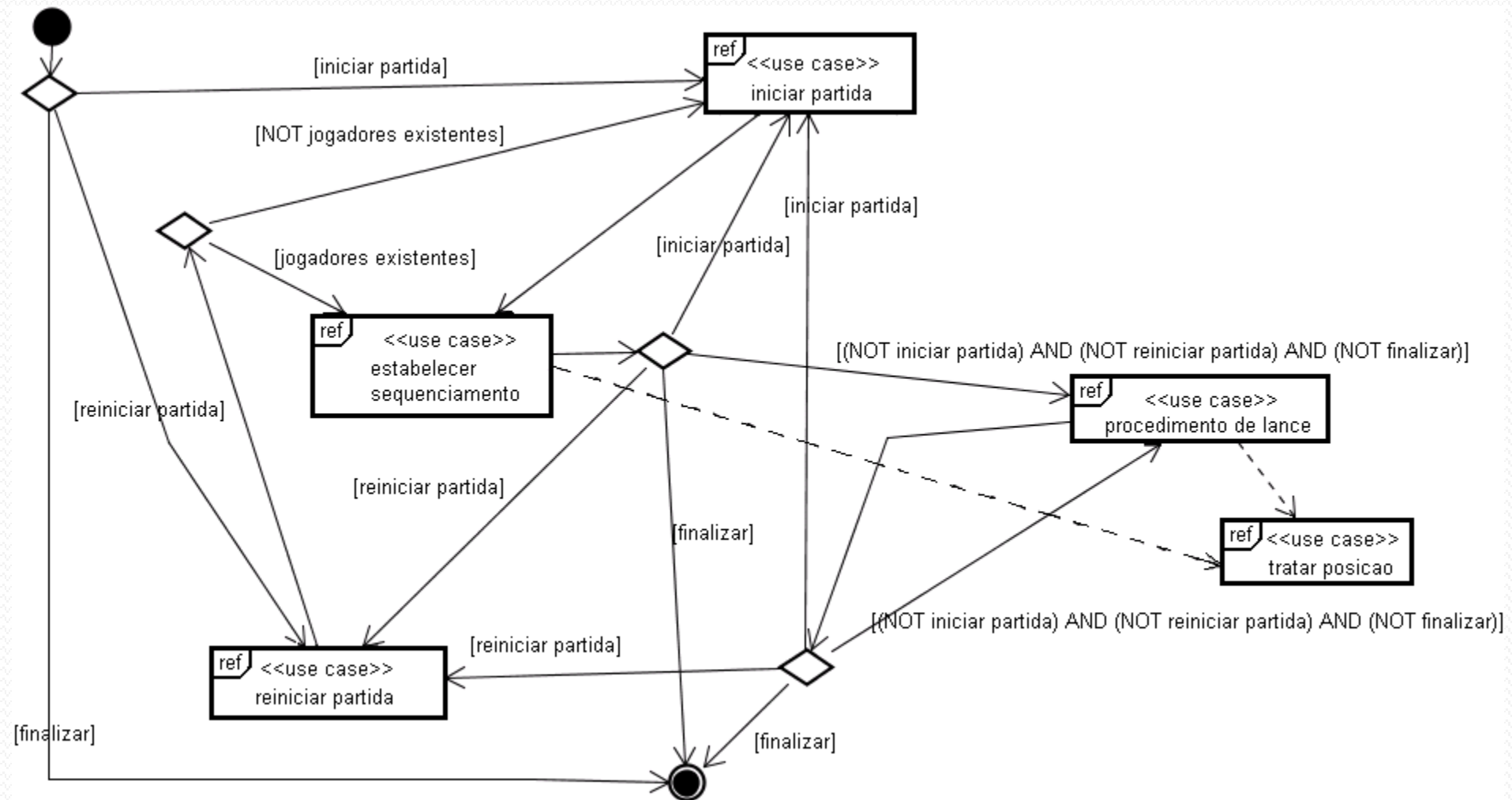
## Casos de uso podem influenciar outros casos de uso (exemplo)

- O caso de uso “iniciar partida” envolve a definição da ordem dos jogadores e o caso de uso “reiniciar partida”, também
- Decisão de projeto → procedimento que se repete nos dois casos de uso será sacado e originará um novo caso de uso: “estabelecer sequenciamento”

# Refinamento afeta a modelagem de casos de uso (exemplo)



# Refinamento afeta o sequenciamento dos casos de uso (exemplo)



# Sobre a modelagem em diagrama de casos de uso

- É comum que no primeiro esforço de modelagem (etapa 1) apenas associações entre ator e caso de uso sejam identificadas
- Associações entre casos de uso costumam ser identificadas neste estágio do processo

# Ênfase a quem faz (cenário principal): modelagem com diagrama de comunicação

- Voltado a modelar apenas o cenário principal
  - Quando a meta do caso de uso é atingida
  - Ex.: “procedimento de lance” → um lance regular

# Ênfase a quem faz (cenário principal): modelagem com diagrama de comunicação

- Etapa intermediária entre diagrama de atividades e diagrama de sequência
- Recomendável apenas quando
  - O caso de uso está associado a muitos cenários
  - O cenário principal e outros cenários apresentam alguma complexidade
  - Julgamento da complexidade → decisão de projeto



A identificação dos métodos durante a construção do diagrama de comunicação

***Responsabilidade***



***Atribuição da responsabilidade***



***Identificação de método(s)***

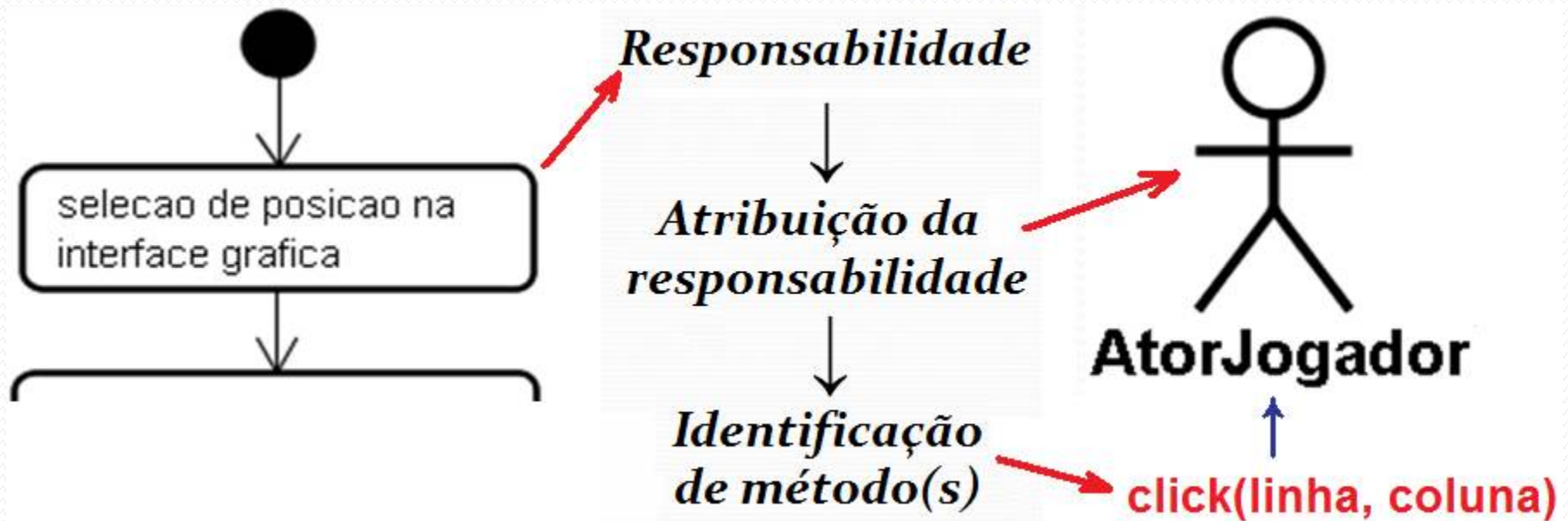
# A identificação dos métodos durante a construção do diagrama de comunicação

1. Identificar uma responsabilidade expressa em um nodo do diagrama de atividades
2. Definir que elemento (instância de que classe) é o “dono” dessa responsabilidade
  - Um ou mais donos
3. Associar à classe correspondente um método capaz de cumprir a tal responsabilidade

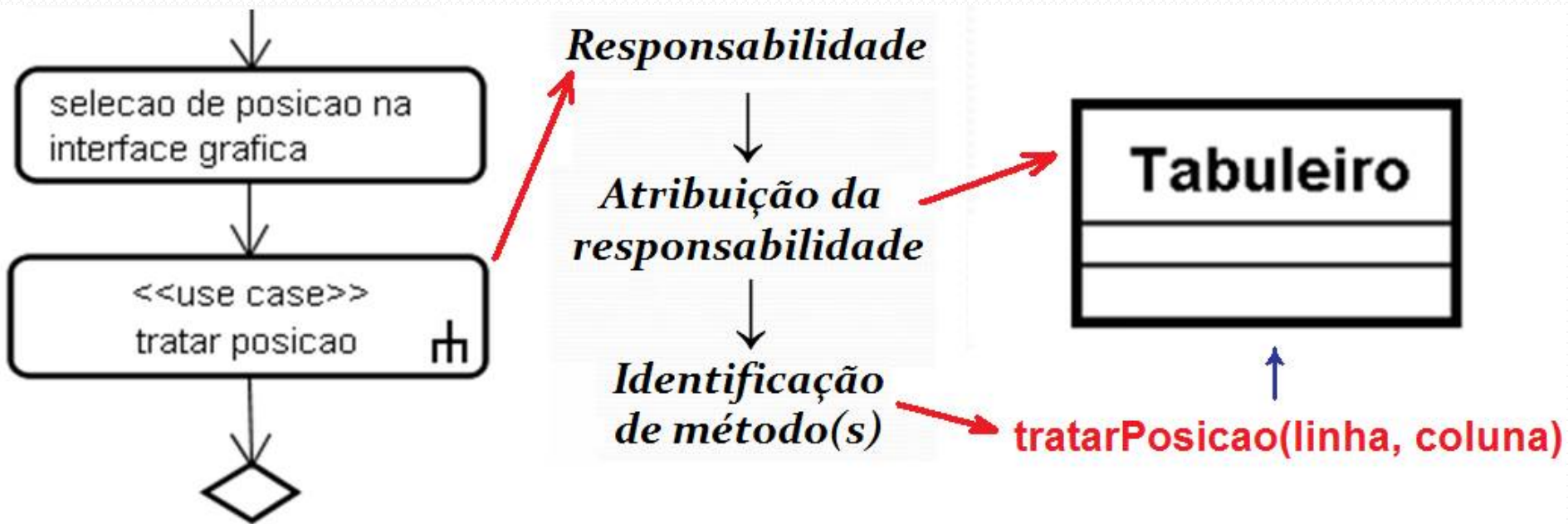
## Exemplo: “procedimento de lance” (JV)

- Questão ainda pendente
  - Precisa avaliar a precondição no escopo do caso de uso?
    - Haver um jogador habilitado
  - Conclusão: não, pois “procedimento de lance” ocorre apenas após “iniciar partida” ou “reiniciar partida”, que habilitam um dos jogadores

# Exemplo: “procedimento de lance” (JV)

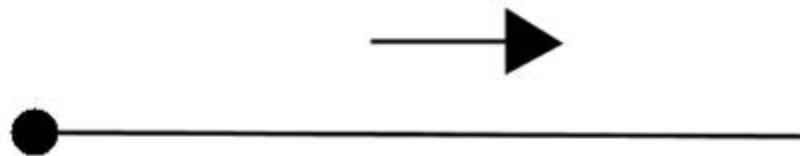


# Exemplo: “procedimento de lance” (JV)



# Exemplo: “procedimento de lance” (JV)

1: click(linha:int, coluna:int) : void



: AtorJogador

1.1.1: tratarPosicao(linha:int, coluna:int) : void



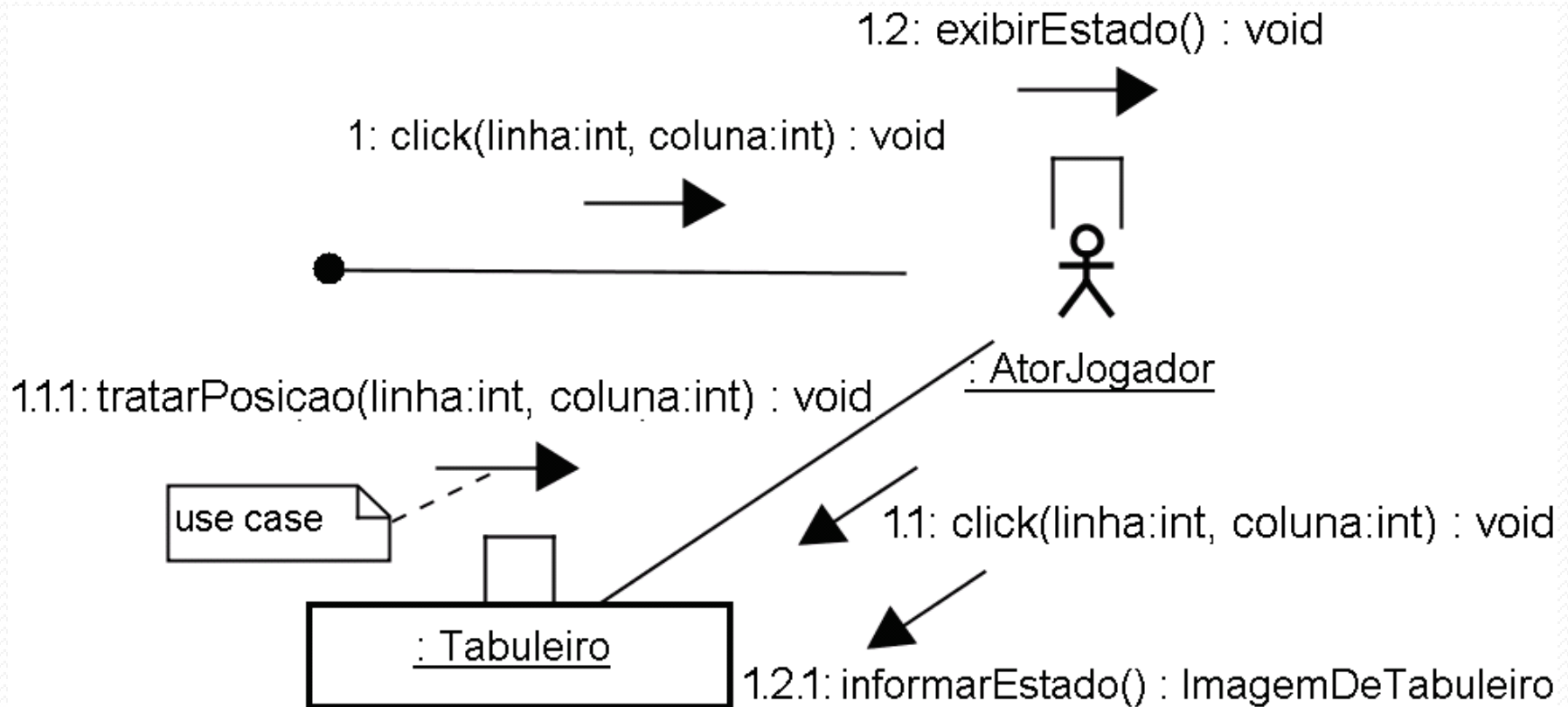
método para  
passagem de  
controle

: Tabuleiro

1.1: click(linha:int, coluna:int) : void

# Exemplo: “procedimento de lance” (JV)

## – final



## Ênfase a quem faz (todos os cenários): modelagem com diagrama de sequência

- Etapa final do refinamento de casos de uso
- Considera todos os cenários e identifica todos os métodos envolvidos
  - Além dos já identificados na modelagem do cenário principal com diagrama de comunicação (se tiver sido elaborado)



A identificação dos métodos durante a construção do diagrama de sequência

***Responsabilidade***

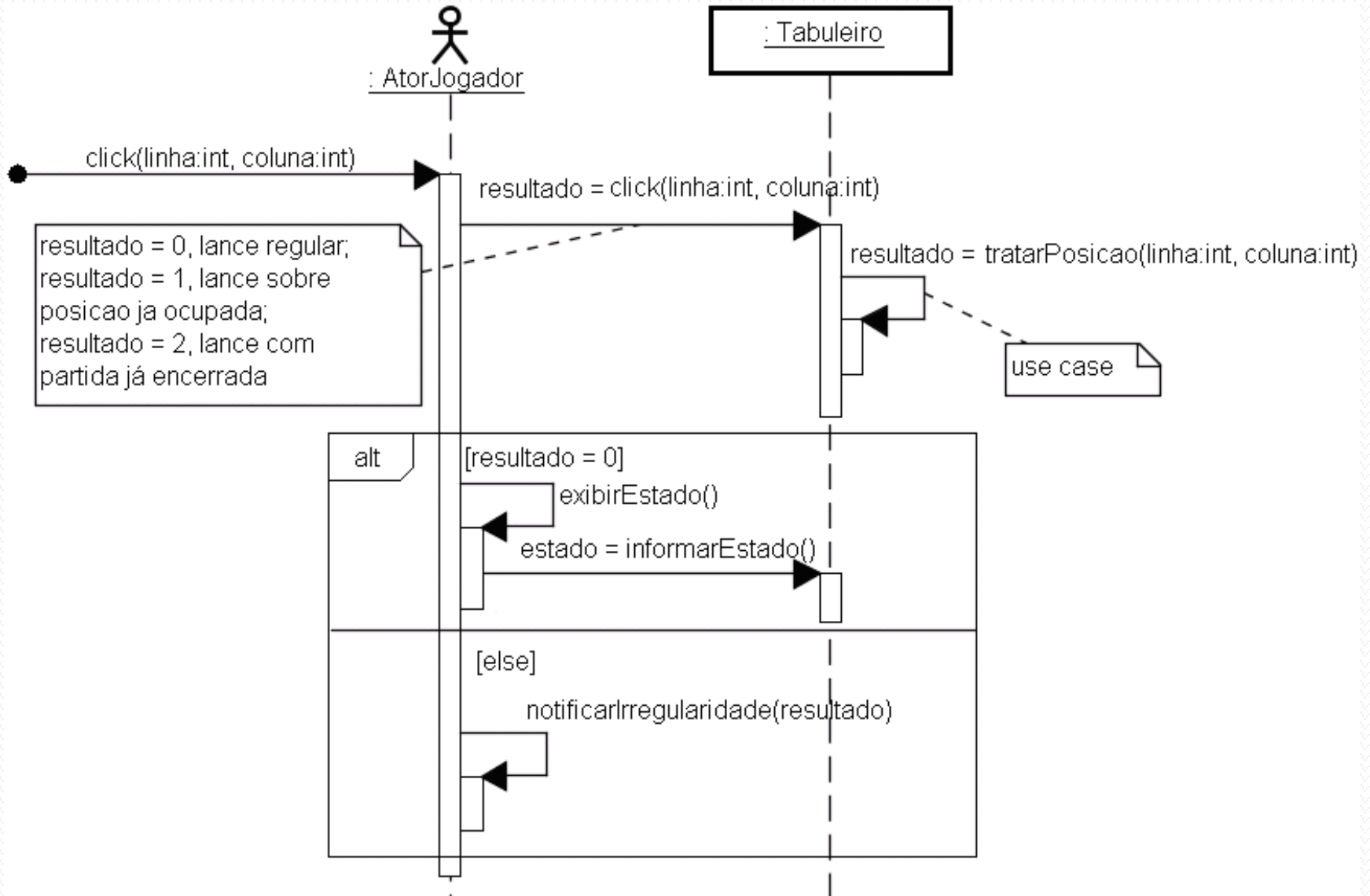


***Atribuição da responsabilidade***



***Identificação de método(s)***

# Exemplo: “procedimento de lance” (JV)



# Exemplo: “procedimento de lance” (JV)

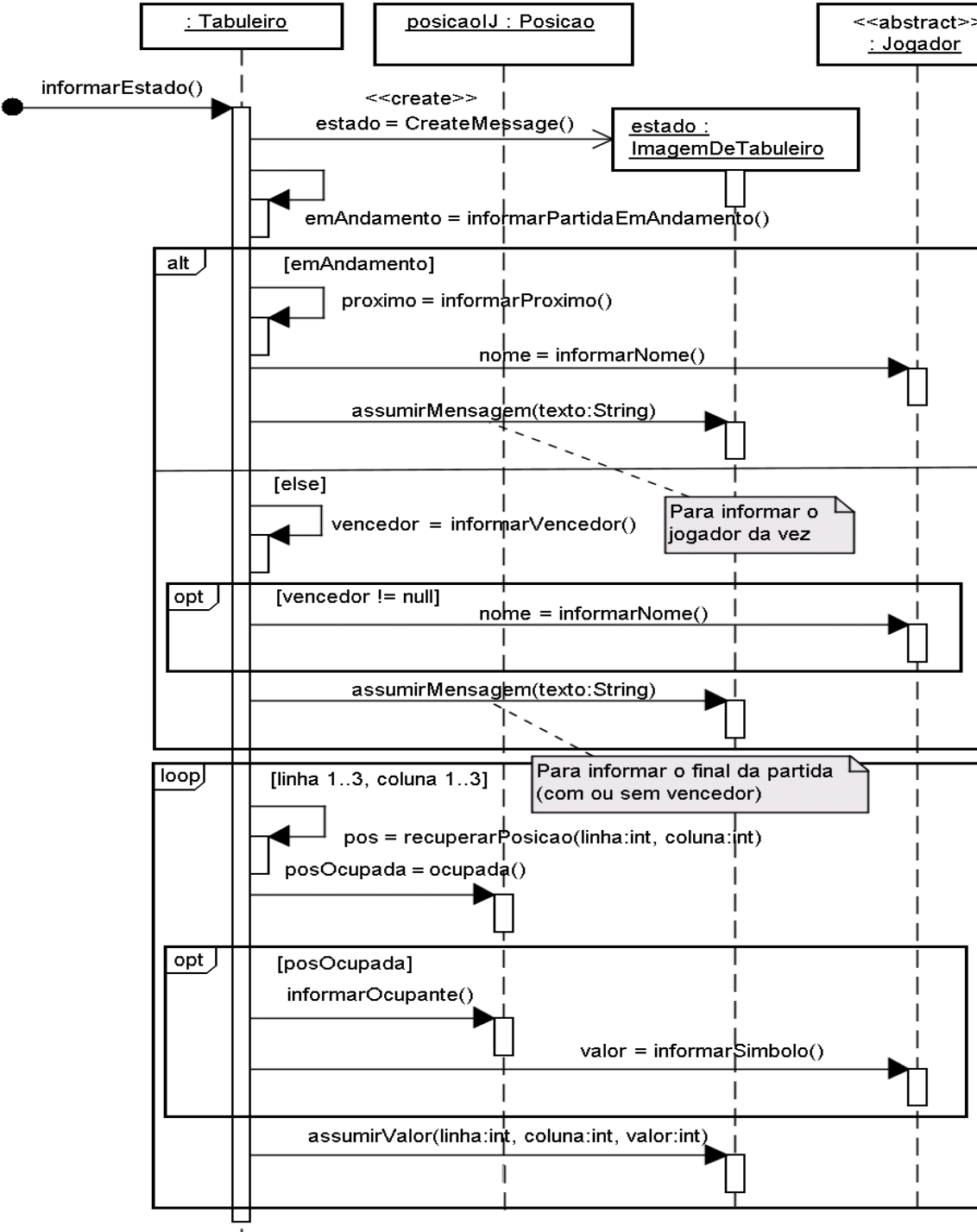
- Em relação à modelagem com diagrama de comunicação
  - Considera todas as possibilidades previstas no diagrama de atividades
    - Posição regular escolhida (cenário principal)
    - Posição irregular escolhida
  - Mais detalhes identificados
    - Retornos de métodos identificados e tratamento dos retornos
    - Novos métodos identificados

## Exemplo: “procedimento de lance” (JV)

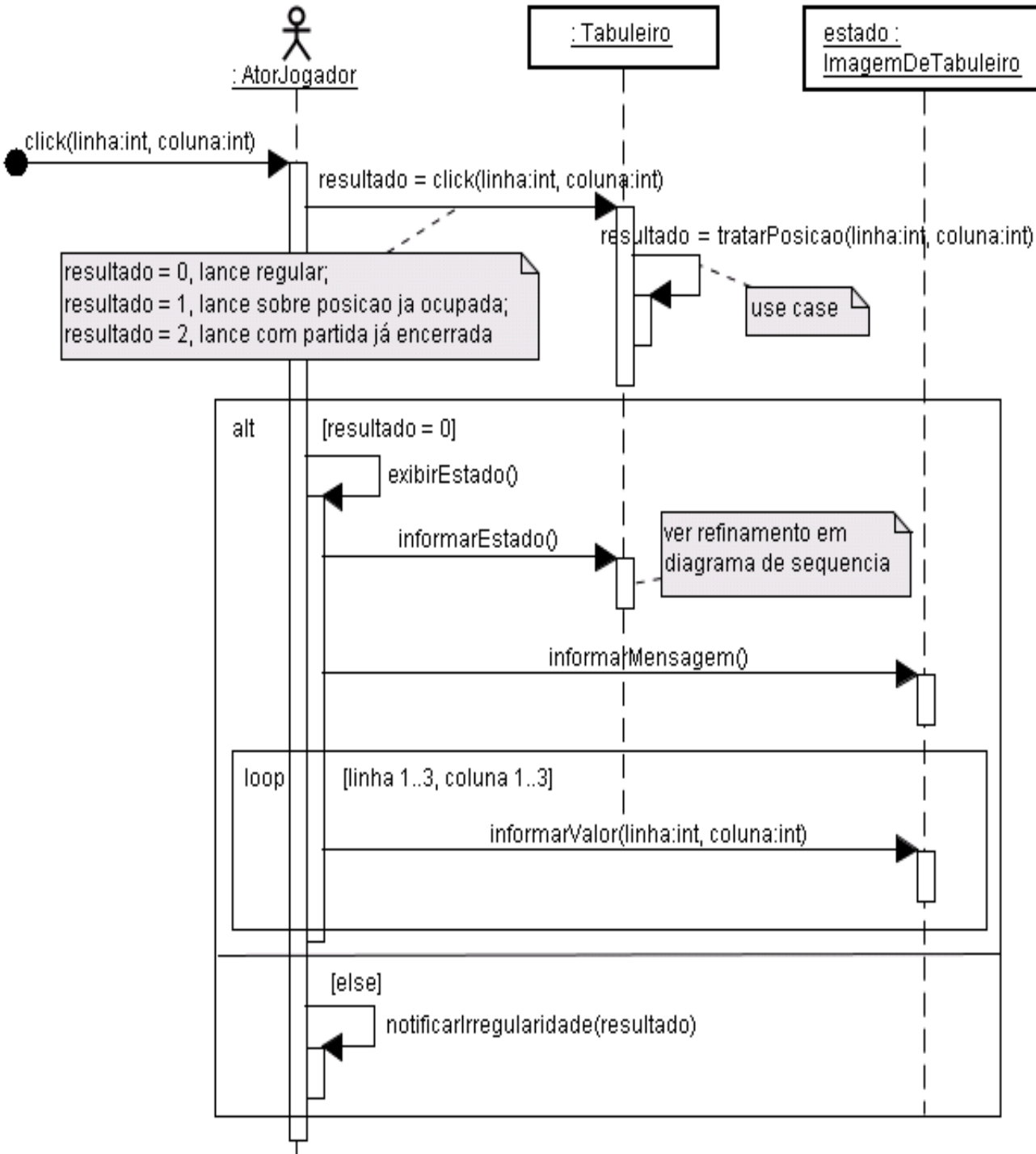
- Detalhando mais a modelagem → o método `informarEstado` (responsabilidade atribuída a tabuleiro) não é elementar
  - Possível detalhá-lo no próprio diagrama
  - Possível produzir um novo diagrama para tratá-lo

# Exemplo – método informarEstado

- Qual o tipo de retorno (o estado do jogo)?
  - Decisão de projeto → instância de uma nova classe, ImagemDeTabuleiro contendo a mensagem a ser apresentada na interface e uma matriz de inteiros, representando as posições
  - Decisão de projeto → Tabuleiro responsável por instanciar ImagemDeTabuleiro e compor a instância (valores dos atributos)

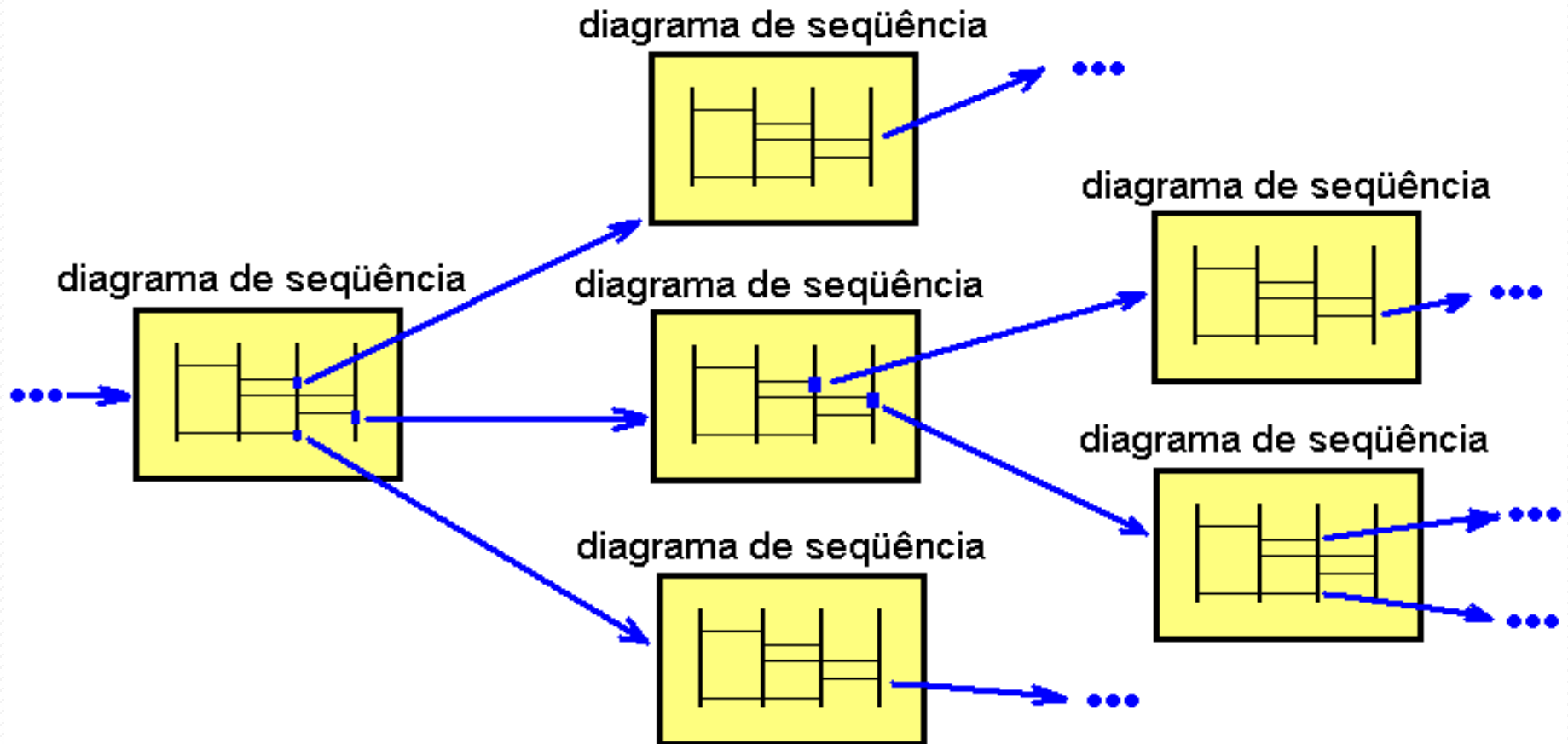


# Exemplo – método informarEstado



Exemplo:  
“procedimento  
de lance” (JV)

# Possibilidade de detalhar diagrama de sequência em outro diagrama de sequência





# Critérios para definir o encerramento da modelagem com diagrama de sequência

- Métodos que apresentem algoritmos curtos e óbvios não demandam refinamento em diagrama de sequência

# Critérios para definir o encerramento da modelagem com diagrama de sequência

- Métodos que apresentem algoritmos não triviais e que suas execuções envolvam quantidades significativas de interações demandam refinamento em diagrama de sequência

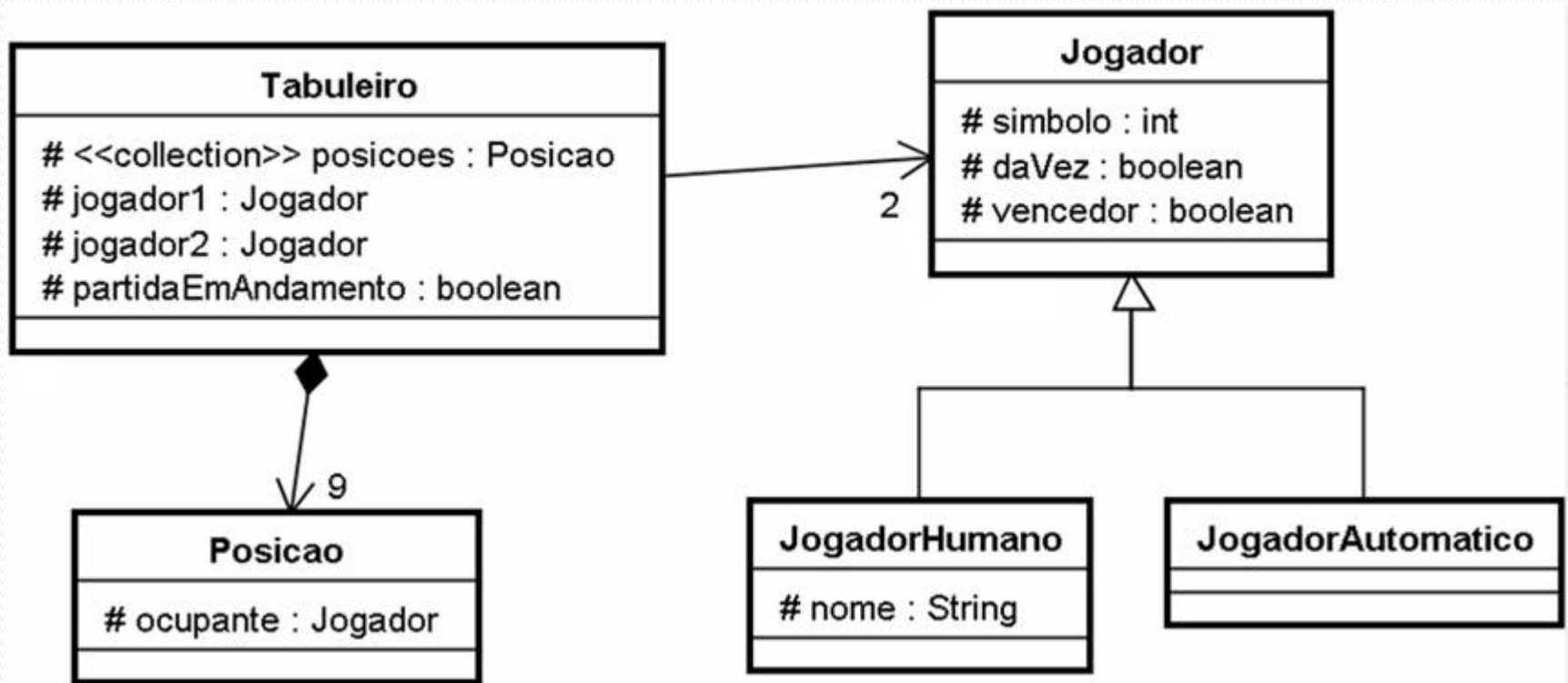
# Critérios para definir o encerramento da modelagem com diagrama de sequência

- Métodos que apresentem algoritmos não triviais, mas que suas execuções não envolvam quantidades significativas de interações, não demandam refinamento em diagrama de sequência
  - Modelagem com diagrama de sequência não é voltada especificamente para modelar algoritmo
  - Tais métodos podem ter seus algoritmos detalhados em diagramas de atividades

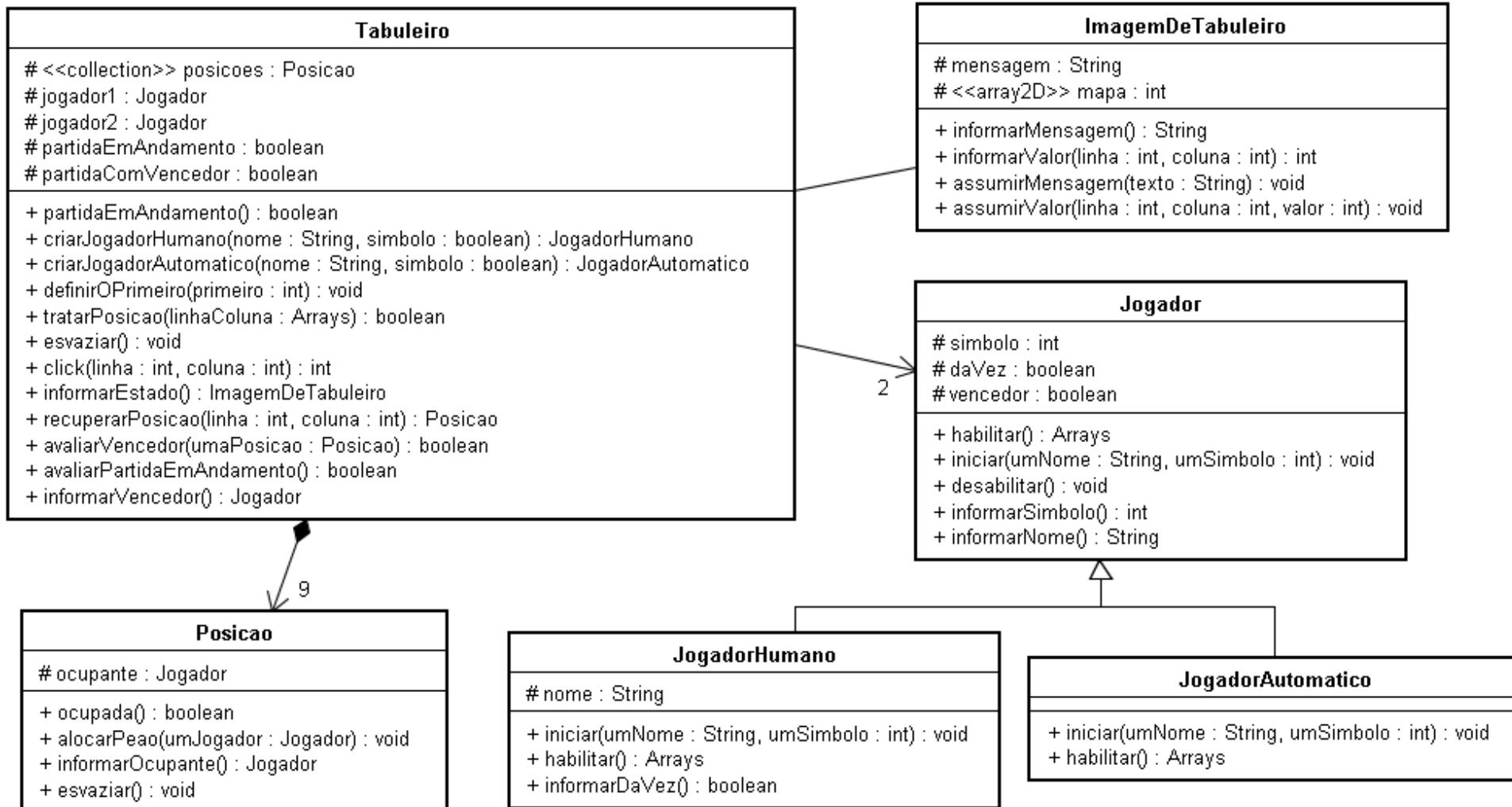
# Refinamento de casos de uso simples

- Casos de uso que consistem em simples operações de leitura ou escrita em base de dados
  - CRUD (Criar, Recuperar, Atualizar, Excluir)
- Possibilidade do refinamento consistir em
  - Apenas um diagrama de atividades
  - Apenas um diagrama de sequência
  - Uma simples nota associada ao caso de uso, no diagrama de casos de uso

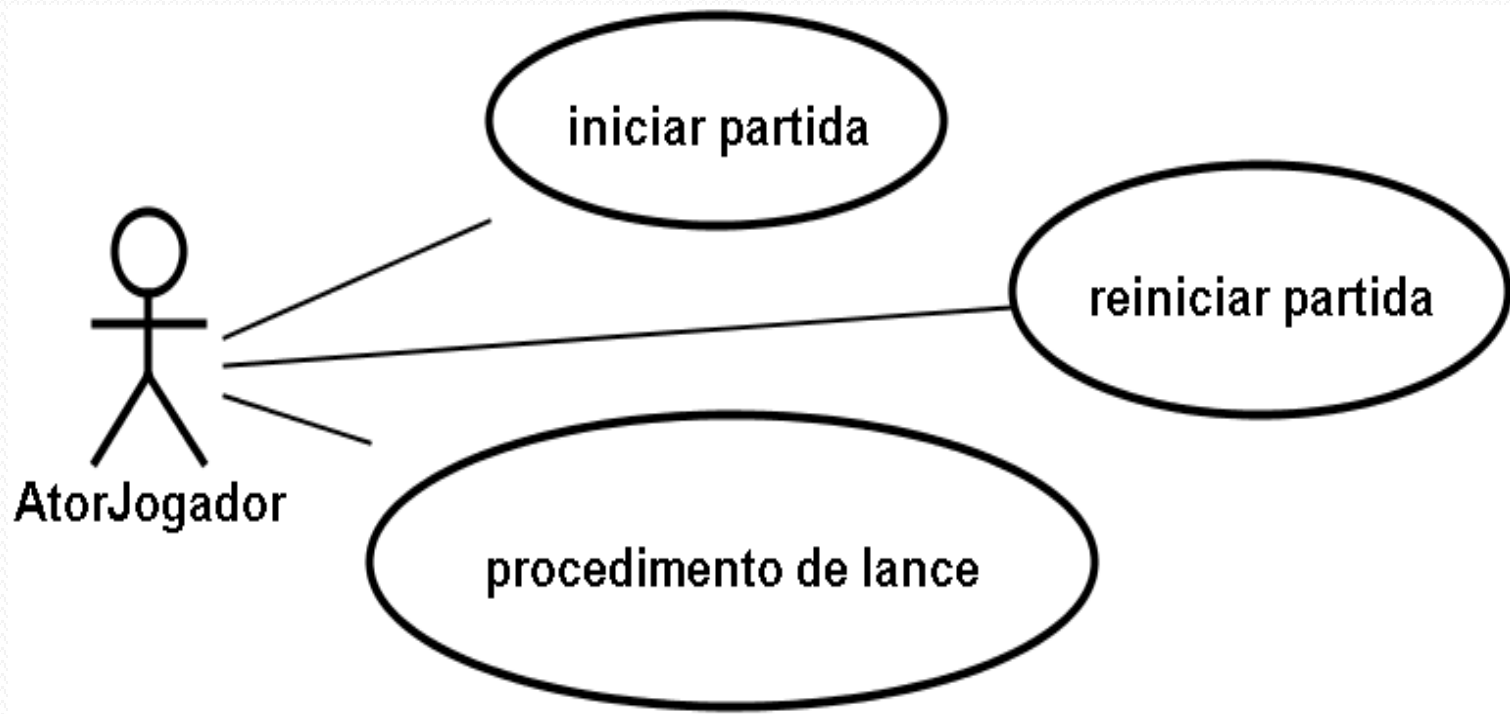
# Efeitos colaterais do refinamento de casos de uso – diagrama de classes “de”



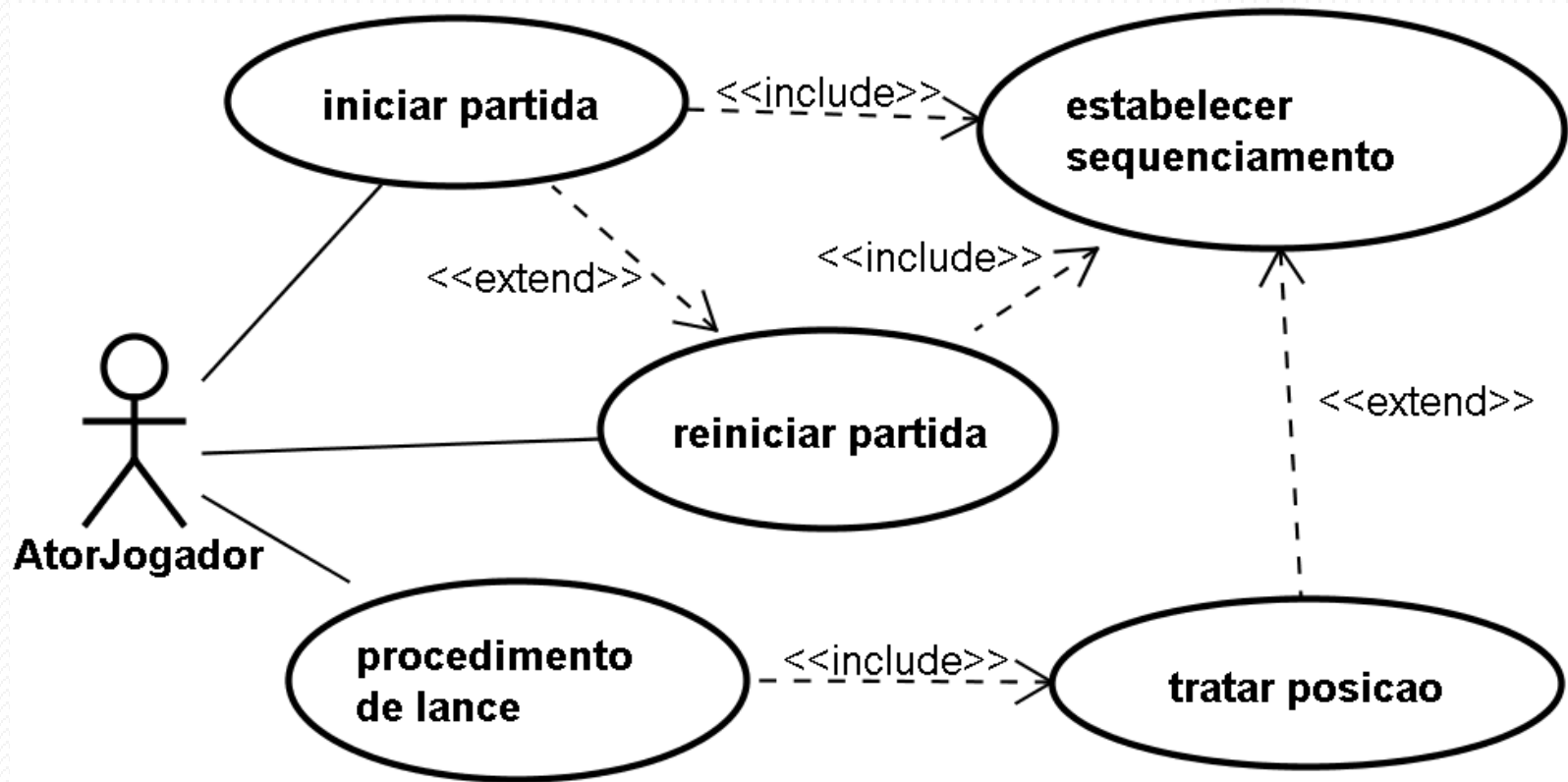
# Efeitos colaterais do refinamento de casos de uso – diagrama de classes “para”



# Efeitos colaterais do refinamento de casos de uso – diagrama de casos de uso “de”

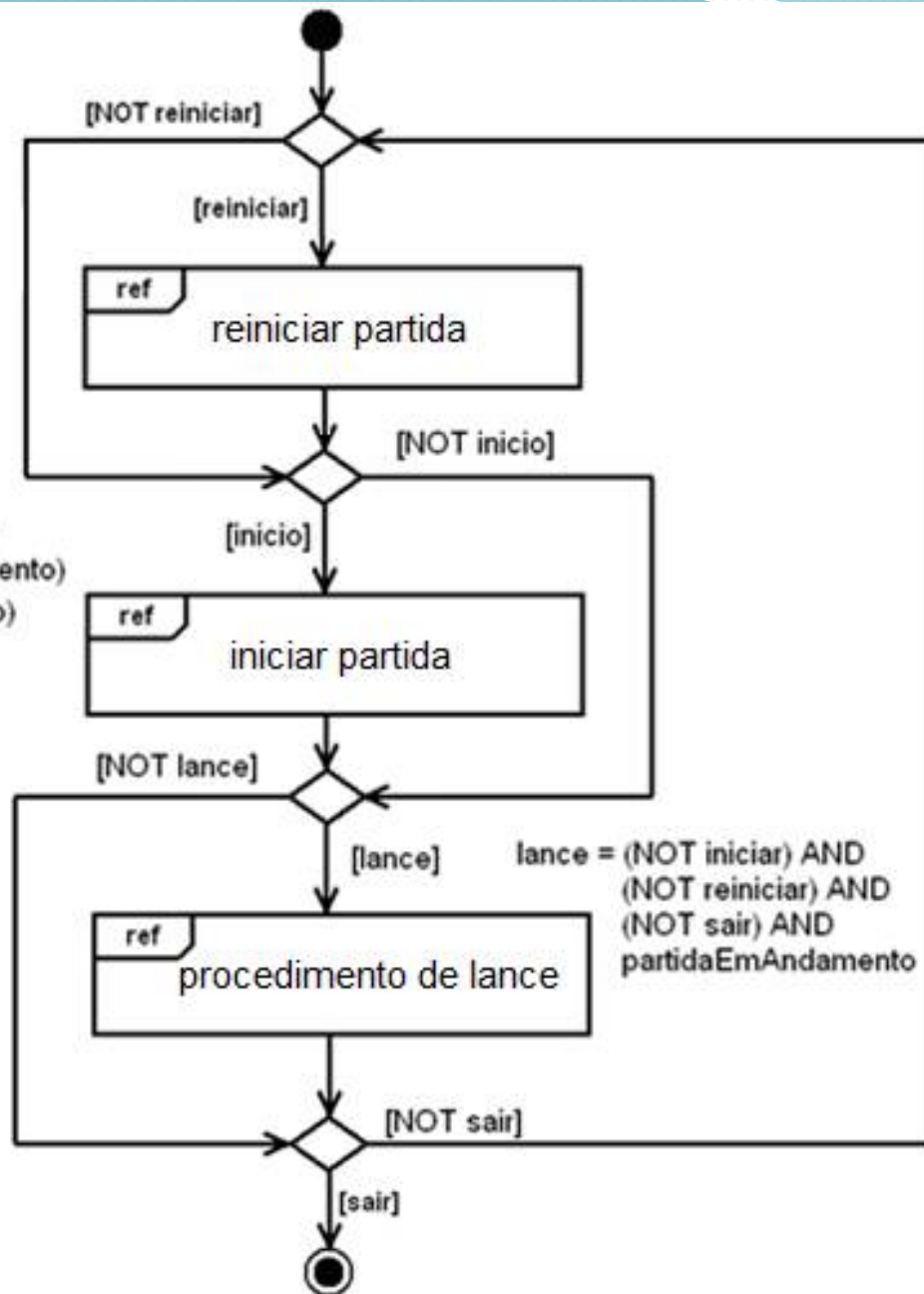


# Efeitos colaterais do refinamento de casos de uso – diagrama de casos de uso “*para*”





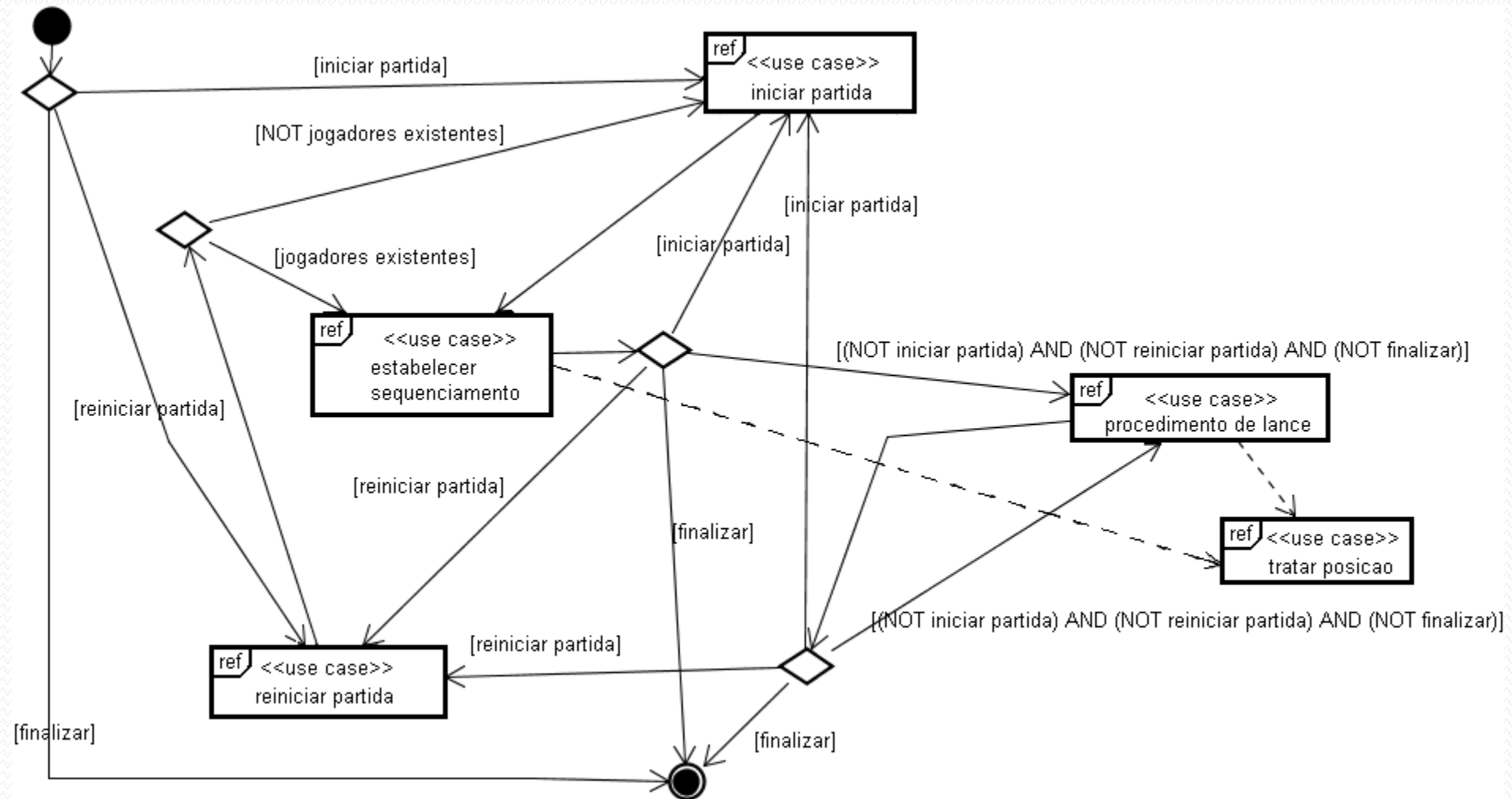
início = (NOT sair) AND  
((NOT partidaEmAndamento)  
OR confirmarInterrupcao)



Efeitos  
colaterais do  
refinamento de  
casos de uso –  
diagrama de  
visão geral de  
interação “de”

# Efeitos colaterais do refinamento de casos de uso

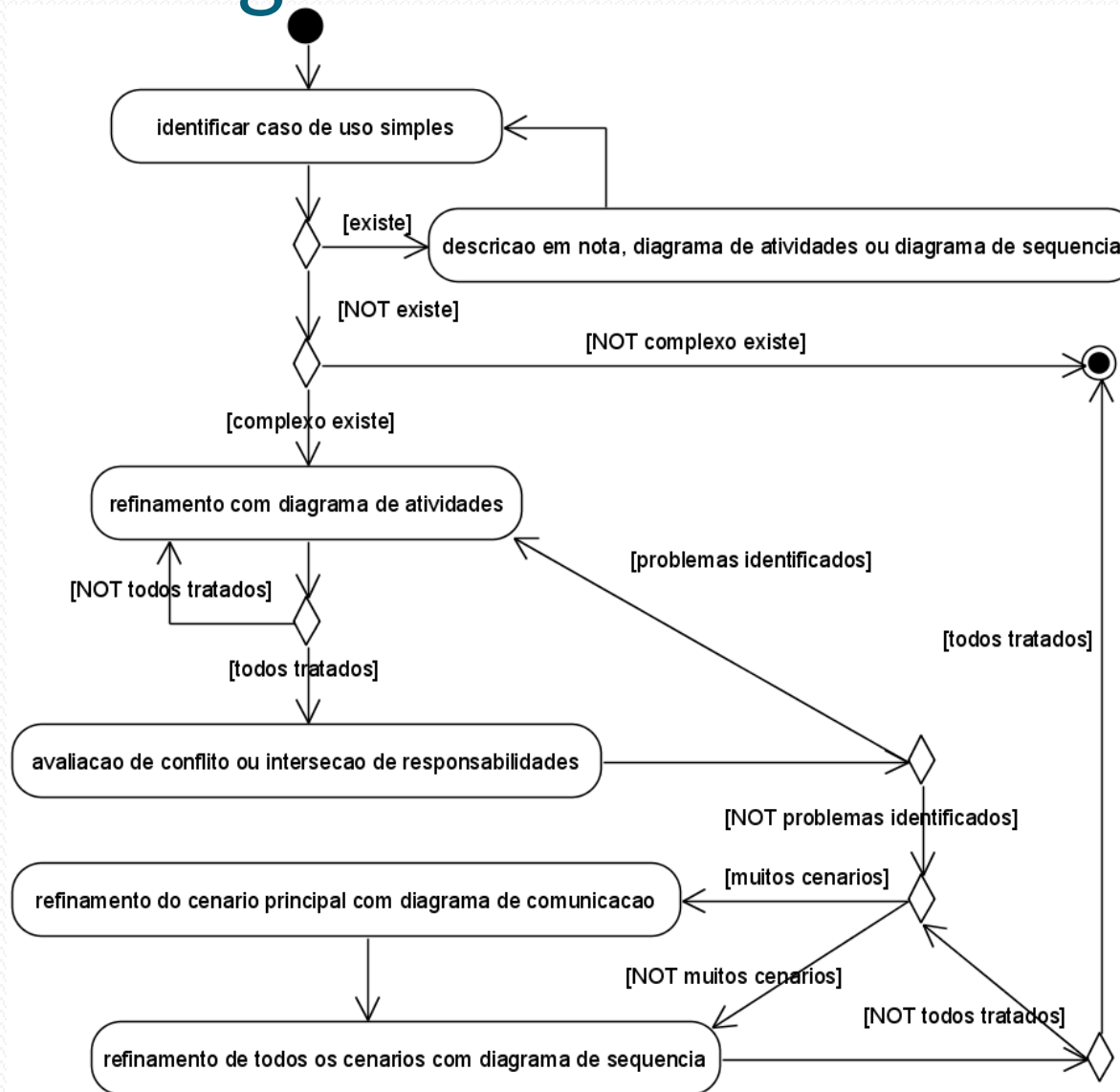
## - diagrama de visão geral de interação “para”



# Resultados parciais após a terceira etapa do processo de modelagem

- Etapa 1
  - Modelagem estrutural de sistema
  - Modelagem dinâmica de sistema
- Etapa 2
  - Identificação de atributos
- Etapa 3
  - Refinamento dos casos de uso
    - Diagramas modelando os casos de uso
    - Identificação de métodos
    - Evolução das modelagens estrutural e dinâmica

# Sumário da terceira etapa do processo de modelagem



# Considerações sobre esta aula

- Etapa 3 do processo de modelagem → Refinamento de casos de uso
  - ***responsabilidade → atribuição da responsabilidade → identificação de método(s)***
  - Evolução da modelagem dinâmica produzida na etapa 1
  - Refinamento da modelagem estrutural produzida nas etapas 1 e 2
- Passos da etapa ilustrados com a modelagem do programa Jogo-da-velha

# Referências

Booch, G.; Jacobson, I. e Rumbauch, J. **UML: Guia do Usuário**. Campus, 2006.

Silva, R. P. **UML 2 em modelagem orientada a objetos**. Visual Books, 2007.