

# Metodologia de modelagem

## Etapa 1

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

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

# Objetivo

- Apresentar a primeira etapa da metodologia de modelagem
  - Etapa 1 – Modelagem estrutural e dinâmica em alto nível de abstração

# Diagramas UML usados

- Diagrama de classes
- Diagrama de casos de uso
- Diagrama de visão geral de interação

# Etapa 1 – Modelagem estrutural e dinâmica em alto nível de abstração

- **Modelagem estrutural em alto nível de abstração** → identificar os elementos do domínio do problema e, a partir deles, compor diagrama(s) de classes

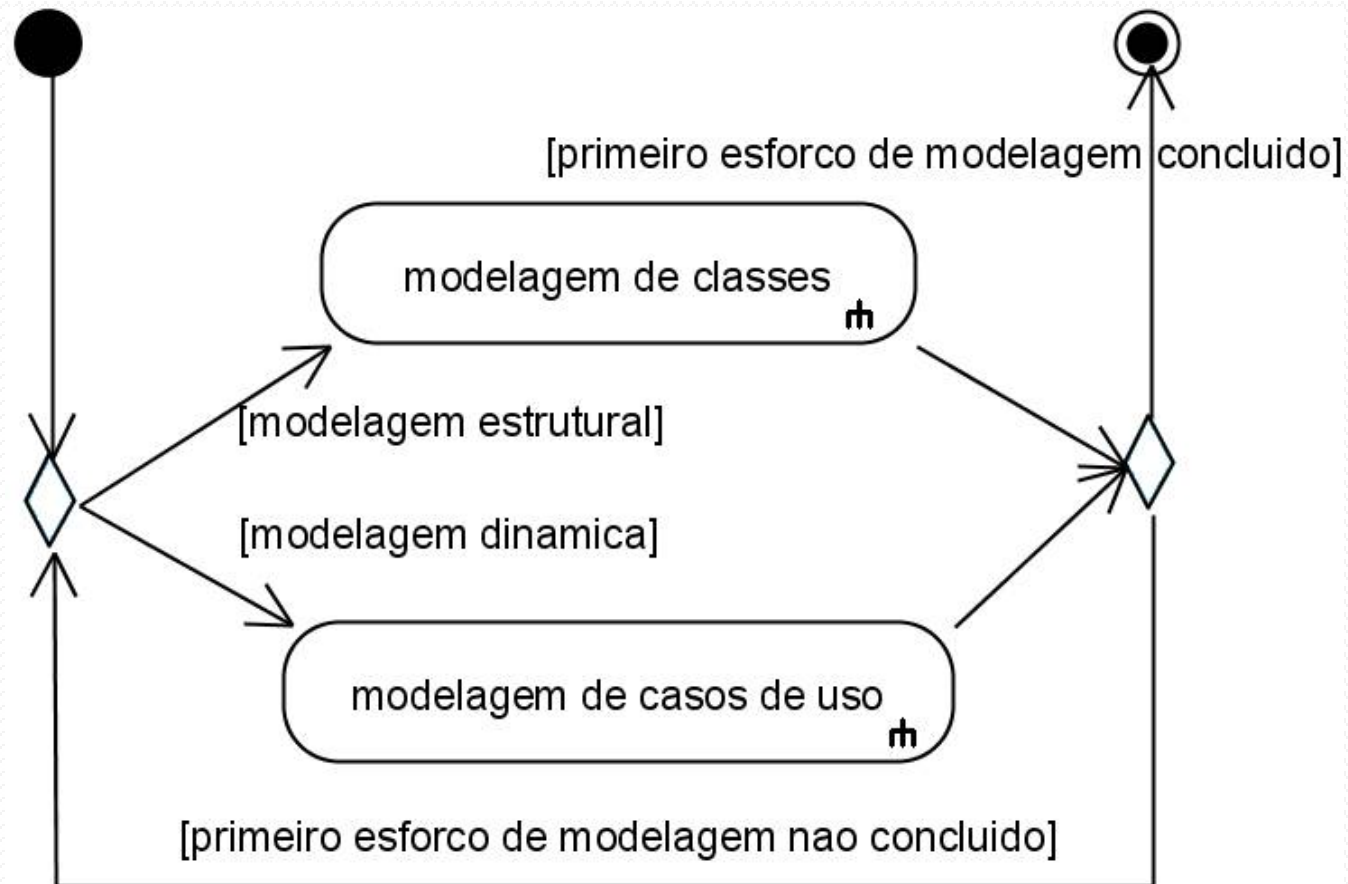
# Etapa 1 – Modelagem estrutural e dinâmica em alto nível de abstração

- **Modelagem dinâmica em alto nível de abstração** → identificar as funcionalidades do software a desenvolver e, a partir delas, construir diagrama(s) de casos de uso

# Etapa 1 – Modelagem estrutural e dinâmica em alto nível de abstração

- No início do desenvolvimento, não há relação causal entre modelagem estrutural e modelagem dinâmica
  - Pode-se chegar ao mesmo resultado começando por uma ou por outra

# Etapa 1 – Modelagem estrutural e dinâmica em alto nível de abstração





# Primeiro esforço de modelagem estrutural

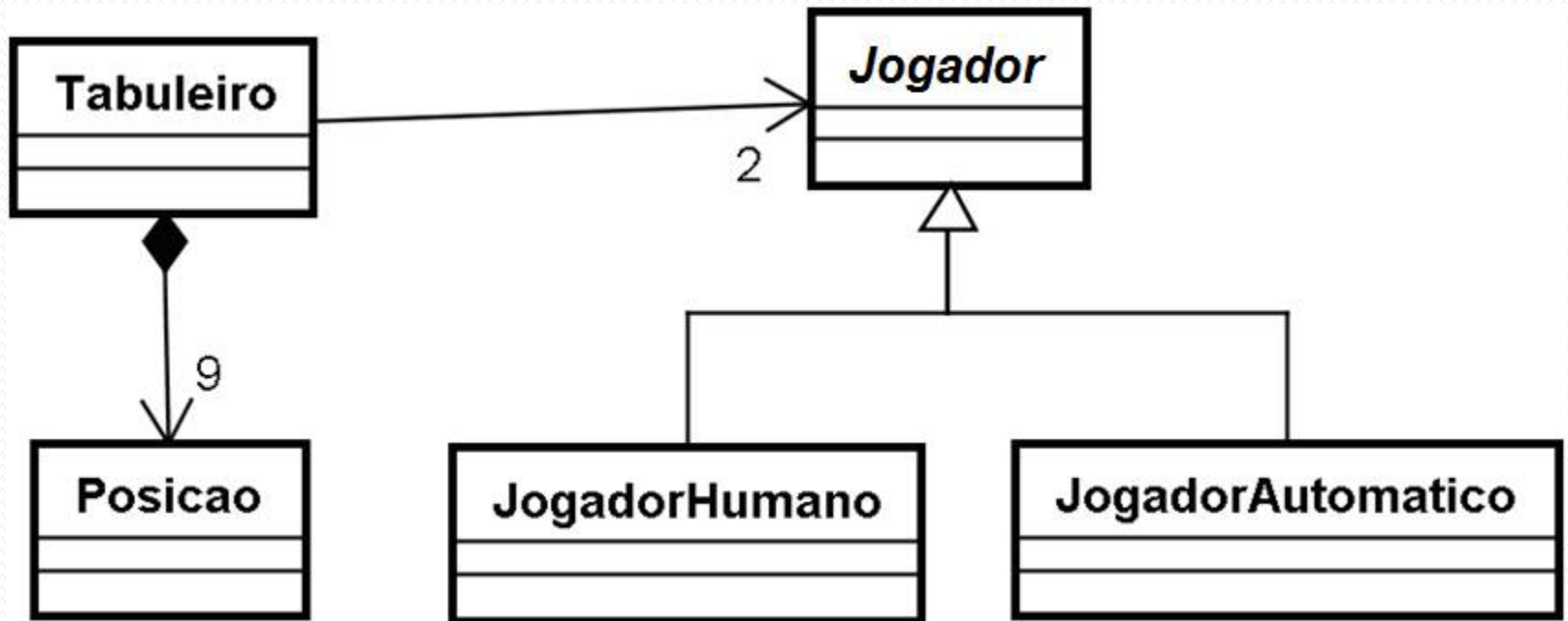
- A pergunta clássica é: *quais os elementos do domínio do problema?*
- A resposta contém tanto elementos físicos, quanto conceituais
- Esses elementos correspondem à primeira aproximação do conjunto de classes do sistema



# Exemplo – modelagem estrutural do Jogo-da-velha

- *Quais os elementos do domínio do problema?*
  - Tabuleiro é um elemento do domínio do problema
  - O tabuleiro agrupa 9 posições
  - 2 jogadores estão envolvidos
  - Pode-se pensar em 2 tipos de jogador
    - Correspondente ao usuário
    - Correspondente ao elemento do programa que disputa partidas contra usuário

# Exemplo – modelagem estrutural do Jogo-da-velha



# Comentários sobre o exemplo

- Identificação de classes e relacionamentos
- Algumas decisões de projeto tomadas
  - As regras do jogo poderiam originar uma classe, mas decidiu-se embuti-las em Tabuleiro
  - Tabuleiro é quem aponta os demais elementos
  - Decidiu-se agrupar os dois tipos de jogador em uma hierarquia de herança

# Comentários sobre o exemplo

- Identificadas classes sem atributos ou métodos e alguns relacionamentos
  - Sem a pretensão de identificar todas as classes ou todos os relacionamentos

# Comentários sobre o exemplo

- É possível reconhecer a existência de alguma ligação entre classes, mas não o tipo específico de relacionamento
  - A inserção de parte dos relacionamentos pode ser adiada para estágios posteriores do processo de modelagem
  - Pode-se alterar o tipo de relacionamento entre duas classes
    - Como uma composição ser trocada por uma agregação, por exemplo

# Comentários sobre o exemplo

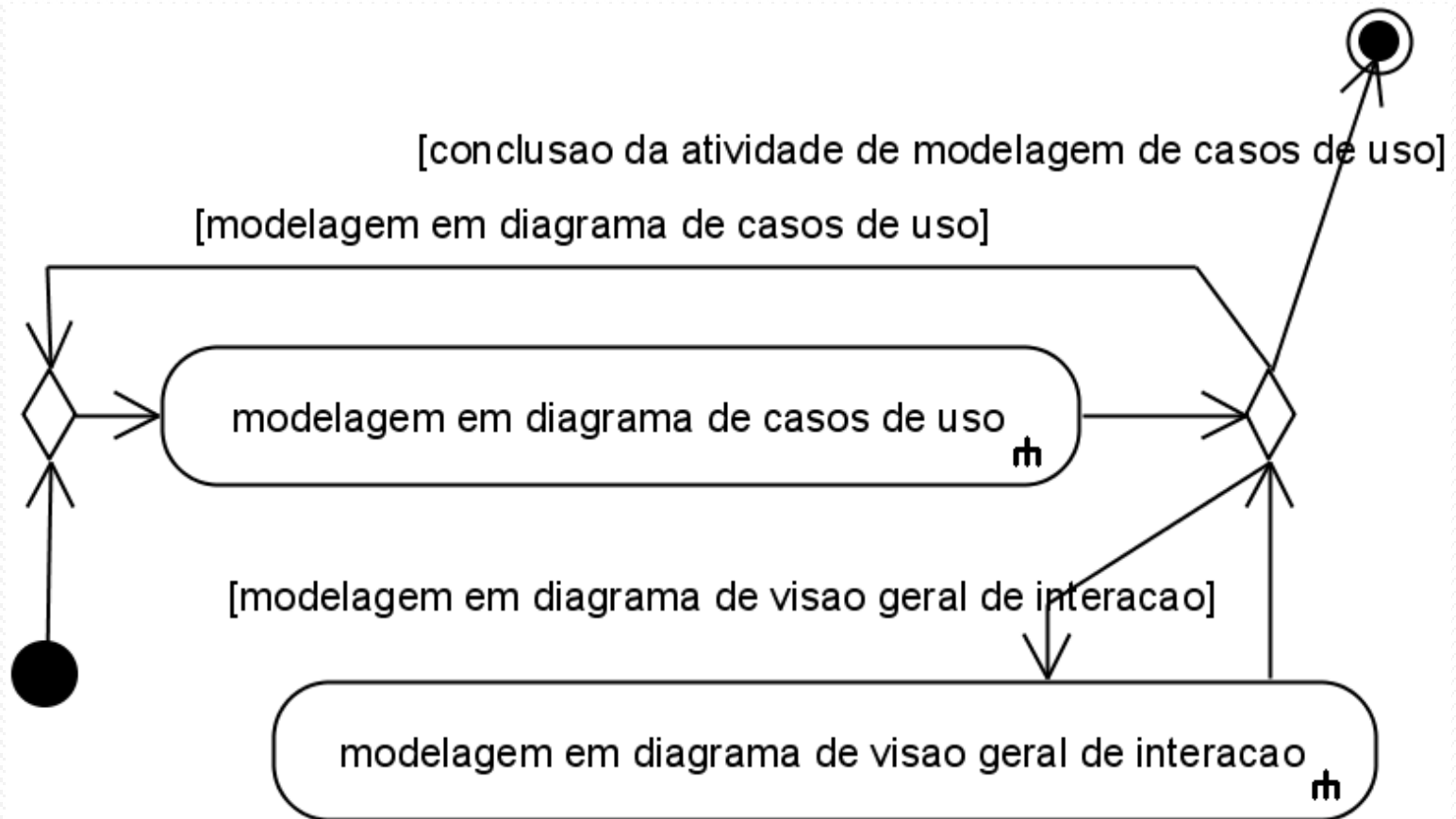
- Não há o compromisso nessa primeira etapa de produzir uma modelagem precisa e completa
  - Registra-se o que é possível reconhecer com as informações disponíveis

# Primeiro esforço de modelagem dinâmica

- Um ou mais diagramas de casos de uso
  - Funcionalidades do sistema
  - Elementos externos com que o sistema interage
- Diagrama de visão geral de interação (um ou mais)
  - Nós terão correspondência com os casos de uso
  - Definição da ordem em que os casos de uso podem ocorrer
- Inicia com a modelagem em diagrama de casos de uso



# Primeiro esforço de modelagem dinâmica



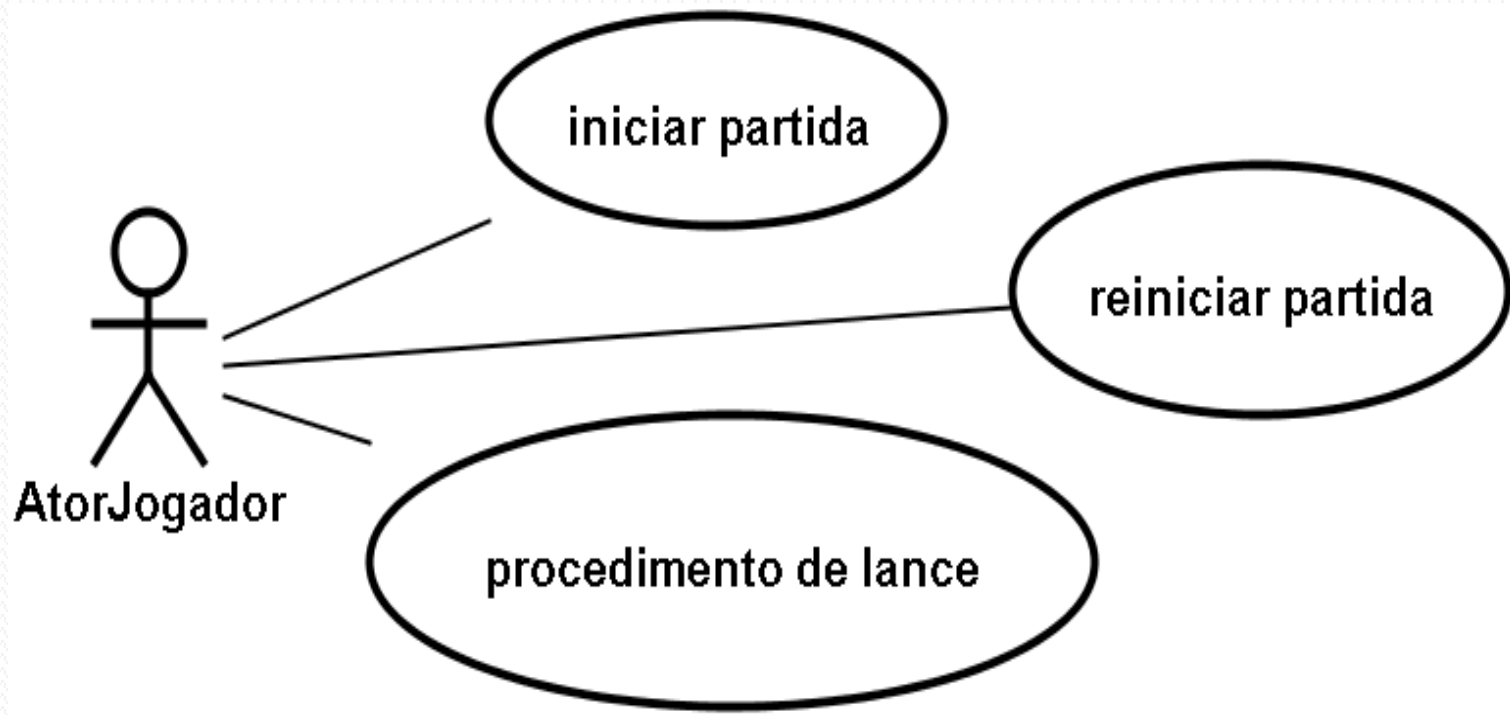
# Modelagem com diagrama de casos de uso

- A pergunta: *quais as funcionalidades do software?*
  - A meta da pergunta é identificar os casos de uso
  - Funcionalidades completas, externamente perceptíveis
- Próxima pergunta: *que elementos externos interagirão com o software em desenvolvimento?*
  - Identificação de atores

# Exemplo – modelagem dinâmica do Jogo-da-velha

- *Quais as funcionalidades do software?*
  - Iniciar partida
  - Reiniciar partida
  - Procedimento de lance
- *Que elementos externos interagirão com o software em desenvolvimento?*
  - Apenas um tipo de usuário (jogador)

# Exemplo – modelagem dinâmica do Jogo-da-velha



# Comentários sobre o exemplo

- Conjunto de funcionalidades → decisão de projeto
- Identificação de casos de uso e atores
  - Além deles, é preciso definir que ator(es) participa(m) de cada caso de uso
  - Apenas associações entre ator e caso de uso são identificadas no primeiro esforço de modelagem

# Comentários sobre o exemplo

- Ator que modela a interação com o jogador chamado de AtorJogador e não Jogador, que é o identificador de um elemento do domínio do problema
  - Outra sugestão: InterfaceJogador
- Ator ou caso de uso isolado → inconsistência
- Não há o compromisso de identificar todos os casos de uso ou atores nesse primeiro esforço

# Modelagem com diagrama de visão geral de interação

- Pergunta: *em que ordem os casos de uso podem ocorrer em uma execução do programa?*
  - Pode não haver restrição de ordem → qualquer um a qualquer momento
  - Casos de uso que devem ocorrer em sequência
  - Caso de uso que não possa ocorrer antes da ocorrência de outro **etc.**

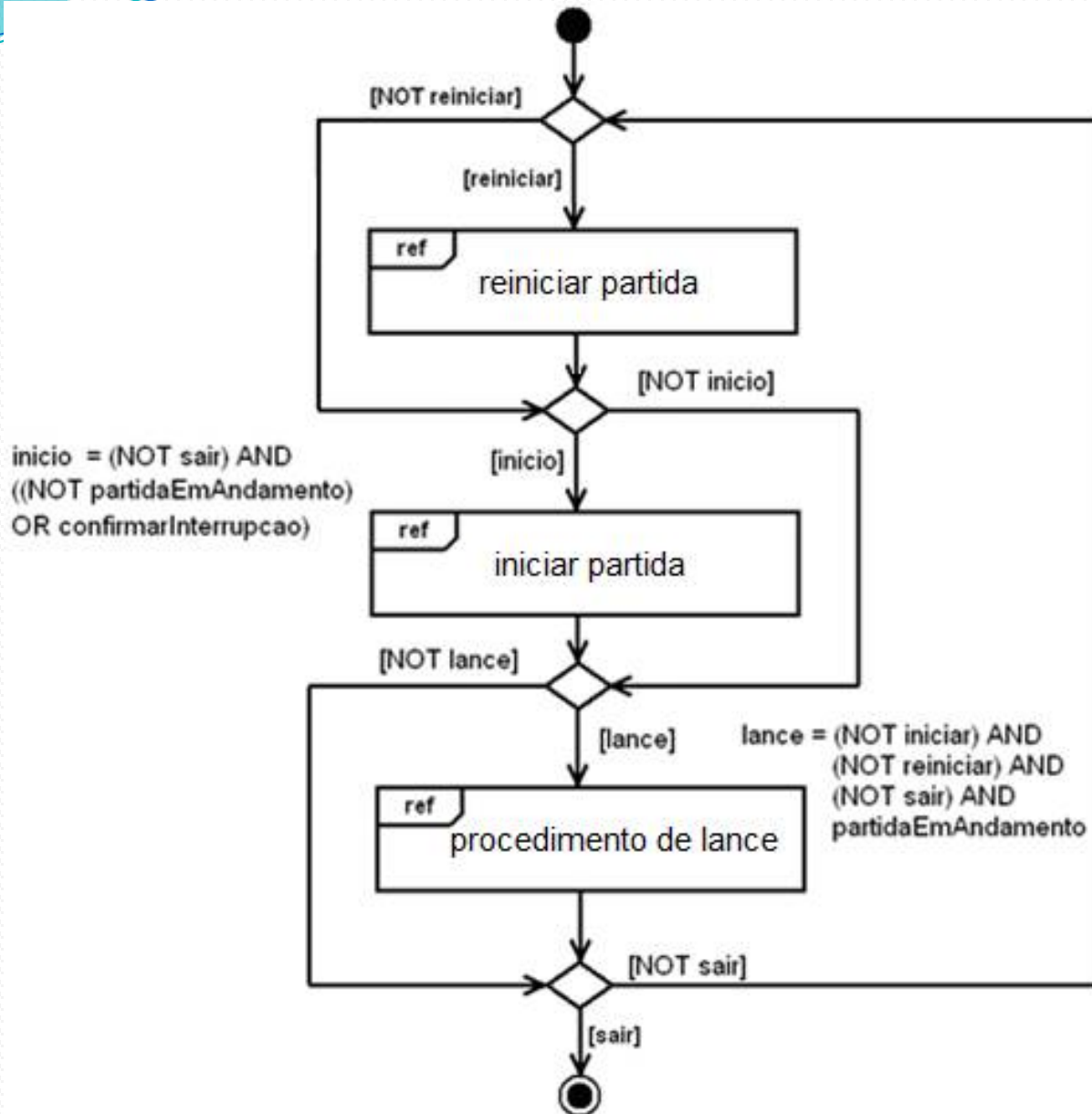


# Modelagem com diagrama de visão geral de interação

- Na construção do diagrama → nodos correspondem a casos de uso, observando
  - Nodos referenciam futuros diagramas de sequência
  - Diagrama com mesmo nome do caso de uso a que corresponde
  - Correspondência um para um entre nodos e casos de uso
  - Nodo → *uso de interação*

# Exemplo – modelagem dinâmica do Jogo-da-velha

- *Em que ordem os casos de uso podem ocorrer em uma execução do programa?*
  - Iniciar partida e reiniciar partida → a qualquer instante
  - Procedimento de lance → apenas após iniciar partida ou reiniciar partida



Exemplo –  
modelagem  
dinâmica do  
Jogo-da-velha

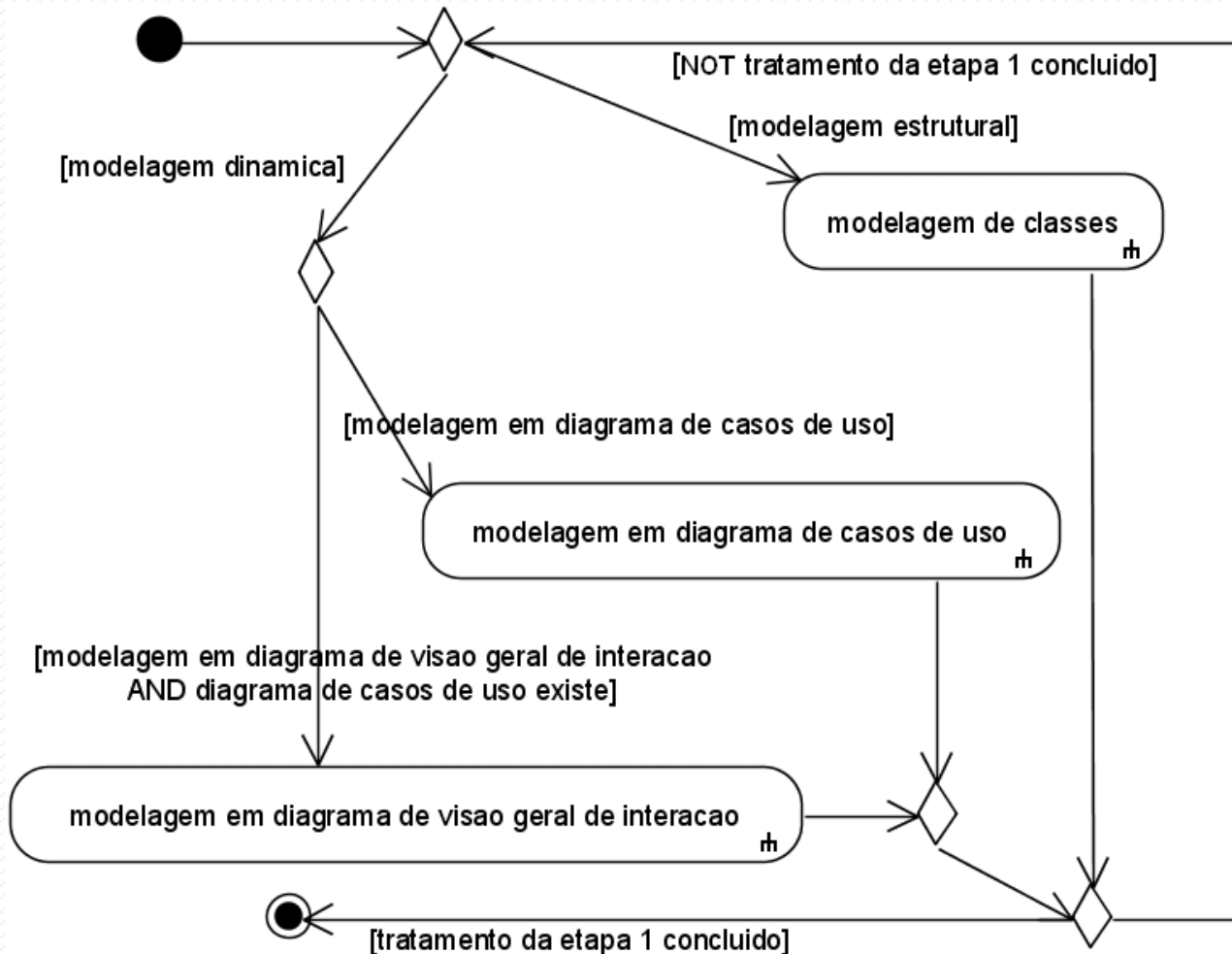
# Comentários sobre o exemplo

- Não se espera que o diagrama esteja completo e acabado neste primeiro esforço
  - Podem faltar informações para definir os percursos
  - Podem faltar informações para definir as guardas

# Resultados parciais após a primeira etapa do processo de modelagem

- Modelagem estrutural (de sistema) → diagrama de classes
- Modelagem dinâmica (de sistema)
  - Diagrama de casos de uso
  - Diagrama de visão geral de interação

# Sumário da primeira etapa do processo de modelagem



# Considerações sobre esta aula

- Etapa 1 do processo de modelagem → Modelagem estrutural e dinâmica em alto nível de abstração
- Exercitar modelagem → considerar o sistema a seguir e aplicar os procedimentos apresentados



# Exercício

## Sistema de Gestão de Escola de Línguas

- O sistema deve gerir todos os tipos de usuários tais como professores, alunos e funcionários;
- Permitir cadastro de todas as turmas juntamente com seus horários de aula e professor responsável;
- Fazer matrícula de um aluno em uma turma específica;
- Cadastrar todos os materiais que são utilizados em um determinado curso;
- Registrar a entrega de material do curso para um determinado aluno;
- Registrar presença do aluno nas aulas da turma;
- Registrar avaliações previstas para cada turma;
- Registrar notas de avaliações de cada aluno para cada turma;
- Registrar pagamento de mensalidade de aluno de um curso;
- Gerar boletim de notas de aluno por curso;
- Gerar certificado de conclusão de curso para aluno;
- Gerar relatório financeiro da escola.

# 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.