

Metodologia de modelagem

Etapa 5

Prof. Murillo G. Carneiro
FACOM/UFU

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

Objetivo

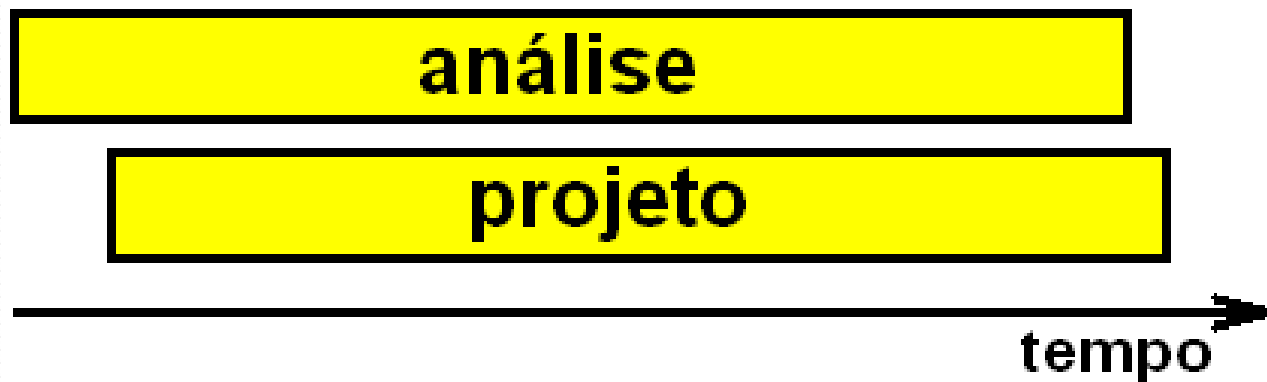
- Apresentar a quinta etapa da metodologia de modelagem
 - Etapa 5 – Introdução de elementos do domínio da solução computacional

Diagramas UML usados

- Diagrama de classes
- Diagrama de casos de uso
- Diagrama de visão geral de interação
- Diagrama de atividades
- Diagrama de comunicação
- Diagrama de sequência
- Diagrama de máquina de estados

Etapa 5 – Introdução de elementos do domínio da solução computacional

- Etapa de projeto → foco nos elementos do domínio da solução computacional
- Etapa de análise (tratada nas quatro etapas anteriores) → foco nos elementos do domínio do problema
- Esforços intercalados no tempo:



Etapa 5 – Introdução de elementos do domínio da solução computacional

- Etapa de análise abstrai especificidades computacionais
 - Ex.: Interfaces tratadas como atores, abstraindo tecnologias específicas
 - Interfaces com usuários, bancos de dados, etc. → sempre atores

Etapa 5 – Introdução de elementos do domínio da solução computacional

- Etapa de projeto → definição das especificidades tecnológicas que comporão o software
 - Ex.: Interface web, stand-alone, com que opções tecnológicas?
 - Elementos das soluções tecnológicas adotadas são incorporados à modelagem

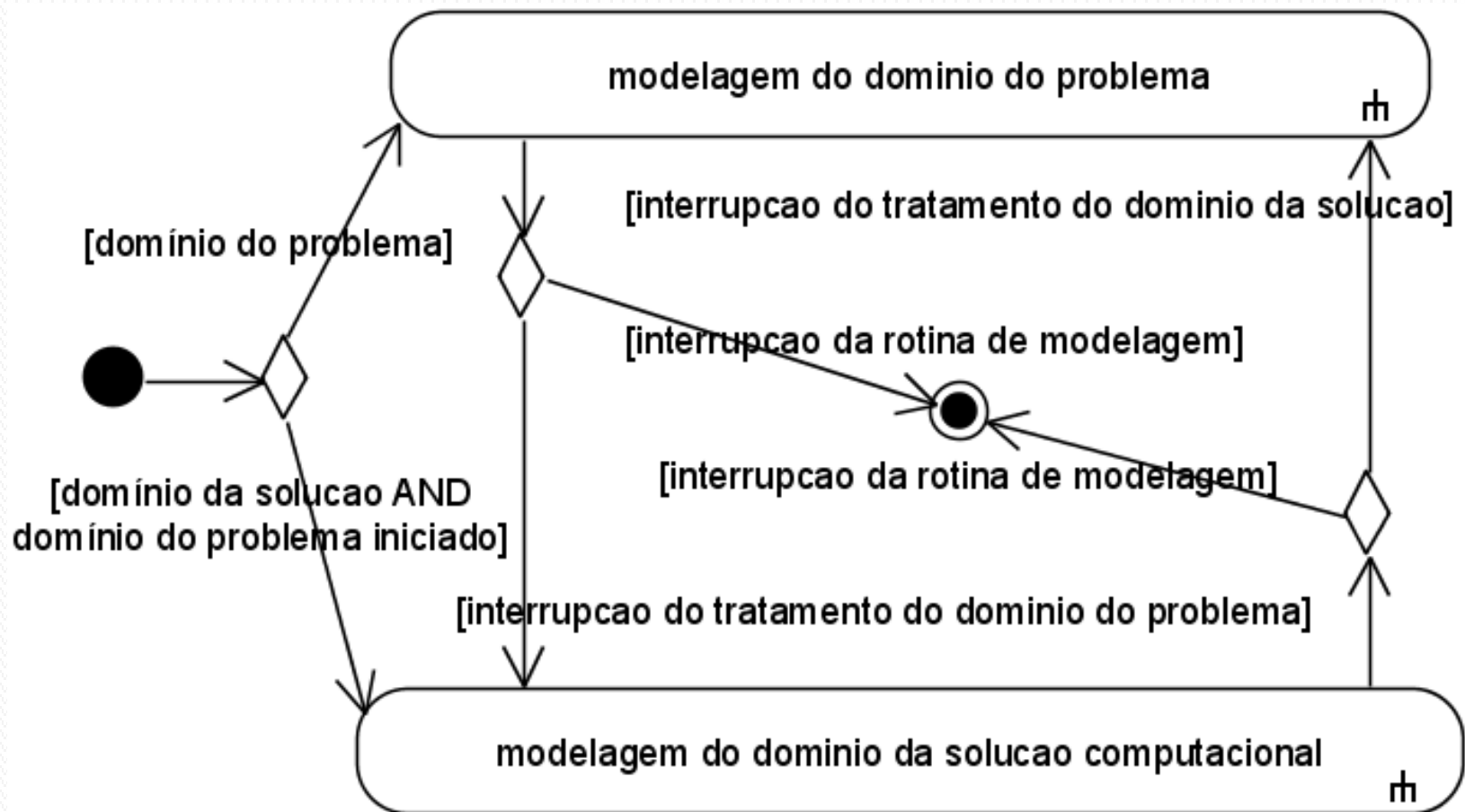
Preocupação em manter baixo acoplamento

- Minimizar referências dos elementos do domínio do problema para os elementos do domínio da solução e vice-versa

Tratamento dos elementos do domínio da solução computacional

- Identificar as especificidades computacionais necessárias
- Modelá-las como é feito com elementos do domínio do problema
 - Percorrer as quatro primeiras etapas
 - Inserir novos elementos ou modificar os existentes

Análise e projeto → iteratividade



Atividades da etapa 1 → foco no domínio da solução

- Identificação de novas classes que modelem especificidades computacionais
 - Interface, persistência etc.
- Identificação de novos casos de uso correspondentes a ações inerentes ao uso do computador
 - Salvar informação, recuperar informação, validação de usuário etc.
- Eventual quebra da modelagem de classes, de casos de uso ou de visão geral de interação em mais diagramas

Atividades da etapa 2 → foco no domínio da solução

- Identificação de novos atributos
 - Nas classes novas
 - Nas previamente identificadas
- Necessidade → atributos que estabeleçam referências entre os elementos dos domínios do problema e da solução

Atividades da etapa 3 → foco no domínio da solução

- Inserção de procedimentos referentes ao tratamento de especificidades computacionais nos casos de uso previamente refinados
- Refinamento dos novos casos de uso identificados
 - Se houver

Atividades da etapa 4 → foco no domínio da solução

- Modelagem de estados das novas classes
 - Se julgado adequado, em função de sua dinâmica

Identificação de necessidades computacionais

- Buscar na especificação de requisitos
- Possibilidades (exemplo: persistência)
 - Não incluída na especificação de requisitos, mas necessidade identificada ao longo da modelagem
 - Faz parte dos requisitos, sem menção a como implementar
 - Definição parcial de como tratar
 - Estabelecer persistência em banco de dados, por exemplo, sem especificar qual
 - Estabelecer uma opção tecnológica específica
 - Um banco de dados específico, por exemplo

Caminho para a inserção do domínio da solução computacional

1. Identificar as demandas tecnológicas do programa
 - Explícitas ou não na especificação de requisitos
2. Definir a opção tecnológica para cada uma
 - Pode envolver maior ou menor grau de liberdade
3. Incluir as opções selecionadas na especificação
 - Acrescentar ou alterar elementos da modelagem

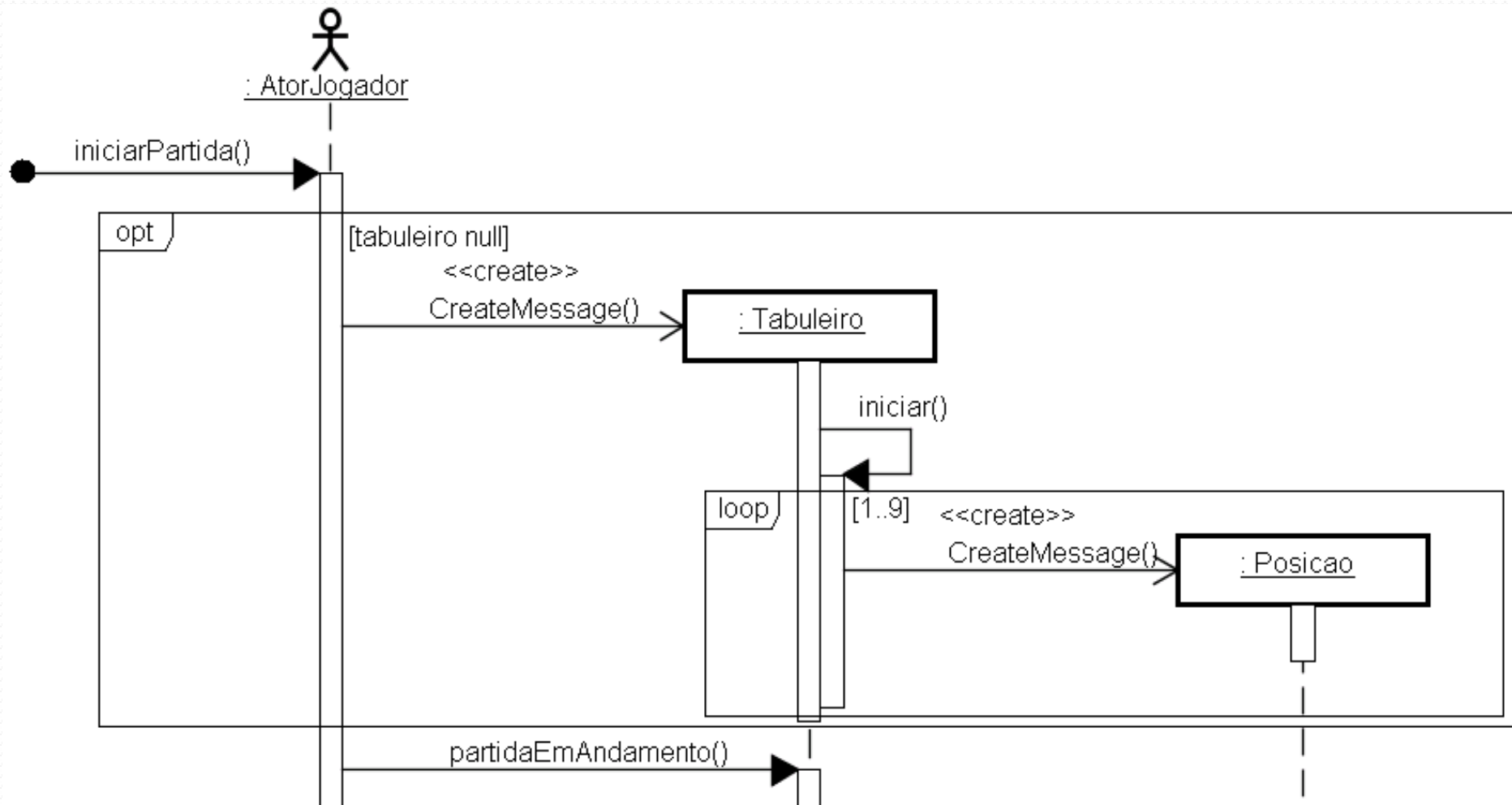
Exemplo → elementos do domínio da solução computacional no Jogo-da-velha

- Necessidades computacionais identificadas
 - Interface gráfica stand-alone, baseada em Java Swing
 - Persistência em arquivo baseada em serialização de Java
 - Armazenar partida em andamento e recuperá-la

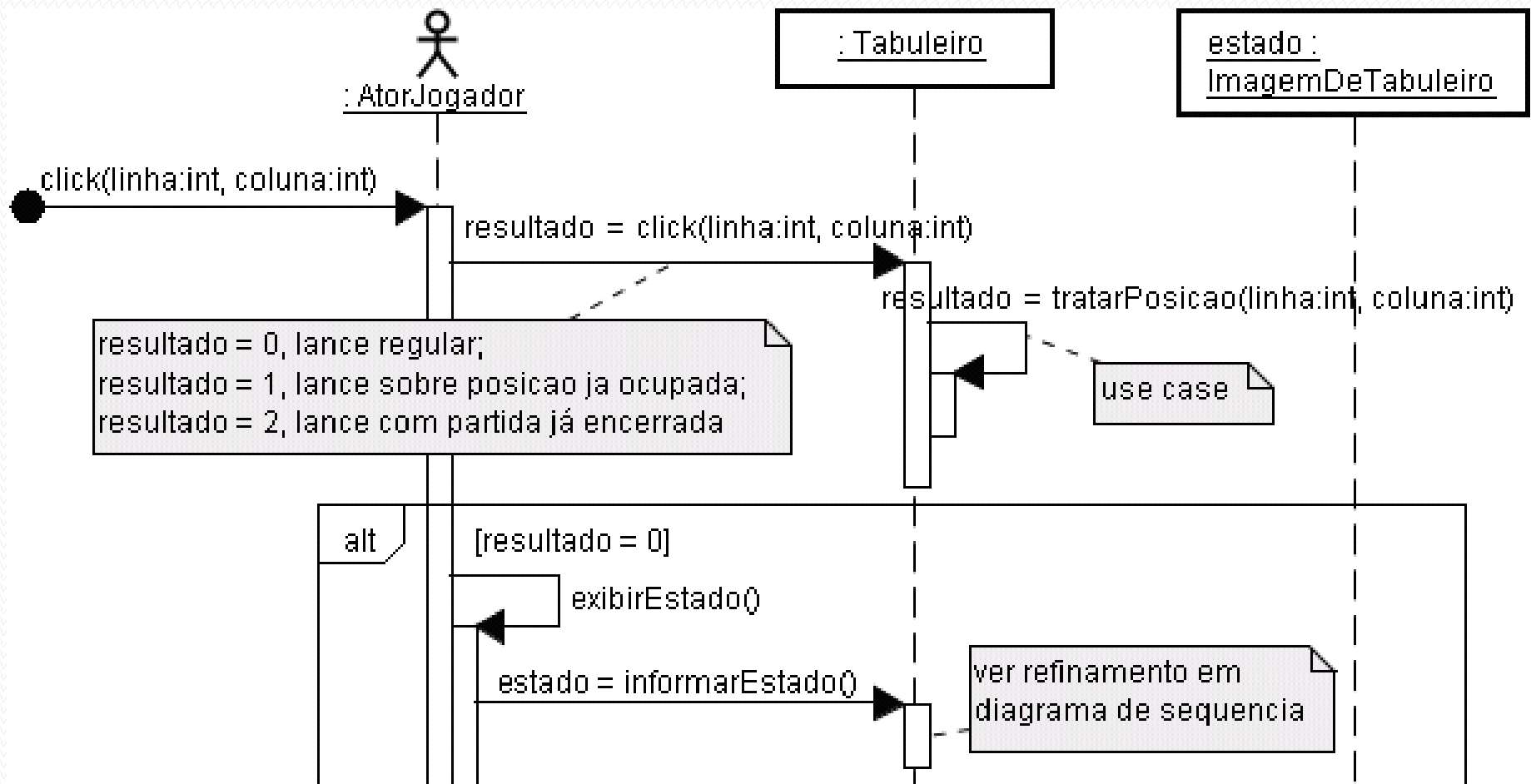
Exemplo 1: interface para o Jogo-da-velha

- A necessidade de interface aparece no início da modelagem, na forma do ator *AtorJogador*
- Ator não está associado a qualquer tecnologia específica
- Possui atributos e métodos identificados nas etapas anteriores
 - Não está especificado quem invoca esses métodos

Exemplo: ator modelando a interface (*caso de uso “iniciar partida”*)



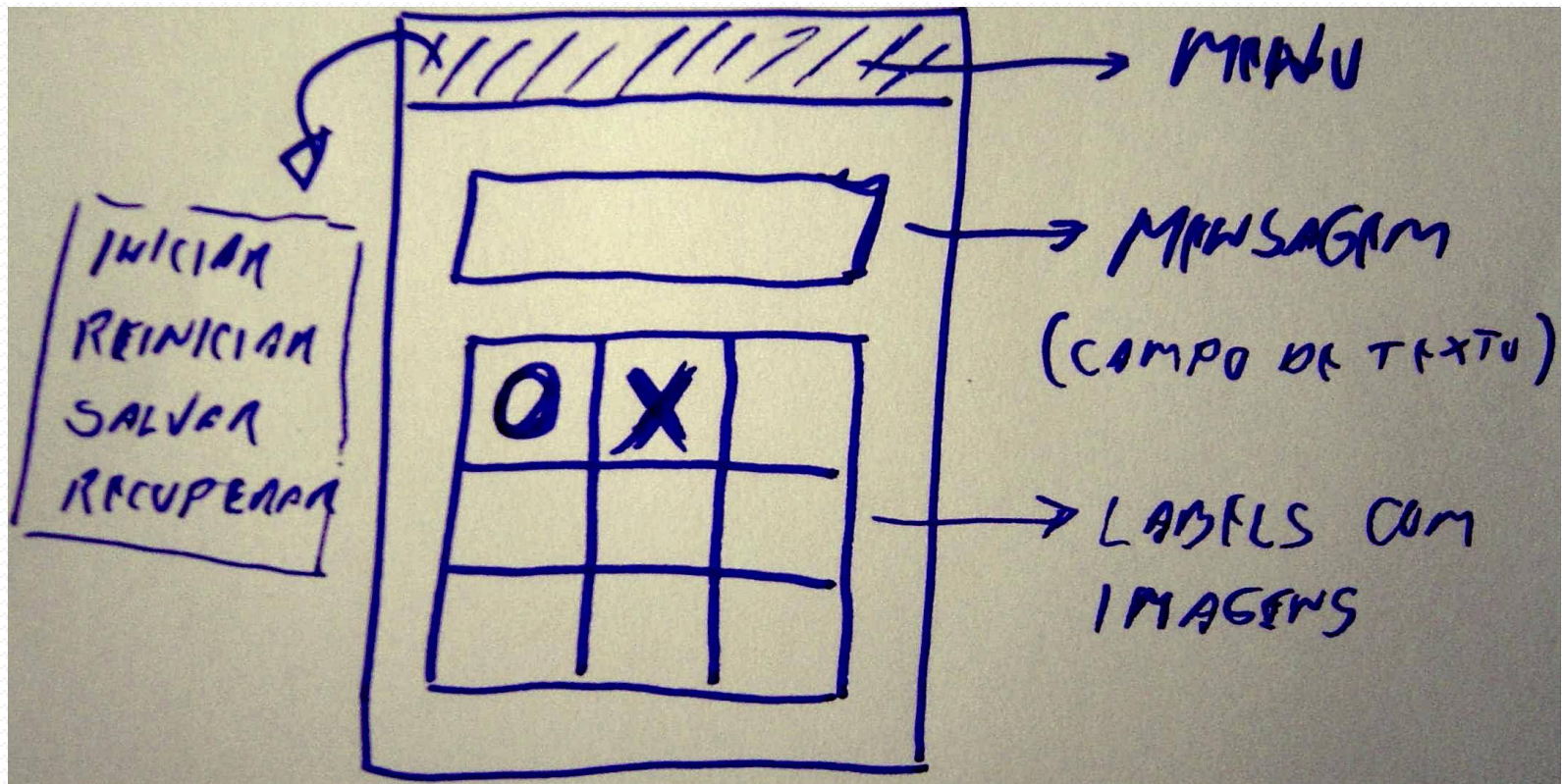
Exemplo: ator modelando a interface (caso de uso “procedimento de lance”)



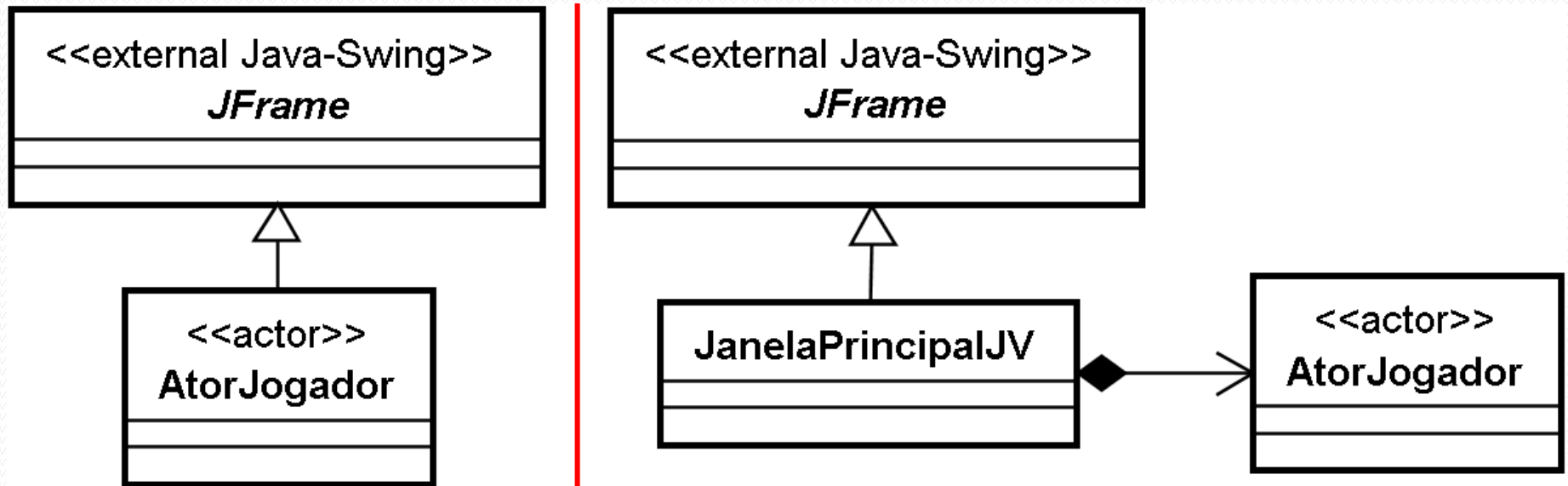
Exemplo: ator modelando a interface

<<actor>> AtorJogador
tab : Tabuleiro
<ul style="list-style-type: none">+ iniciarPartida() : void+ avaliarInterrupcao() : boolean+ obterIdJogador() : Stack+ estabelecerSequenciamento() : void+ definirOPrimeiro() : int+ obterSimboloJogador() : boolean+ reiniciarPartida() : void+ click(linha : int, coluna : int) : void+ exibir estado() : void+ notificarIrregularidade(codigo : int) : void

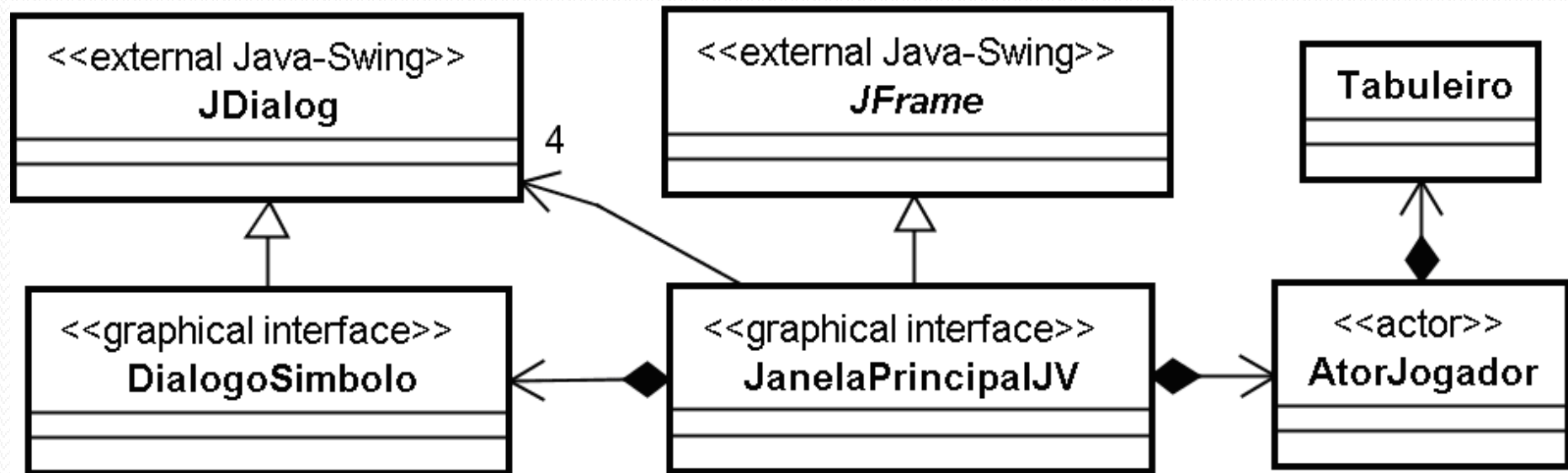
Começando a definir a interface



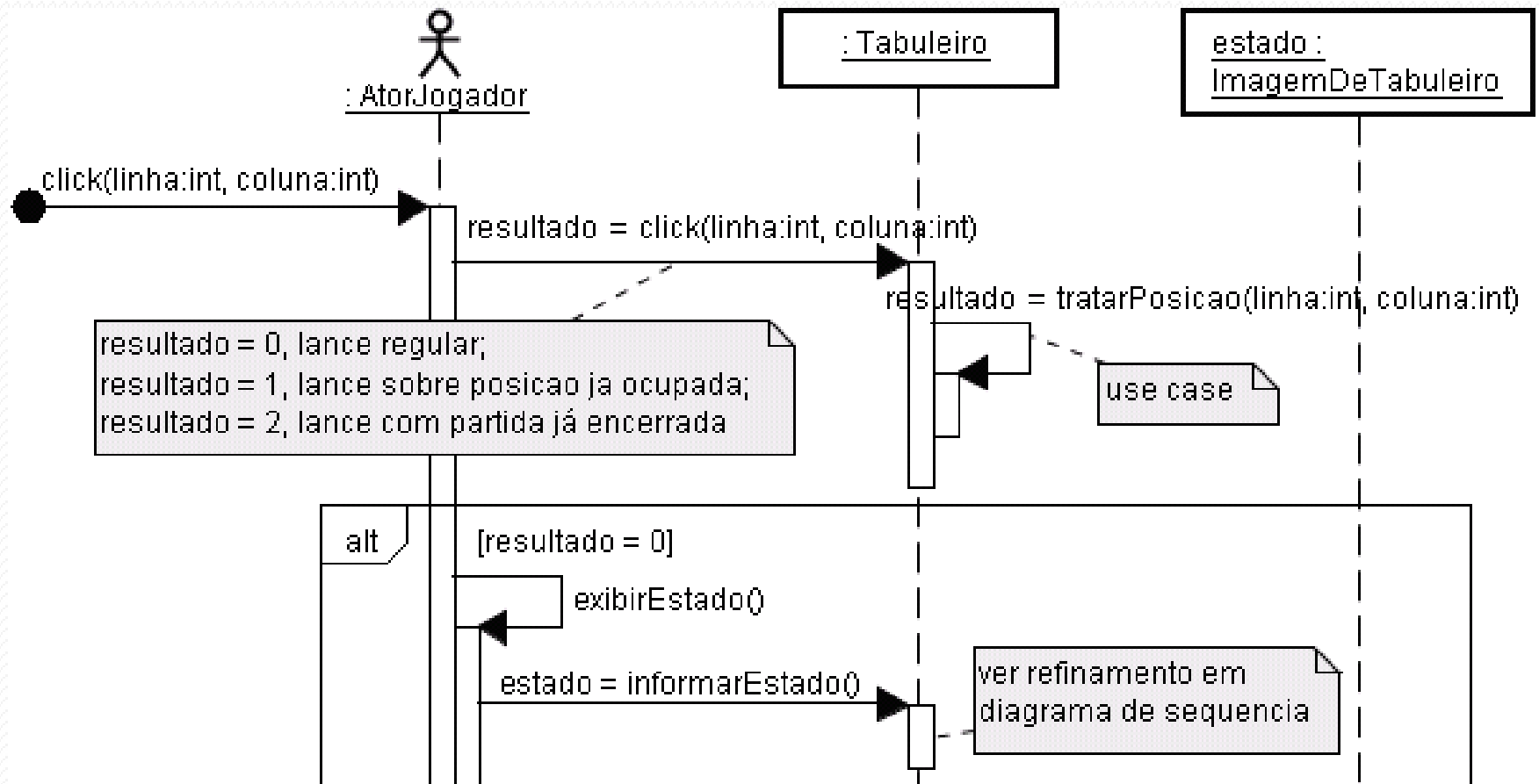
Java Swing → duas possibilidades



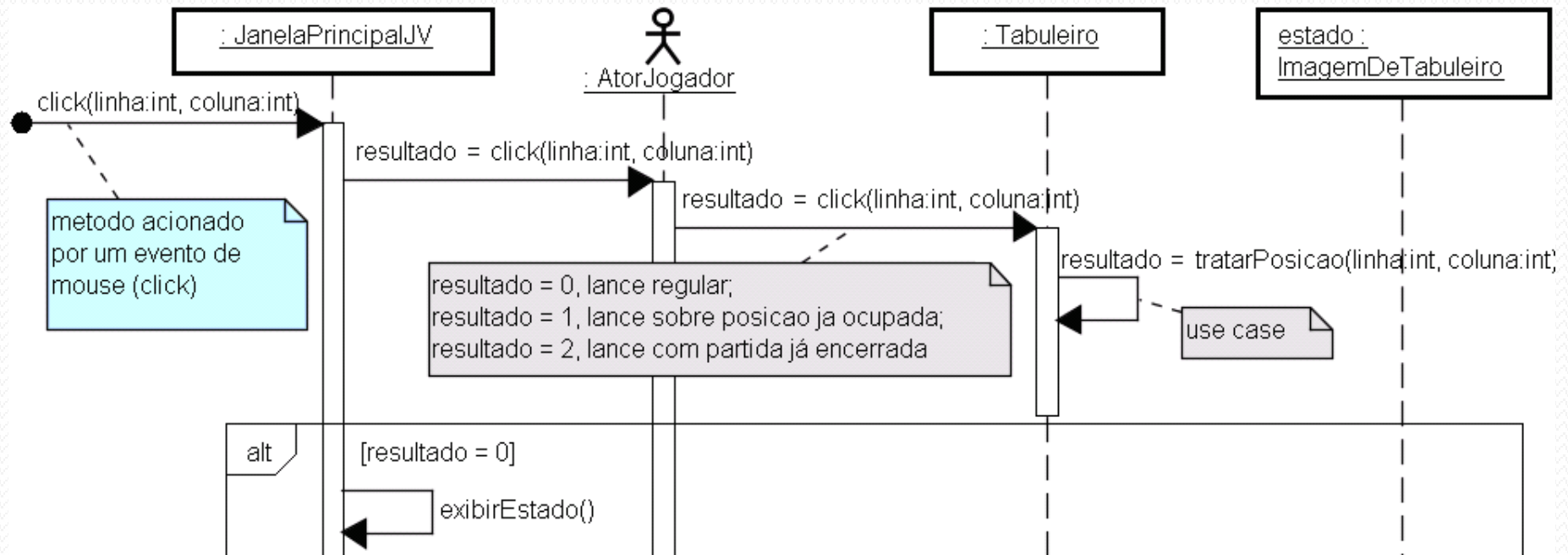
Interface → opção tecnológica adotada



Necessidade de revisar responsabilidades do ator (ex.: antes)



Necessidade de revisar responsabilidades do ator (ex.: depois)



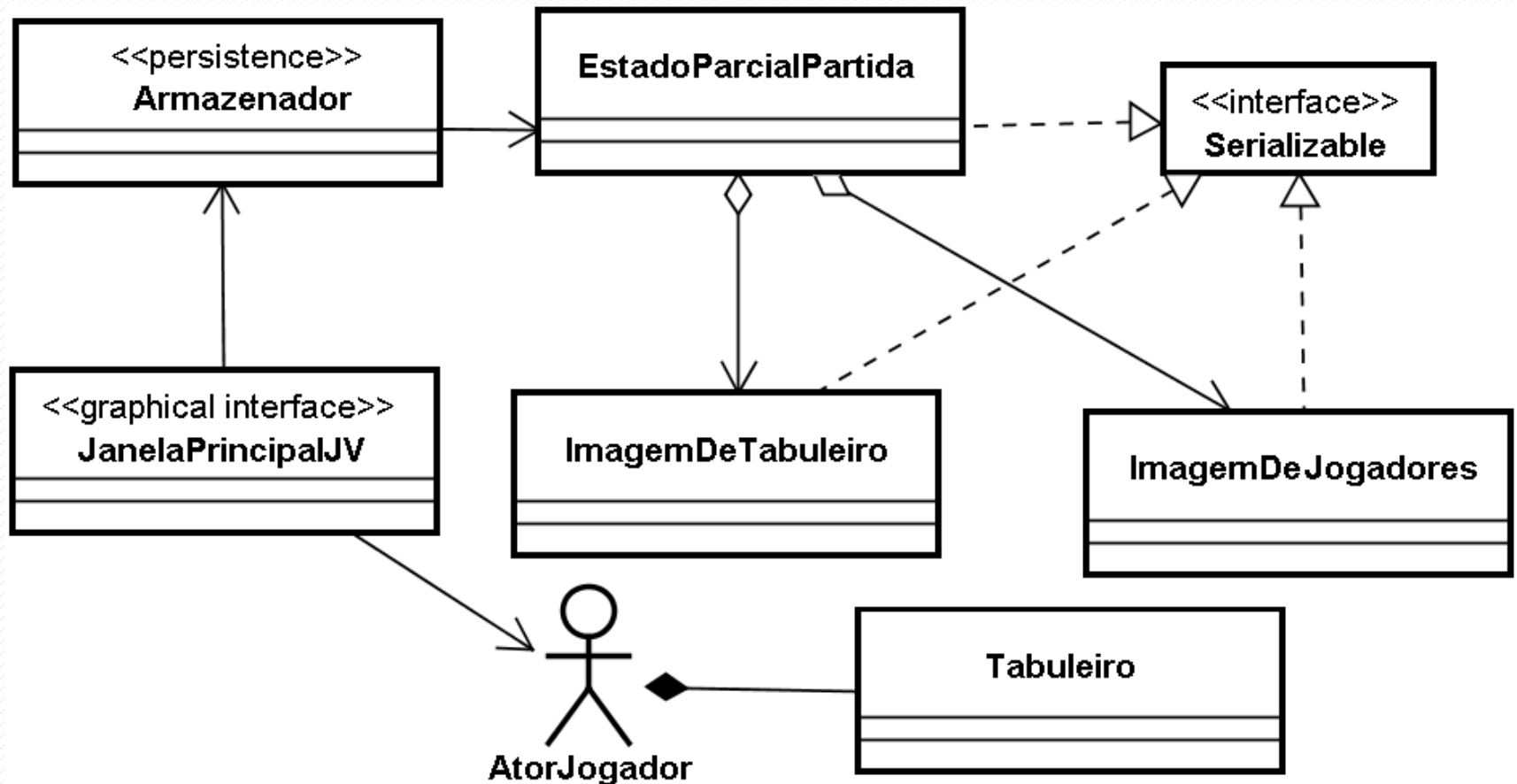
Exemplo 2: persistência para o Jogo-da-velha – serialização

- A persistência de dados não faz parte da modelagem até então desenvolvida
- Requisitos estabelecem essa necessidade → necessários dois novos casos
 - “*armazenar partida*”
 - “*recuperar partida*”

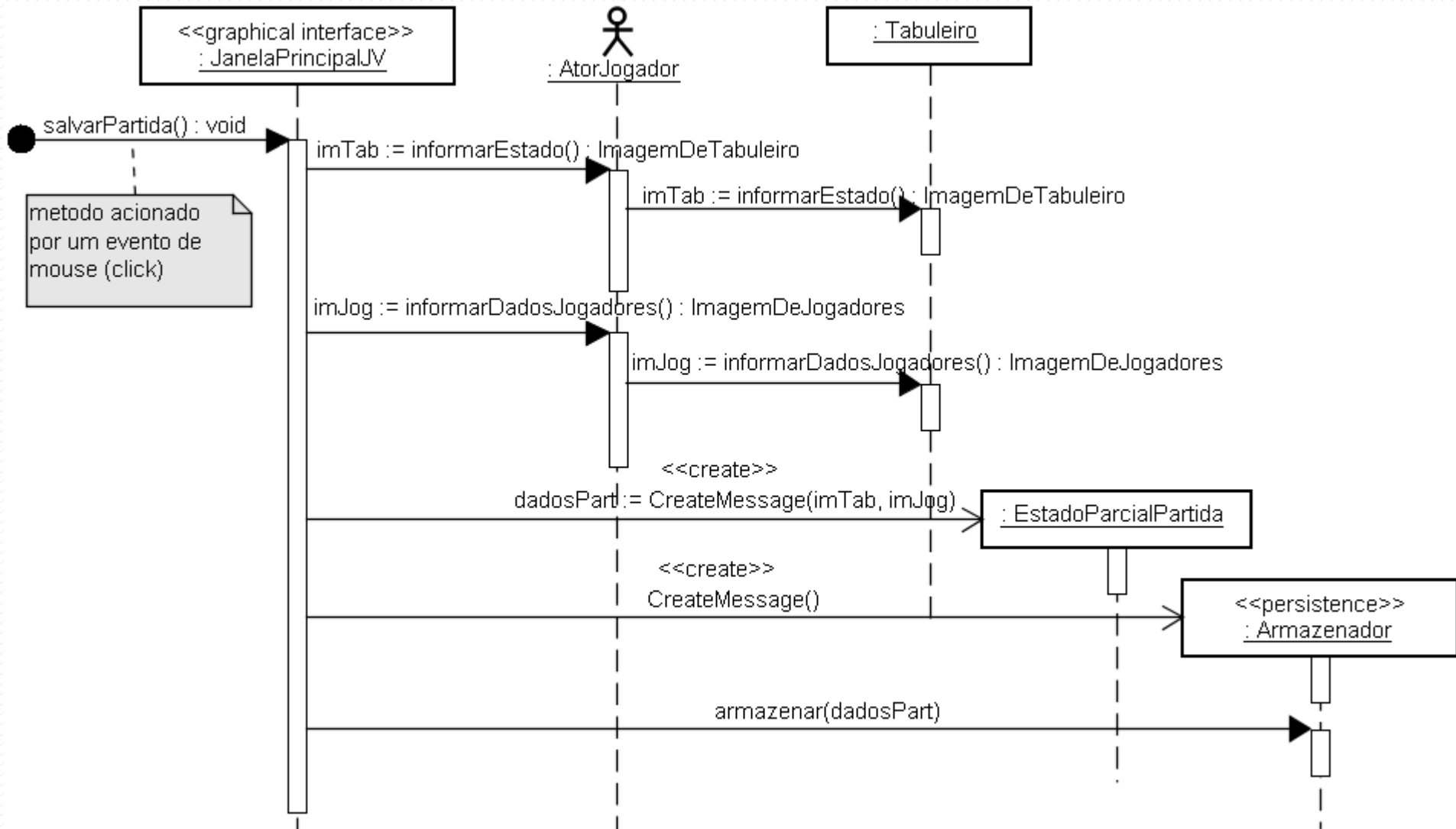
Exemplo 2: persistência para o Jogo-da-velha – serialização

- Necessidade de definir
 - O que vai ser armazenado
 - Quem vai armazenar

Estrutura da solução adotada



Exemplo → novo caso de uso



Sobre os exemplos apresentados

- Soluções bastante simples para a implementação de interface e de persistência
- Objetivo → ilustrar como soluções tecnológicas específicas se encaixam no processo de modelagem
 - Domínio do problema e domínio da solução não são universos isolados

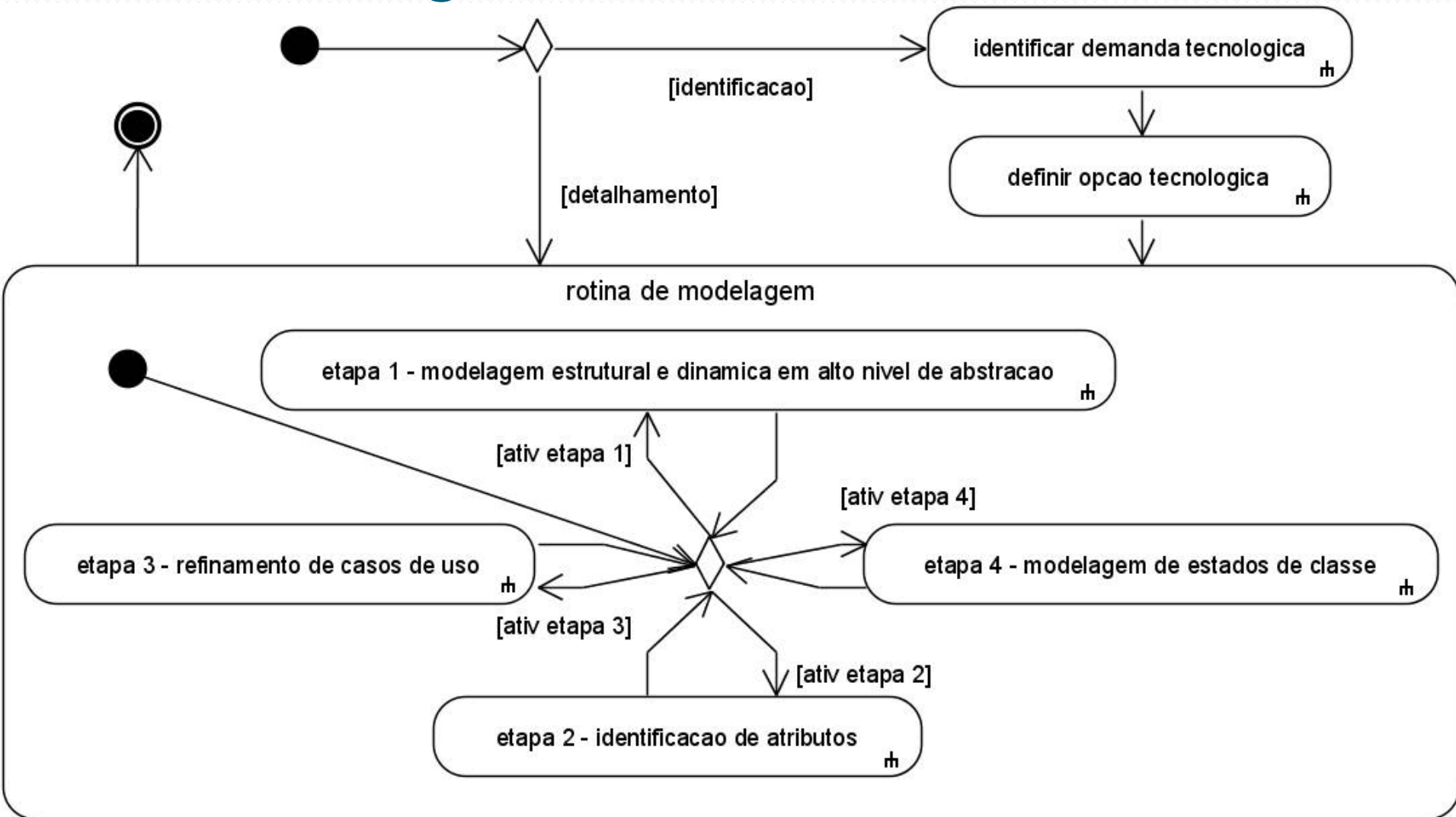
Generalizando a introdução do domínio da solução

- Demanda o conhecimento das tecnologias específicas adotadas
- É necessário ter a capacidade de incorporá-las ao projeto
 - Casamento harmonioso do domínio do problema com o domínio da solução

Resultados parciais após a quarta etapa do processo de modelagem

- Etapa 1 → Primeira modelagem estrutural e dinâmica de sistema
- Etapa 2 → Identificação de atributos
- Etapa 3 → Refinamento dos casos de uso
- Etapa 4 → Modelagem de estados das classes
- Etapa 5 → Inserção do domínio da solução
 - Identificar as demandas tecnológicas
 - Definir as opções tecnológicas
 - Incluí-las especificação

Sumário da quinta etapa do processo de modelagem



Considerações sobre esta aula

- Etapa 5 do processo de modelagem → Modelagem do domínio da solução
 - Início da etapa de projeto

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.