

Modelagem Dinâmica

Revisão: Modelagem Estática

- Diagrama de classes de análise representa a estrutura estática do sistema.
- Foco nas **informações relevantes** para a construção do sistema.
- Esse diagrama contém os conceitos (classes) importantes para o domínio do problema e as ligações entre eles.
- Cada classe é especificada em termos das informações que armazena (atributos) e dos serviços que provê (operações).

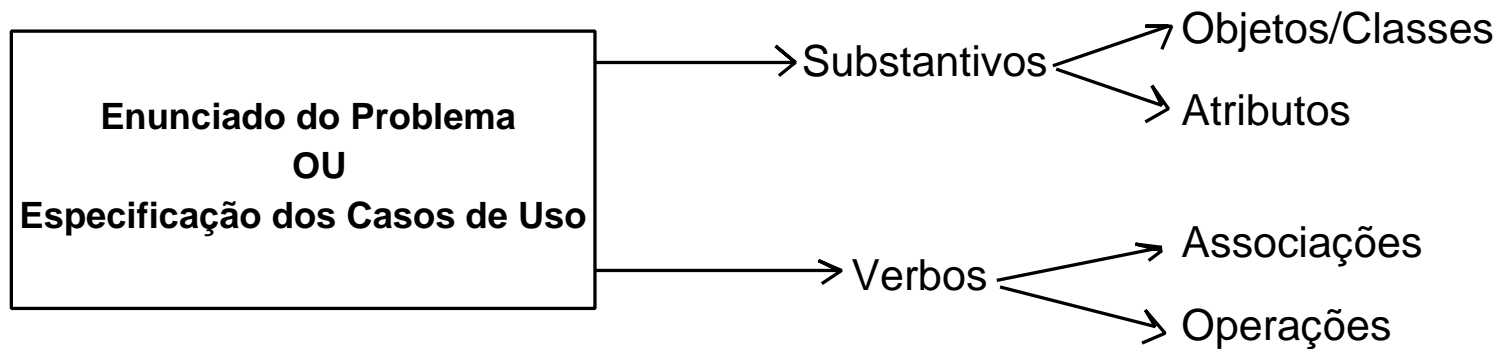
Modelagem Dinâmica

- Identifica e modela os aspectos do sistema de software que podem mudar durante a sua execução, devido à ocorrência de eventos.
- Foco no **comportamento** que o sistema deve apresentar.
- Usa os diagramas dinâmicos da UML (seqüência, colaboração, estados).
- Especifica uma versão inicial das interfaces públicas das classes de análise.
- Sub-etapa de Análise OO - Foco no domínio do problema!

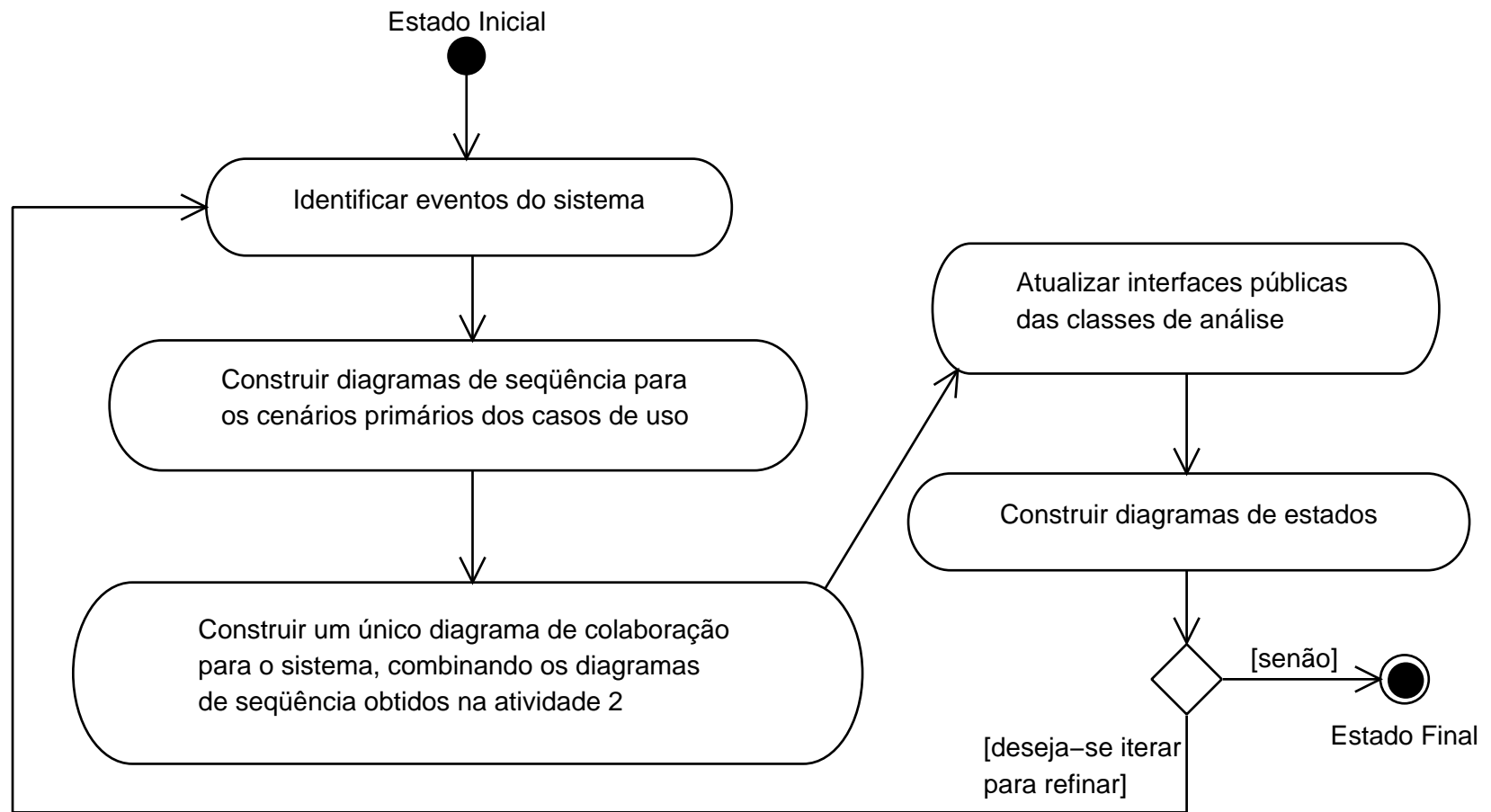
Eventos

- Ocorrências dignas de nota relativas ao sistema e envolvendo algum tipo de troca de informação.
- O evento não é a informação trocada e sim o fato de alguma informação ter sido trocada.
- O tipo de evento mais comum encontrado durante a análise é a interação entre um ator e o sistema.
 - Outros tipos também são possíveis.
- Modelamos o comportamento do sistema através de eventos e das ações executadas em resposta a eles.

Extração de Informações a partir do Enunciado do Problema



Atividades da Modelagem Dinâmica



Atividade 1: Identificar Eventos do Sistema

- Deve ser realizada uma nova análise textual nas especificações dos casos de uso, prestando-se atenção aos pontos nos quais trocas de informação ocorrem.
- Normalmente, esses pontos estão associados a verbos.
- Informações relevantes: verbos e os contextos nos quais aparecem.

Caso de Uso Emprestar Exemplar (I)

Fluxo Básico :

1. O cliente solicita empréstimo de um exemplar de alguma publicação (livro, periódico, tese ou manual), fornecendo o seu número de registro e o número de tombo da publicação desejada.
2. A atendente solicita o empréstimo ao sistema, fornecendo o código do cliente e o tombo da publicação
3. O sistema valida o cliente e verifica o seu status no sistema de cadastro (“Normal” ou “Suspendido”) através de seu número de registro.
(*<< include >>* Validar Usuário)

Caso de Uso Emprestar Exemplar (II)

Fluxo Básico :

4. O sistema verifica se existe algum exemplar disponível da publicação desejada.
5. Se o status do cliente for “Normal” e algum exemplar da publicação estiver disponível
 - 5.1. O sistema registra um novo empréstimo;
 - 5.2. O sistema verifica o período do empréstimo,
que depende do tipo de usuário - 7 dias para alunos e 15 para professores
 - 5.3. O sistema atualiza seu banco de dados
com a informação de que o exemplar não irá se encontrar na biblioteca até completar o período.
 - 5.4. O sistema comunica o sucesso do empréstimo.

Caso de Uso Emprestar Exemplar (III)

Fluxo Alternativo 1 :

No passo 5, se o usuário estiver suspenso, o sistema informa a atendente de sua proibição de retirar exemplares e o empréstimo não é realizado.

Fluxo Alternativo 2 :

No passo 5, se todas as cópias da publicação estiverem emprestadas ou reservadas, o sistema informa a atendente que não será possível realizar o empréstimo.

Recomendações

- Nem todo verbo está ligado à ocorrência de um evento.
- Alguns eventos são escritos de maneira implícita e precisam ser reescritos. Por exemplo, “O empréstimo é realizado” poderia ser reescrito como “O sistema informa à atendente que o empréstimo foi realizado”.
- É importante entender direito quais são os atores. No caso de uso Emprestar Exemplar, o único ator é a atendente.
 - O cliente é apenas uma informação que o sistema manipula.

Eventos Identificados (I)

- O cliente solicita empréstimo de um exemplar de alguma publicação;
- O cliente fornece o seu número de registro e o número de tomo da publicação;
- A atendente solicita o empréstimo ao sistema;
- A atendente fornece o número de registro do cliente e o número de tomo da publicação;
- O sistema valida o cliente e verifica o seu status;

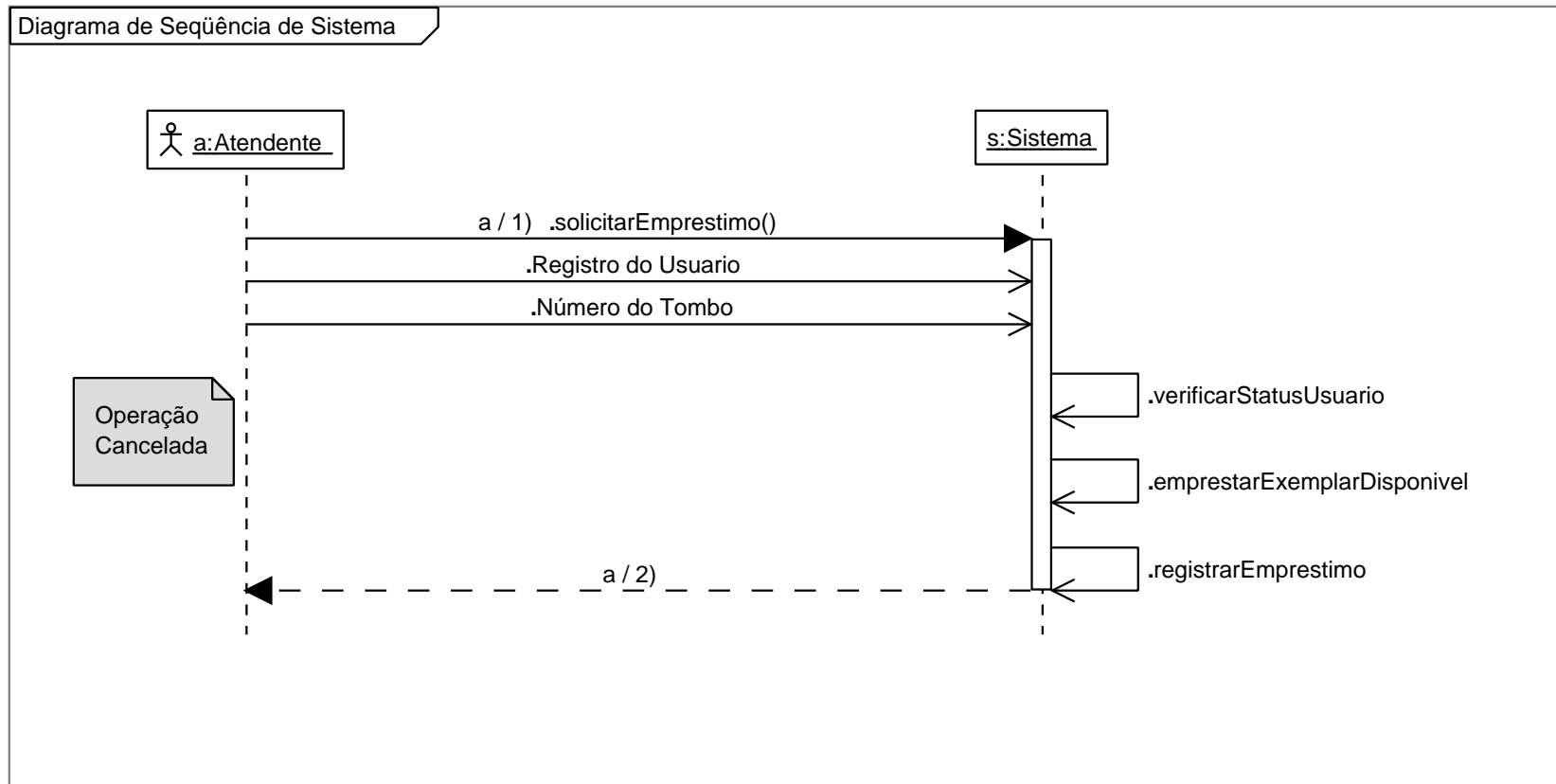
Eventos Identificados (II)

- O sistema verifica se existe algum exemplar disponível da publicação desejada;
- O sistema registra um novo empréstimo;
- O sistema verifica o período do empréstimo;
- O sistema atualiza seu banco de dados;
- O sistema comunica o sucesso do empréstimo;
- (Alternativo) O sistema informa a atendente de sua proibição de retirar exemplares;
- (Alternativo) O sistema informa a atendente que não será possível realizar o empréstimo.

Atividade 2: Construir Diagrama de Seqüência

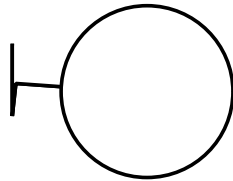
- Primeiro constrói-se o diagrama de seqüência de sistema relativo a cada caso de uso, usando-se os eventos identificados na Atividade 1.
- Em seguida refina-se esse diagrama, substituindo-se o objeto Sistema pelas classes de análise de fronteira, controle e entidade que o materializam.

Diagrama de Seqüência de Sistema para Emprestar Exemplar

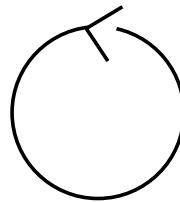


Relembrando: Tipos de Classes de Análise

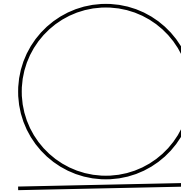
- Para cada tipo de classe de análise, o RUP define um estereótipo.



Classes de
Fronteira
<<boundary>>



Classes de
Controle
<<control>>

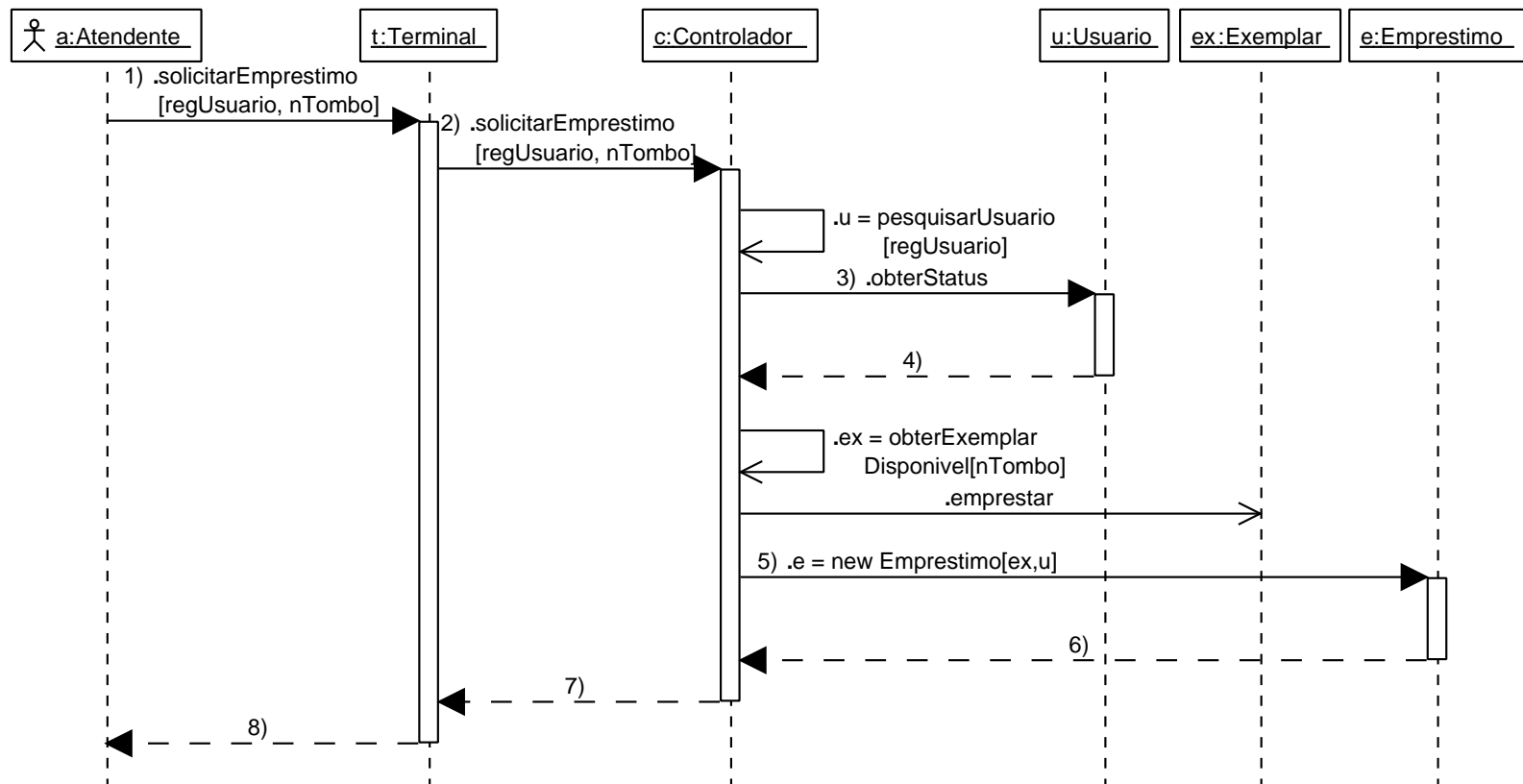


Classes de
Entidade
<<entity>>

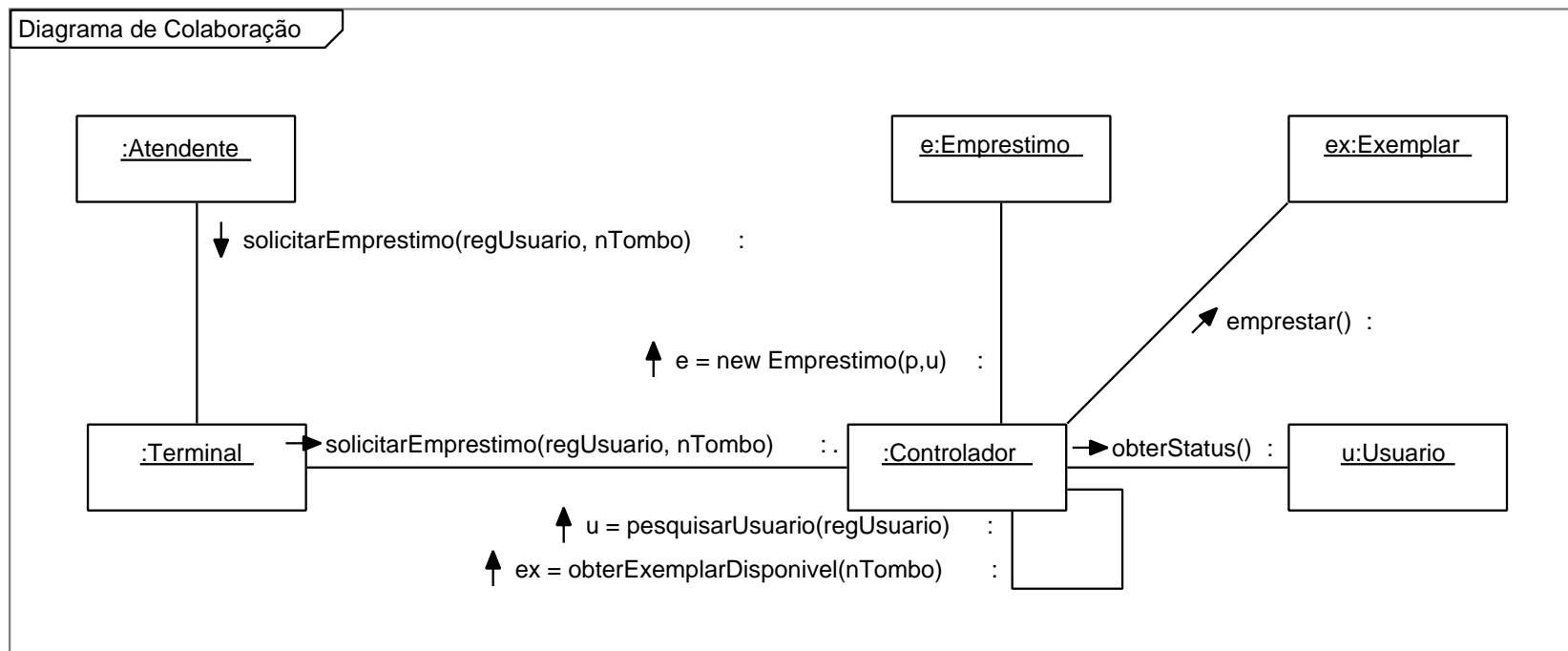
- Interações devem ser organizadas da seguinte maneira:
Fronteira → Controle → Entidade

Diagrama de Seqüência Refinado para Emprestar Exemplar

Diagrama de Seqüência Refinado



Atividade 3: Construir um Diagrama de Colaboração para o Sistema



Atividade 4: Atualizar Interfaces Públicas das Classes de Análise

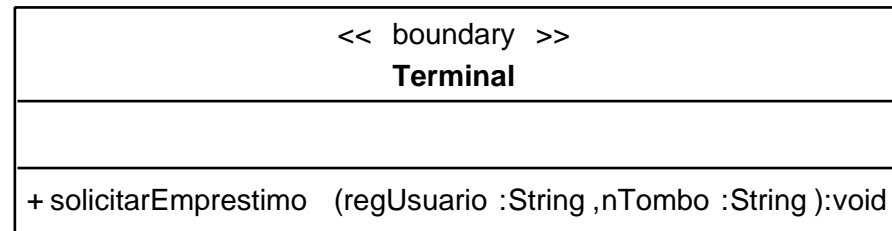
- Neste passo, devemos identificar, nos diagramas de seqüência, todos os eventos do sistema.
- Estes eventos devem ser então associados às classes de análise.
- Para cada evento associado a uma classe de análise, é necessário verificar se as instâncias dessa classe produzem o evento ou o recebem.

Associação entre Eventos e Classes de Análise (I)

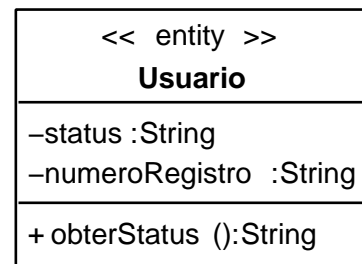
Classe	Eventos Recebidos	Eventos Produzidos
Terminal	“solicitarEmprestimo (regUsuario, nTombo)”	“solicitarEmprestimo (regUsuario, nTombo)”
Usuario	“obterStatus()”	
Exemplar	“ocupar()”	
Emprestimo	“new Emprestimo(ex, u)”	

Operações da Classe Terminal e Usuario

Operações da Classe Terminal



Operações da Classe Usuario



Operações da Classe Exemplar e Emprestimo

Operações da Classe Exemplar

Exemplar

-status :String
-codFisico :String
+ ocupar ():void

Operações da Classe Emprestimo

<< entity >>

Emprestimo

-dataEmprestimo :Date
-dataDevolucao :Date

<< create >> + Emprestimo (ex:Exemplar ,u:Usuario):Emprestimo

Associação entre Eventos e Classes de Análise (II)

Classe	Eventos Recebidos	Eventos Produzidos
Controlador	<p>"solicitarEmprestimo (regUsuario, nTombo)"</p> <p>"u = pesquisarUsuario(regUsuario)"</p> <p>"ex = obterExemplarDisponivel (nTombo)"</p>	<p>"u = pesquisarUsuario(regUsuario)"</p> <p>"obterStatus()"</p> <p>"ex = obterExemplarDisponivel (nTombo)"</p> <p>"ocupar()"</p> <p>"e = new Emprestimo(ex,u)"</p>

Operações da Classe Controlador

Operações da Classe Controlador

<< control >>

Controlador

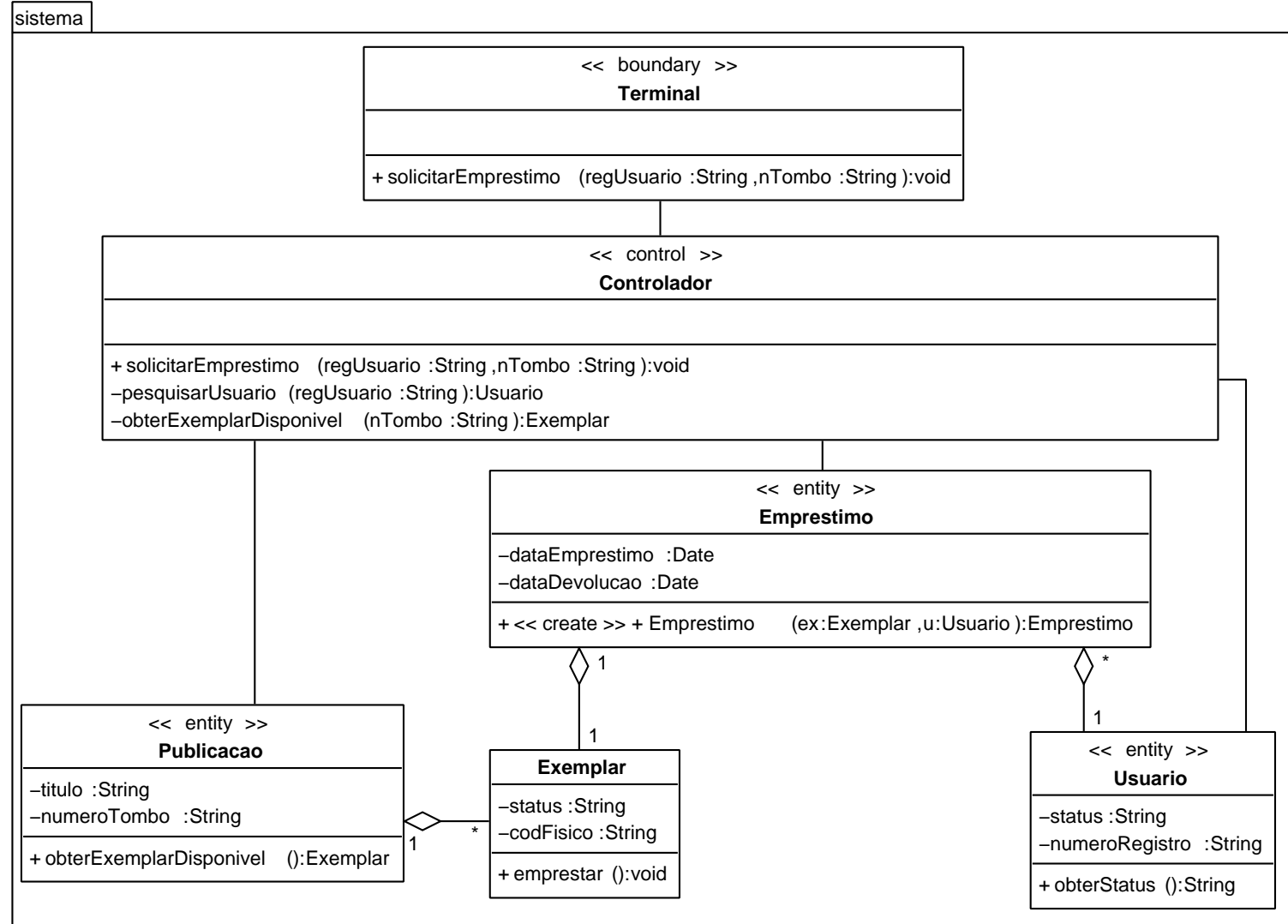
+ solicitarEmprestimo (regUsuario :String ,nTombo :String):void
-pesquisarUsuario (regUsuario :String):Usuario
-obterExemplarDisponivel (nTombo :String):Exemplar

Atualizar o Diagr. de Classes de Análise (I)

- Depois de associar eventos às classes de análise, é necessário atualizar o diagrama de classes, para que passe a incluir essa informação.
- Os eventos recebidos por uma classe se transformam em operações de sua interface pública.
- Eventos produzidos por um objeto podem indicar tanto uma resposta para um evento recebido quanto o fato de que uma classe usa algum serviço oferecido por outra.

Atualizar o Diagr. de Classes de Análise (II)

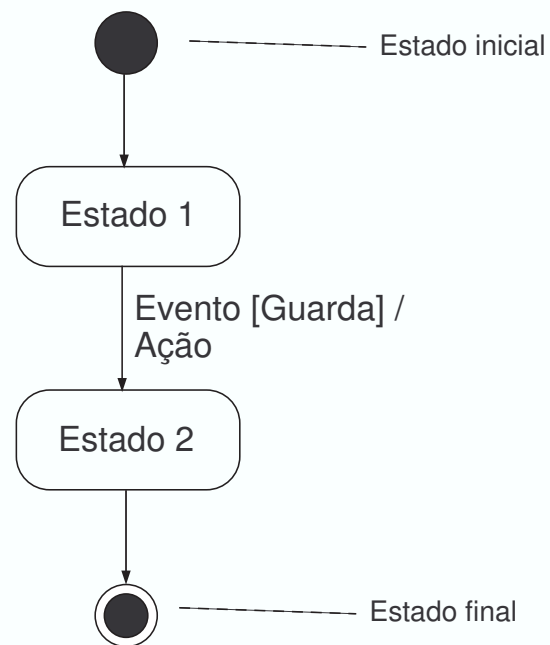
Diagrama de Classes de Análise



Atividade 5: Construir Diagramas de Estados

- Máquinas de estados podem ser usadas para especificar o comportamento de sistemas e classes
- O diagrama de estados é a representação gráfica de máquinas de estados adotado pela UML.
- Diagramas de estados ilustram os eventos de interesse e os estados de um objeto, além da maneira como esse objeto reage a eventos.

Um Exemplo de Diagrama de Estados



Usos de Diagramas de Estados

- Diagramas de estados normalmente são usados para modelar:
 - (i) a seqüência legal de eventos externos que são reconhecidos e tratados por um sistema, no contexto de um caso de uso;
 - (ii) o conjunto de todos os estados e transições do sistema, ao longo da execução de todos os seus casos de uso;
 - (iii) o comportamento de classes complexas que dependem de algum tipo de estado.

Estados da Classe Usuário

1. sem livros e não suspenso;
2. com livros e não suspenso;
3. suspenso.

Algumas Transições entre Estados da Classe Usuário

1. um usuário sem livros e não suspenso pode emprestar um ou mais livros;
2. um usuário com livros e não suspenso pode emprestar ou devolver um ou mais livros;
 - um livro devolvido pode ou não estar atrasado;
 - em caso afirmativo, o usuário é suspenso;
3. um usuário suspenso pode apenas devolver livros;
4. o período de suspensão de um usuário suspenso pode terminar.

Diagrama de Estados da Classe Usuário

