

Modelagem de Software - Parte II

Prof. Pedro Henrique Dias Valle

Diagramas da UML 2.0

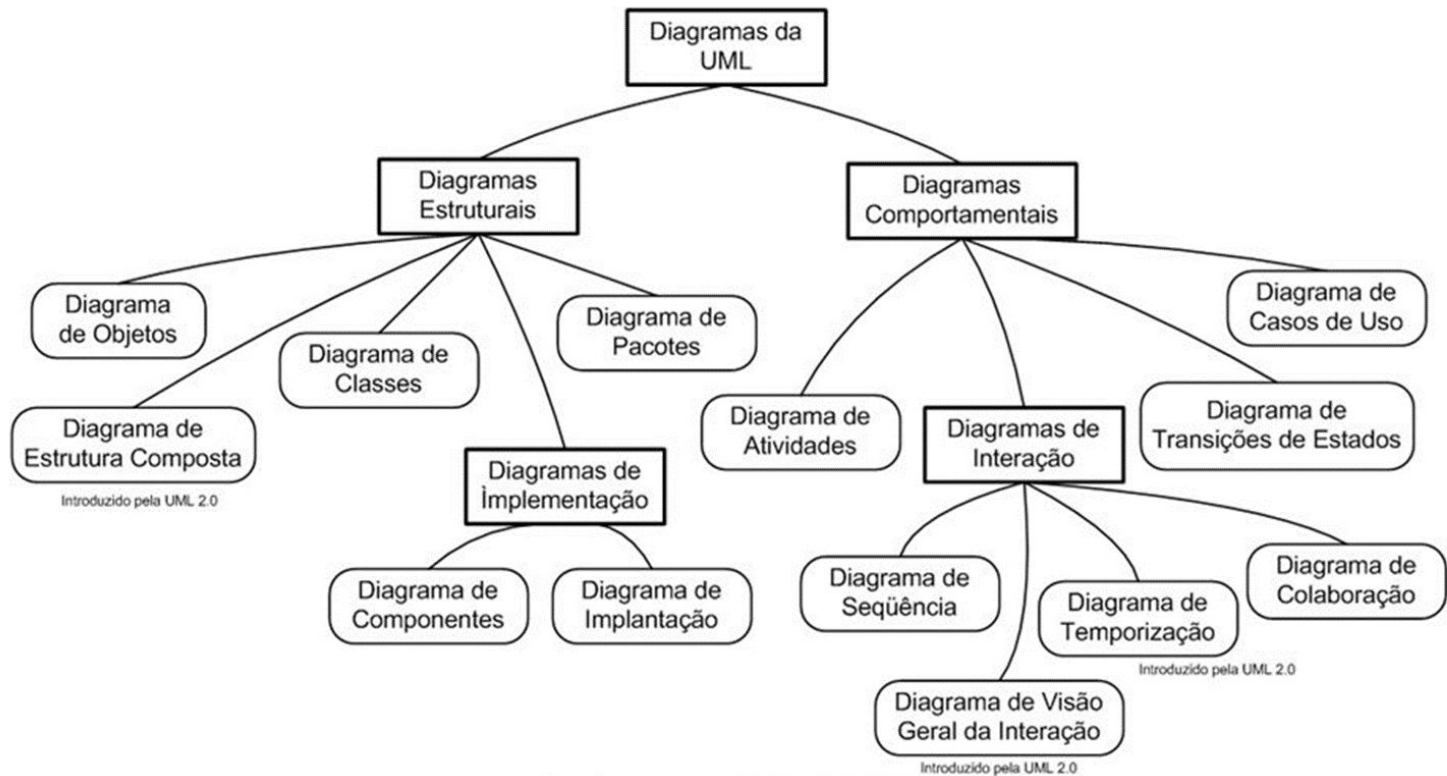


Diagrama de Estados

Diagrama de Estados

- O diagrama mais antigo da UML
 - Foi criado nos anos 60
- Ferramenta útil para mostrar o ciclo de vida de um objeto
- Em um sistema real, somente algumas poucas classes demandam o uso de um DTE
 - Classes críticas
 - Classes que se comportam de forma diferente em função dos seus estados internos

A 1 km de distância...

- Caixas representando os estados
- Linhas representando as transições

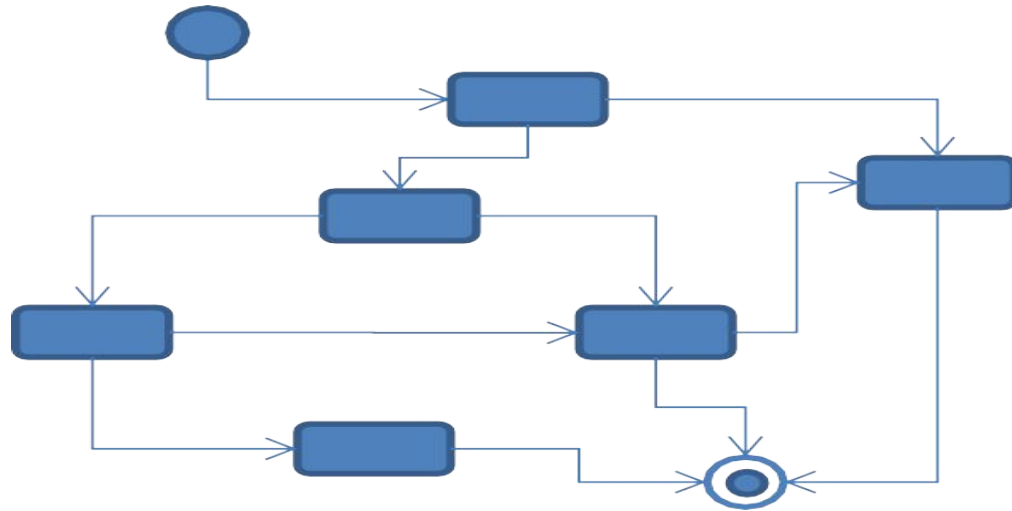


Diagrama de Transição de Estados

- O Diagrama de Estados descreve o comportamento de objetos em relação a **eventos**;
- Apresenta uma **sequência de estados** e ações que ocorrem durante a vida do objeto, em resposta a eventos;
- O Diagrama de Estados mostra o **ciclo de vida** de um objeto, ou seja, seus estados, os eventos que causam a transição de um estado para outro e as ações que resultam de uma mudança de estado.

Diagrama de Transição de Estados

- Por que devo saber do ciclo de vida (estados) de objetos do sistema?
- Alguns exemplos:
 - Uma conta bancária inativa pode voltar à condição de ativa?
 - Uma ordem de serviço depois de emitida sempre é executada ou pode ser cancelada?
 - Um paciente de um hospital, depois de morto, pode ter atendimento atribuído a ele?
- Questões desse tipo devem ser respondidas em tempo de análise e projeto.

Diagrama de Transição de Estados

Estado:


- É uma condição detectada durante o ciclo de vida de um objeto quando ele:
 - Satisfaz alguma condição;
 - Realiza alguma atividade;
 - Aguarda por algum evento;
- O Estado de um objeto é uma das possíveis condições nas quais ele pode existir durante a sua vida.

A 1 metro de distância... dos estados

- Exemplo
 - Um ventilador simples pode ter os estados **desligado** e **ligado**
 - Um ventilador mais elaborado pode ter os estados **desligado**, **ventilando fraco**, **ventilando médio** e **ventilando forte**
 - Ao apertar um botão, em função do estado atual, o ventilador pode desligar ou ventilar forte

Diagrama de Transição de Estados

- Um estado é representado graficamente como um retângulo com cantos arredondados;
- O nome do estado é colocado no centro do mesmo.



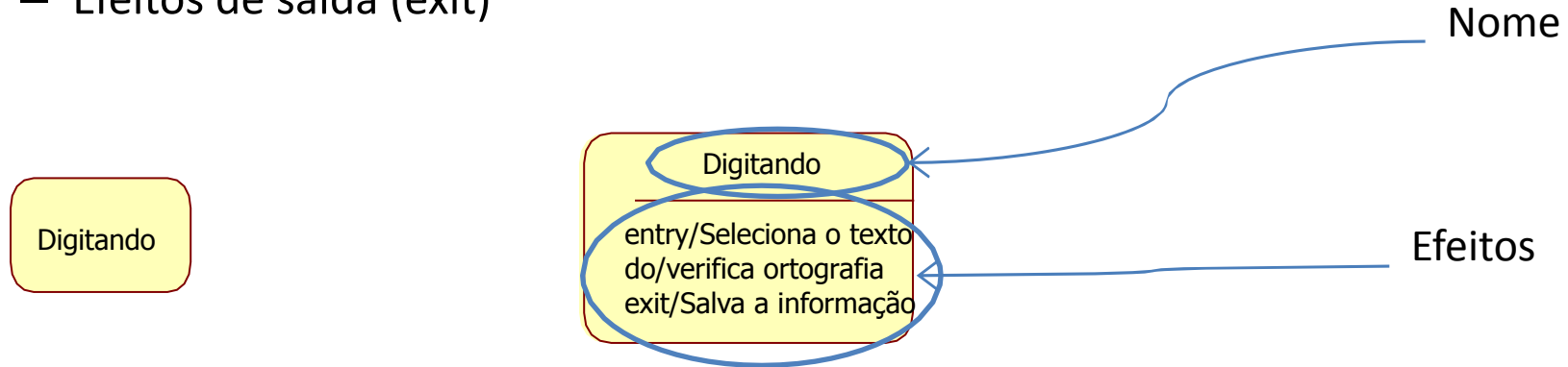
A diagram illustrating two states in a state transition diagram. It consists of two yellow rounded rectangles. The left rectangle contains the text 'Aguardando Pagamento' and the right rectangle contains the text 'Linha Ocupada'.

Aguardando
Pagamento

Linha
Ocupada

Diagrama de Transição de Estados

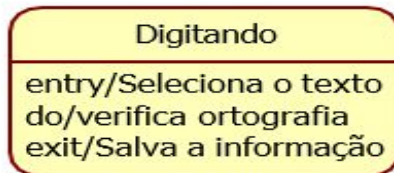
- Os estados são representados por caixas contendo
 - Nome
 - Efeitos de entrada (entry)
 - Efeitos de execução (do)
 - Efeitos de saída (exit)



A 1 centímetro de distância dos efeitos....

- Em análise
 - Os efeitos de entrada, execução e de saída são descritos em linguagem natural
- Em projeto
 - Os efeitos de entrada, execução e de saída são transformados em operações

Análise



Projeto

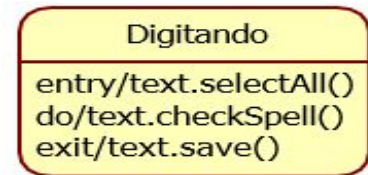


Diagrama de Transição de Estados

Compartimentos:

- Um estado pode ser opcionalmente subdividido em compartimentos:
 - **Compartimento de nome:**
 - Armazena o nome do estado, como uma string;
 - **Compartimento de transições internas:**
 - Armazena uma lista de ações ou atividades internas que são executadas enquanto o objeto se apresenta no estado em questão.

Diagrama de Transição de Estados

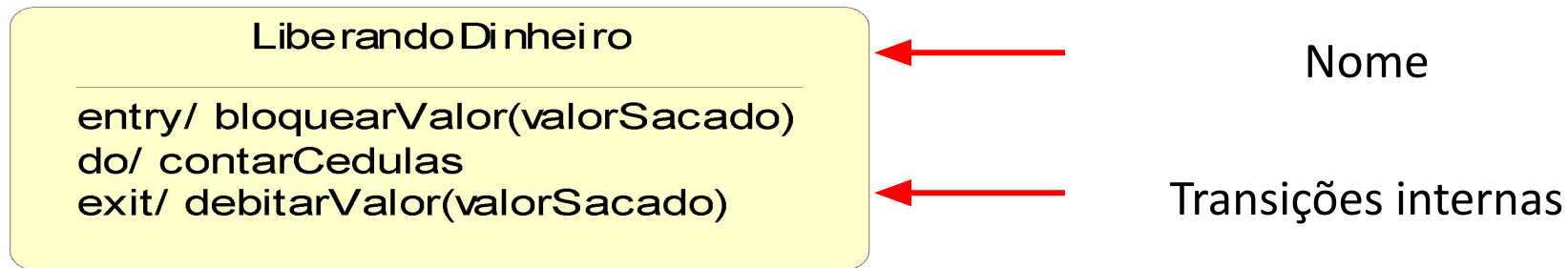


Diagrama de Transição de Estados

Estado inicial:

- Indica o local de início do diagrama de estados;
- Cada diagrama deve possuir um e apenas um estado inicial (exceto para diagramas aninhados);
- Representação:
 - Um círculo preenchido.

Estado Inicial

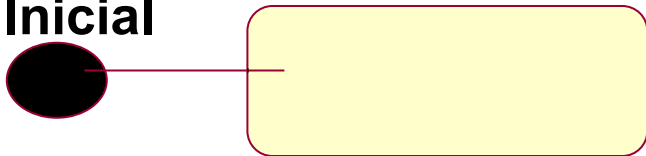


Diagrama de Transição de Estados

Estado final:

- Indica o fim da existência de um objeto ou o final da realização de uma atividade (para diagramas aninhados);
- Não aceita transição de saída
- Um diagrama pode ter múltiplos estados finais.



Diagrama de Transição de Estados

Transição ou Evento:

- Relacionamento entre dois estados;
- Indica que haverá uma mudança de estado e que determinadas ações serão executadas;
- Pode ocorrer como resultado de algum evento;
- Pode ter que satisfazer a alguma condição;
- O estado sucessor pode ser o estado original;
- É representada com uma seta do estado de origem para o estado de destino.

Transições de Estado

- As transições determinam a troca de estados em função de um determinado evento
- Exemplo
 - A partir do **estado desligado**, caso o botão seja apertado, o ventilador vai para o estado **ventilando forte**
 - A partir do estado **ventilando forte**, caso o botão seja apertado, o ventilador vai para o estado **ventilando médio**
 - A partir do estado **ventilando médio**, caso o botão seja apertado, o ventilador vai para o estado **ventilando fraco**
 - A partir do estado **ventilando fraco**, caso o botão seja apertado, o ventilador vai para o **estado desligado**

Transições de Estado

Estados x atributos:

- O estado de um objeto pode ser caracterizado pelo valor de um ou mais atributos.

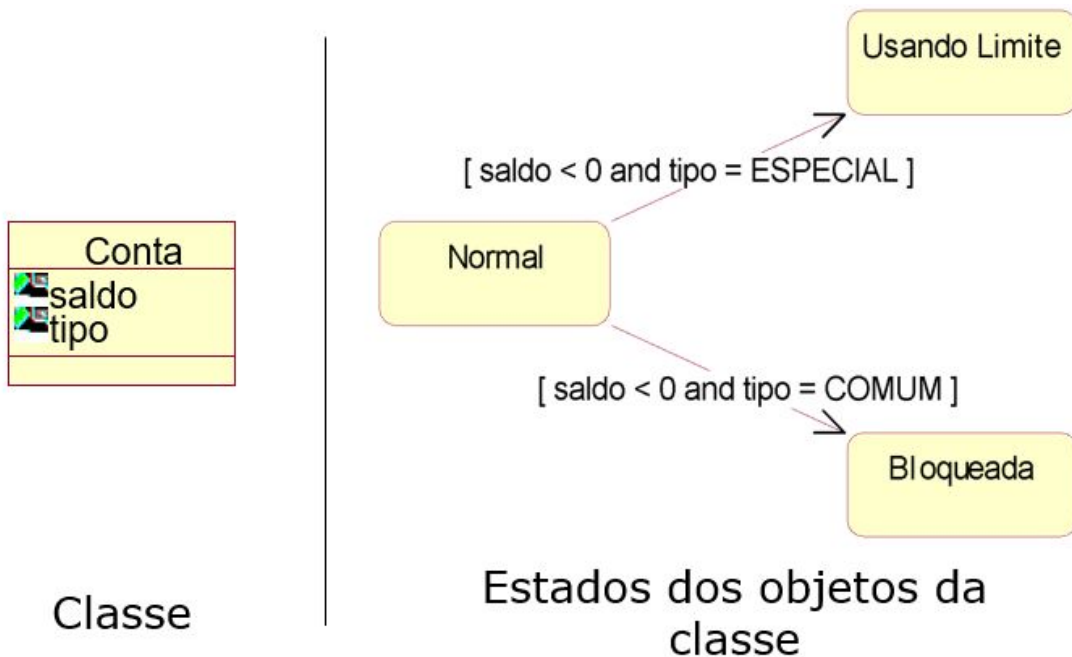
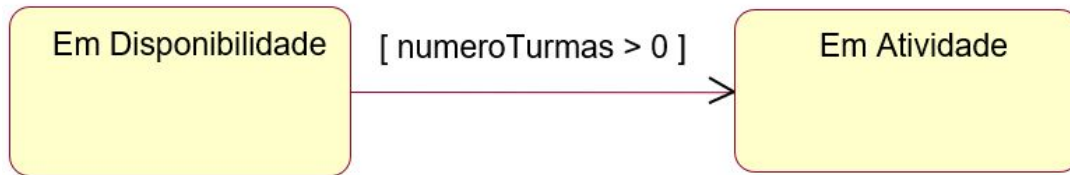


Diagrama de Transições de Estado

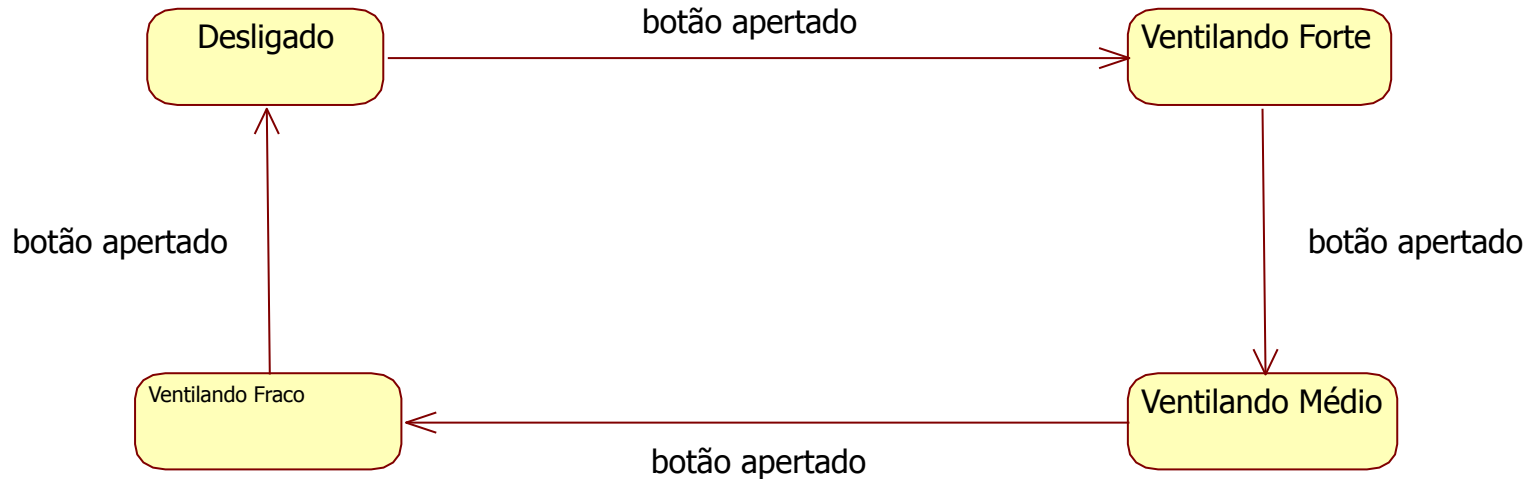
- Estados podem ser caracterizados pela existência de um relacionamento com outro objeto;



Diagrama de estados da classe Professor:

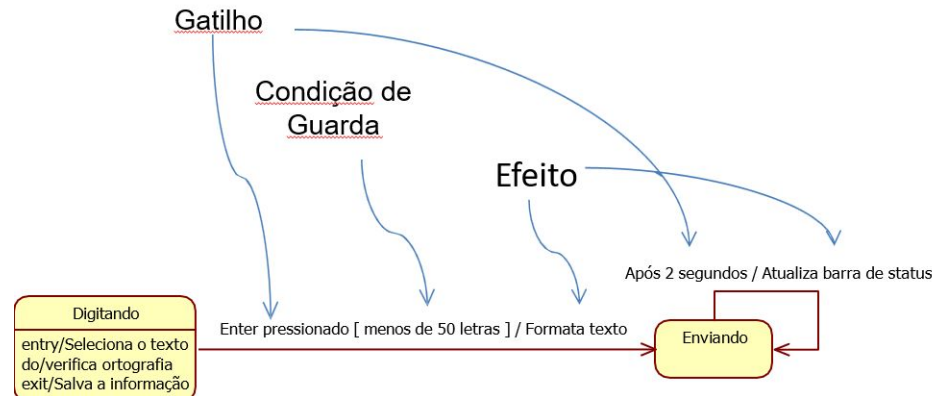


A 1 metro de distância... das transições



A 1 metro de distância... das transições

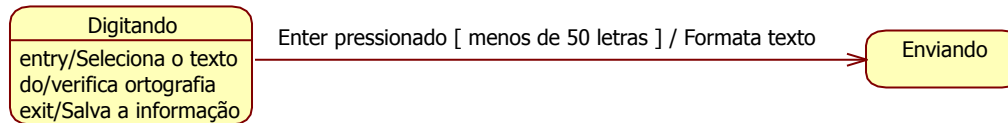
- As transições são representadas por linhas, contendo
 - Gatilho (trigger)
 - Condição de guarda (guard)
 - Efeito
- Sintaxe: GATILHO [CONDIÇÃO] / EFEITO



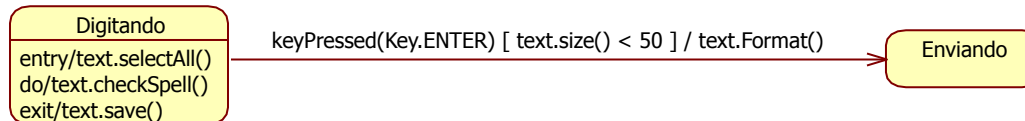
A 1 metro de distância... das transições

- Em análise
 - Utilização de linguagem natural para gatilhos, condição de guarda e efeitos
- Em projeto
 - Gatilhos transformados em eventos
 - Condições de guarda transformadas em expressões booleanas
 - Efeitos transformados em operações

Análise

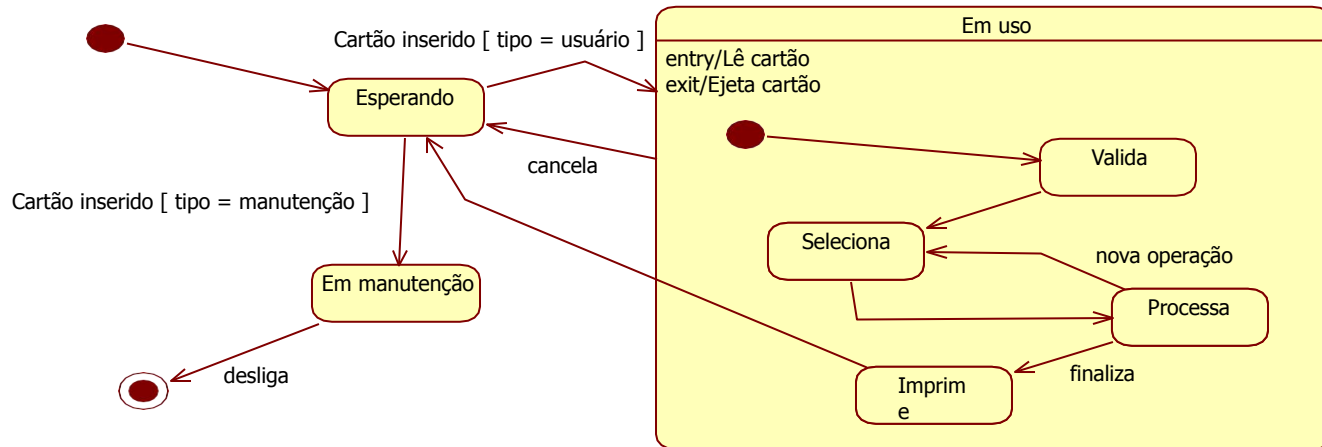


Projeto



A 1 metro de distância... das transições

- Para viabilizar uma melhor organização do diagrama, podem ser criados estados compostos
- Estados compostos permitem a descrição de um diagrama interno de transição de estados



Algumas Dicas

- Crie diagrama de estados apenas para classes com um comportamento dinâmico significativo;
- Faça inicialmente um diagrama simples, e complique somente se for necessário
- O que aparece em quase todo diagrama
 - Estados com nome
 - Transições com gatilho
- Diagrama de Estados podem ser utilizados para derivar Casos de Teste do software.

Diagrama de Transições de Estado

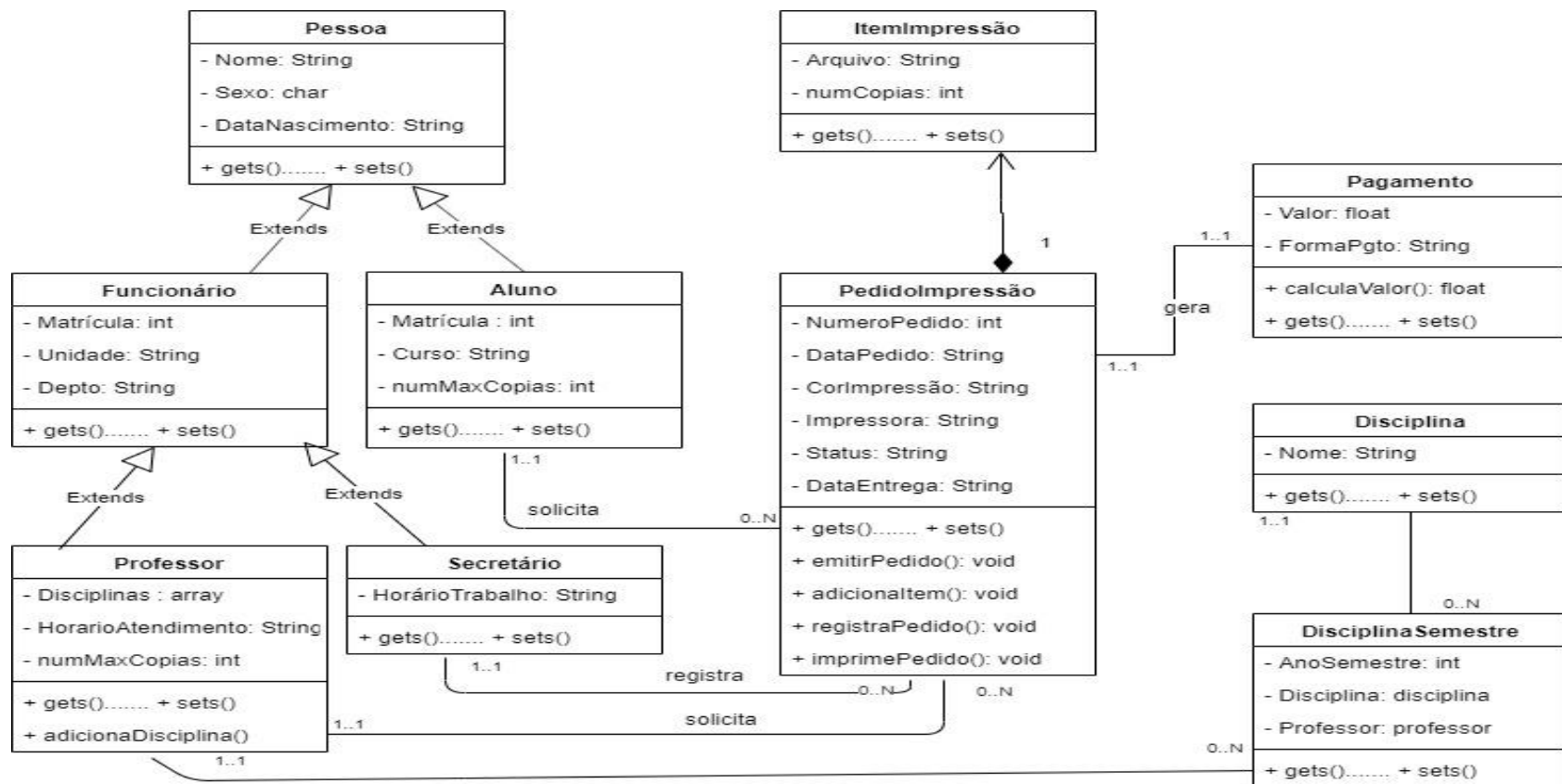
Transições internas:

- Atividades associadas ao estado e que devem ocorrer na entrada, na permanência, ou na saída do estado;
- São associadas com qualquer transição entrando ou saindo do estado;
- São mostradas dentro do ícone do estado precedidas pela palavra “entry”, “exit” ou “do”.

Liberando Dinheiro

entry/ bloquearValor(valorSacado)
do/ contarCedulas
exit/ debitarValor(valorSacado)

Exemplo: Sistema de Impressão – Diag. de Classes

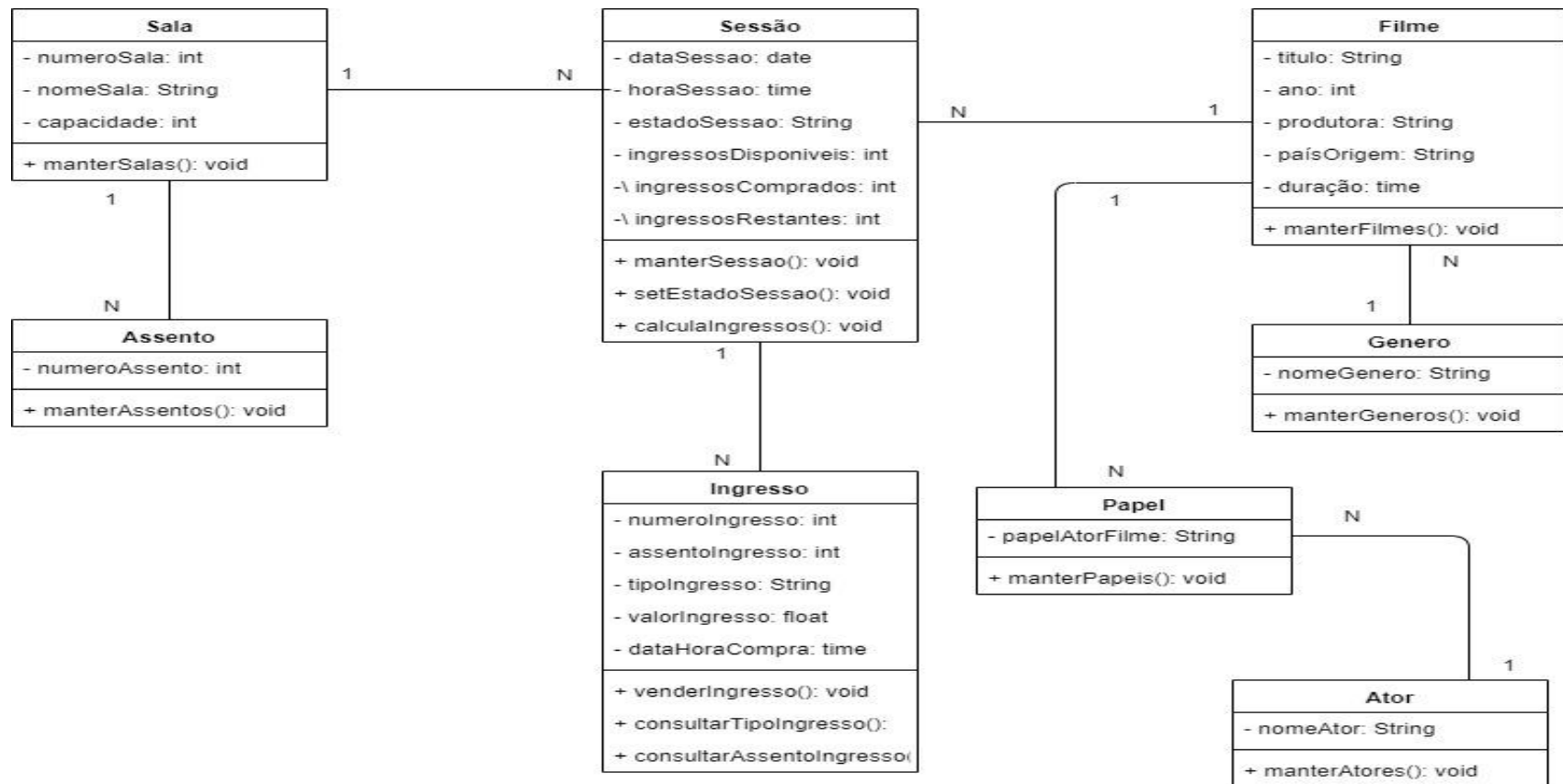


Exemplo: Sistema de Impressão – Diag. de Estados

Diagrama de Transição de Estados - Classe: PedidoImpressão

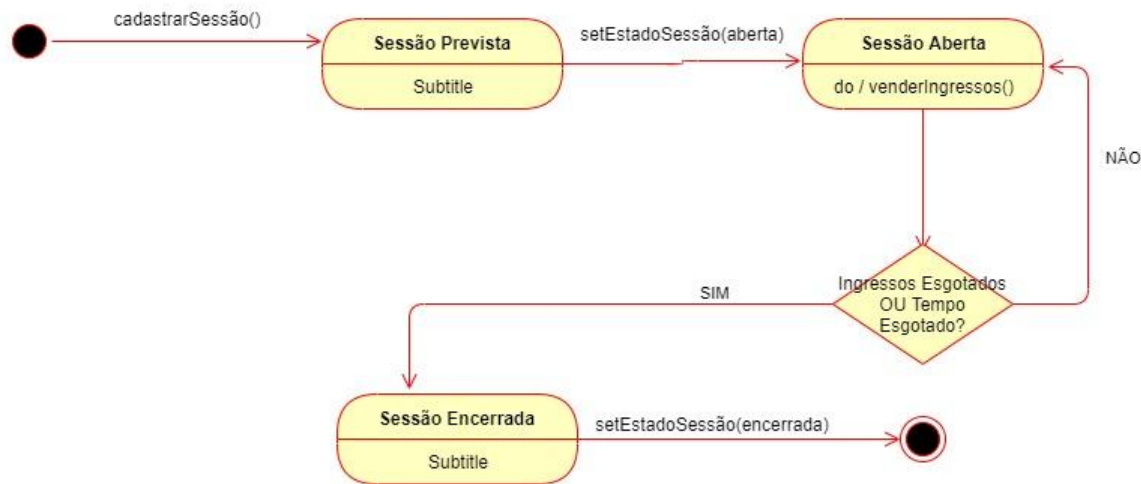


Exemplo: Sistema de Cinema – Diag. de Classes



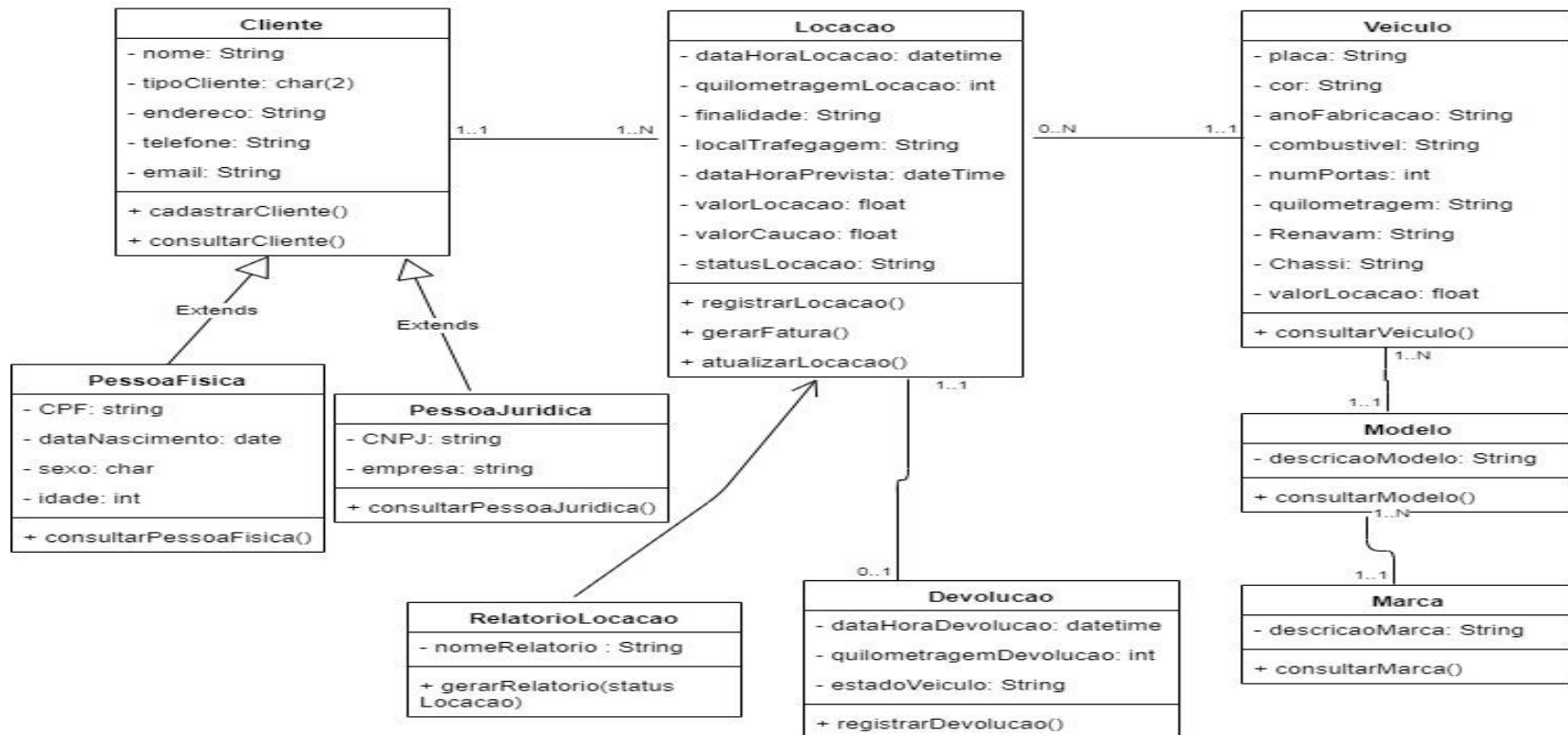
Exemplo: Sistema de Cinema – Diag. de Estados

Diagrama de Transição de Estados - Classe: Sessão



Exemplo: Locadora de Veículos – Diag. de Classes

Sistema de Locação de Veículos - Diagrama de Classes



Exemplo: Locadora de Veículos – Diag. de Classes

Sistema de Locação de Veículos - Diagrama de Estados

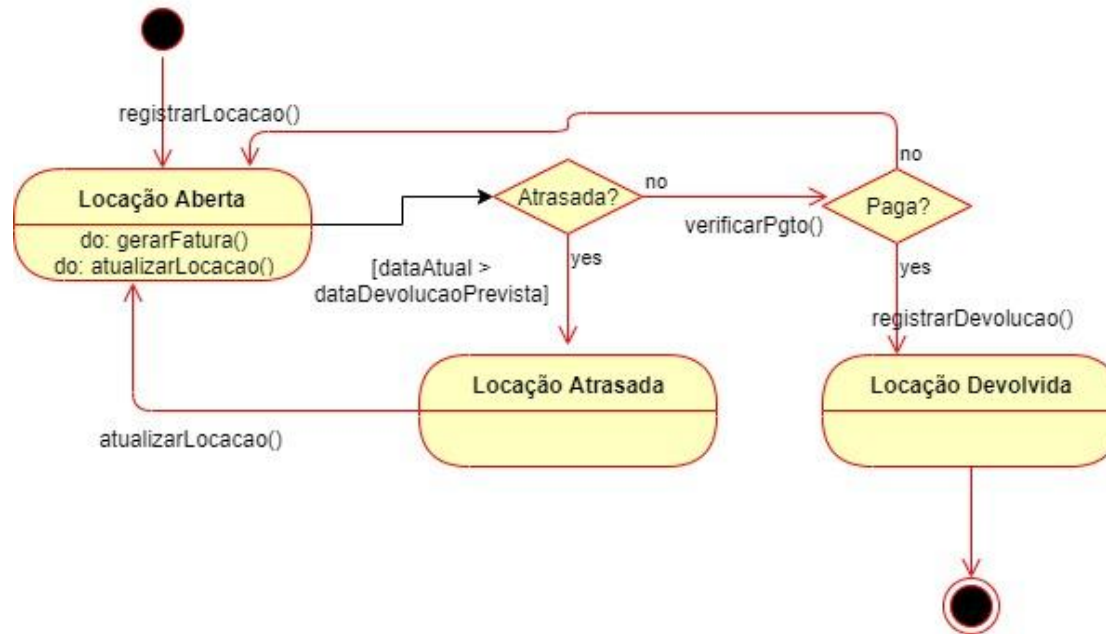


Diagrama de Sequência

O que é?

- Diagrama criado para modelagem da interação entre objetos
 - Detalha como objetos colaboram para implementar um cenário de caso de uso
 - Útil para ajudar na identificação dos métodos das classes

Diagramas de Interação

- Interação corresponde a um conjunto de **mensagens** trocadas entre **objetos**, com o objetivo de alcançar determinado **propósito**, respeitando-se o contexto do sistema
- Mostram como os objetos interagem para executar o **comportamento** de um **caso de uso** ou parte dele, para determinar interfaces e **responsabilidades** das classes e papéis dos objetos

Diagrama de Sequência

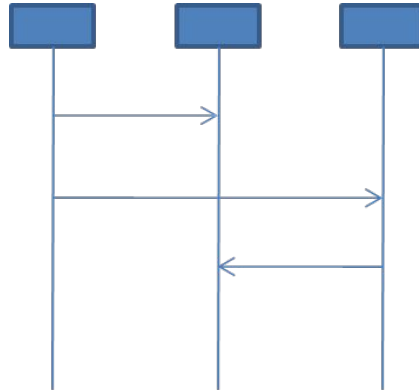
- Devem ser construídos após os diagramas de **Casos de Uso** e de **Classes**.
- Servem para documentar Casos de Uso e **validar** se **todas** as operações das Classes foram identificadas e declaradas;
- Em geral, elabora-se um Diagrama de Sequência para cada Caso de Uso, porém....

Diagrama de Sequência

- Não é necessário fazer diagramas de sequência para todos os casos de uso, **apenas** para os casos de uso **mais complexos**.
- Diagramas de Sequência **não** necessariamente precisam ser criados para **Casos de Uso típicos** de um CRUD, por serem casos mais simples.

A 1km de distância

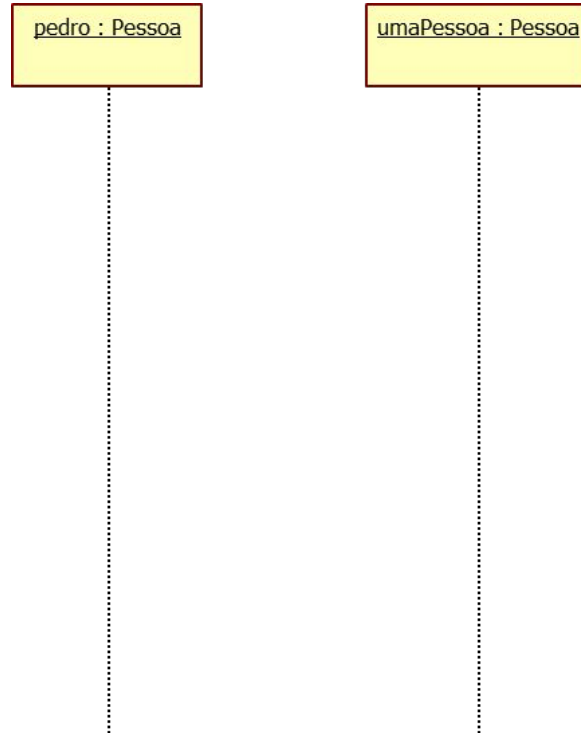
- Caixas representando objetos
- Linhas verticais representando a vida do objeto
- Linhas horizontais representando troca de mensagens (chamadas de métodos)



A 1 metro de distância dos objetos

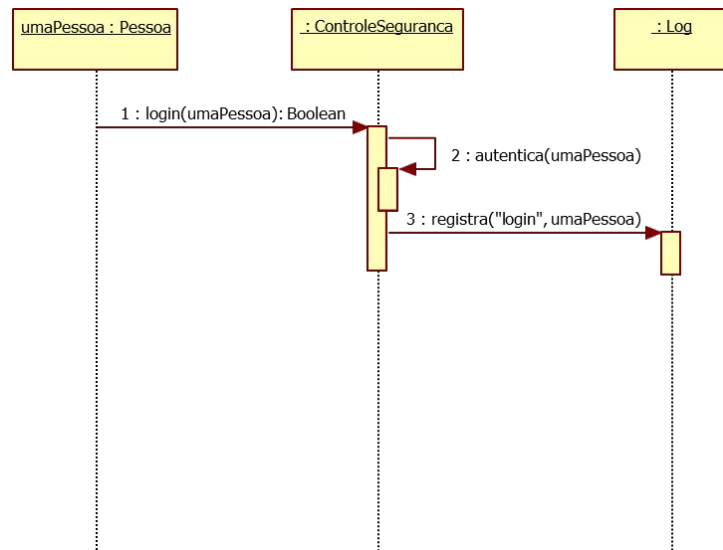
- Os objetos são instâncias de alguma classe definida no diagrama de classes
 - O nome de um objeto é da forma nome : classe
- Em situações onde um nome específico não pode ser identificado (ex.: pedro : Pessoa), utilize:
 - Um nome genérico (ex.: umaPessoa : Pessoa)
 - Um nome único (ex.: aPessoa : Pessoa)
 - Ou omita o nome (ex.: : Pessoa)
- Uma linha pontilhada sai do objeto representando o momento da sua criação em diante
 - Quanto mais para baixo, mais tempo passou

A 1 metro de distância dos objetos



A 1 metro de distância Das mensagens

- A interação entre objetos é representada por troca de mensagens (chamadas de métodos)
 - Para outros objetos
 - Para o mesmo objeto (auto- mensagem)
- Uma mensagem contém a assinatura do método que está sendo chamado
- Uma barra de ativação indica o escopo de execução do método



A 1 metro de distância Das mensagens

- Mensagem de criação
 - Aponta diretamente para o objeto e é marcada com <<create>>
- Mensagem de retorno
 - Opcional, e normalmente é omitida
 - Usa seta tracejada
- Marca de destruição
 - Indica o término da vida de um objeto com um "X"

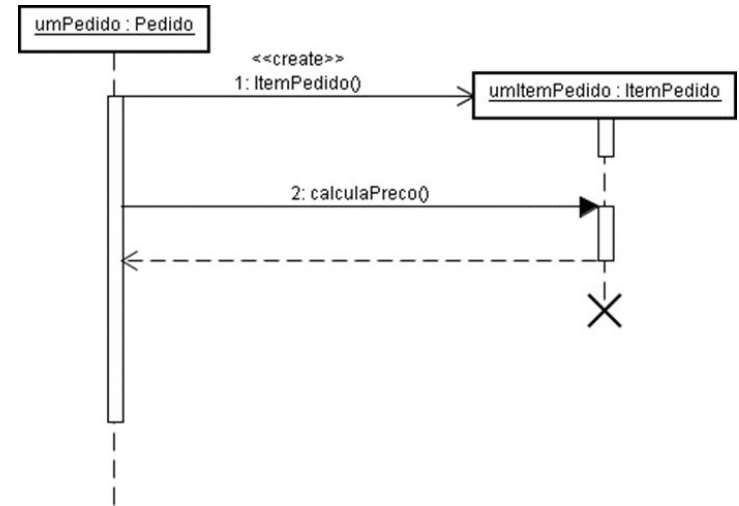


Diagrama de Sequência (Interação)

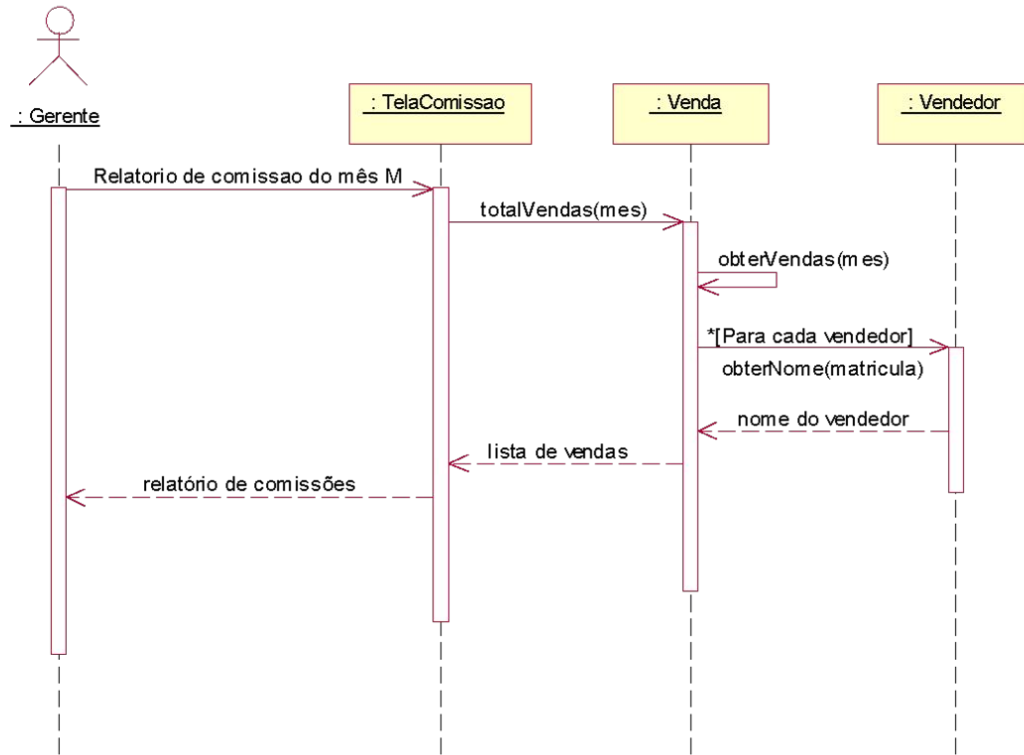
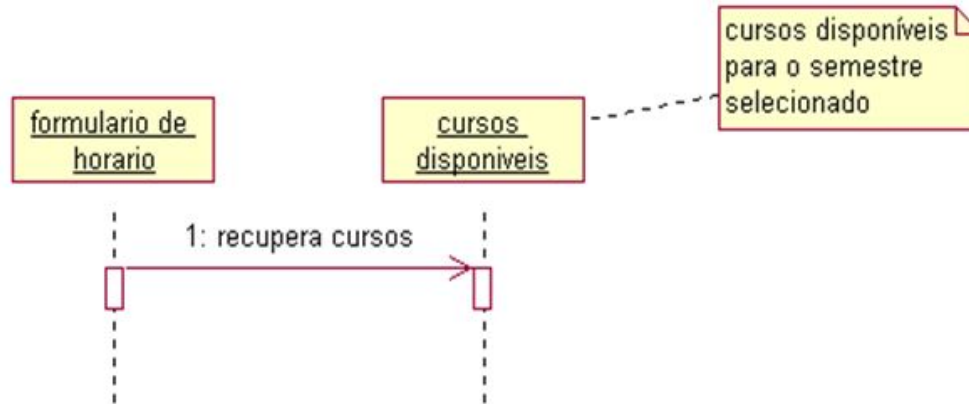


Diagrama de Sequência (Interação)

- **Notas:**

- Um diagrama pode incluir anotações com o objetivo de adicionar informação.



Mas como representar um algoritmo mais complexo?

- **Exemplo**

Para cada item de produto

 Se o valor do produto for maior que 10000 então

 Despacha com cuidado

 Caso contrário

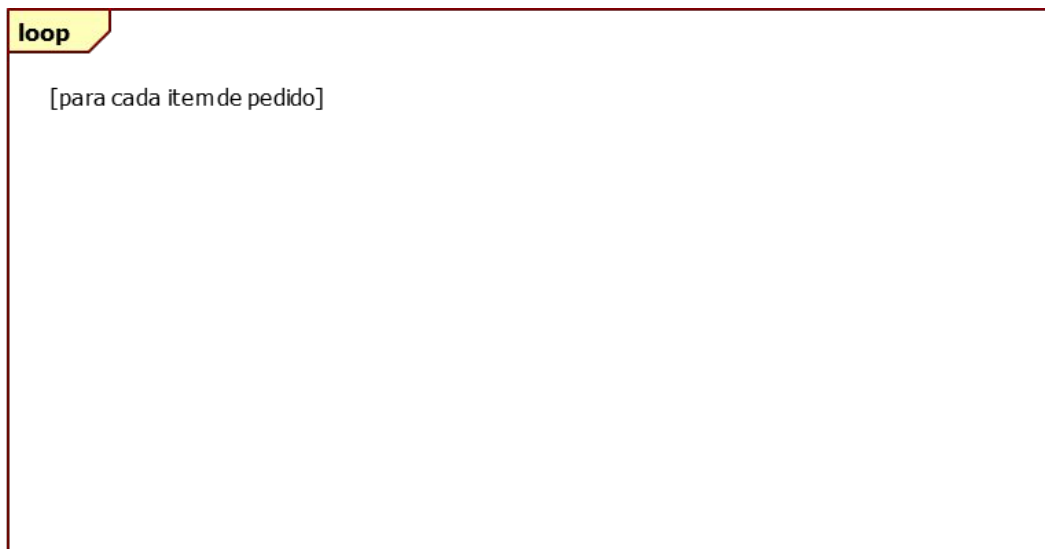
 Despacha normalmente

Se precisa de confirmação

 Envia confirmação

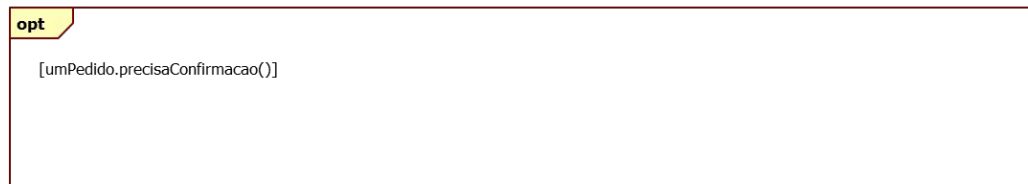
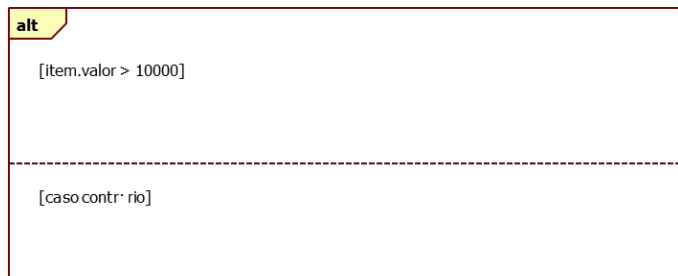
Repetições

- O diagrama de sequência permite que repetições sejam feitas durante o fluxo
- Para isso são utilizados quadros (*frames*) do tipo *loop*

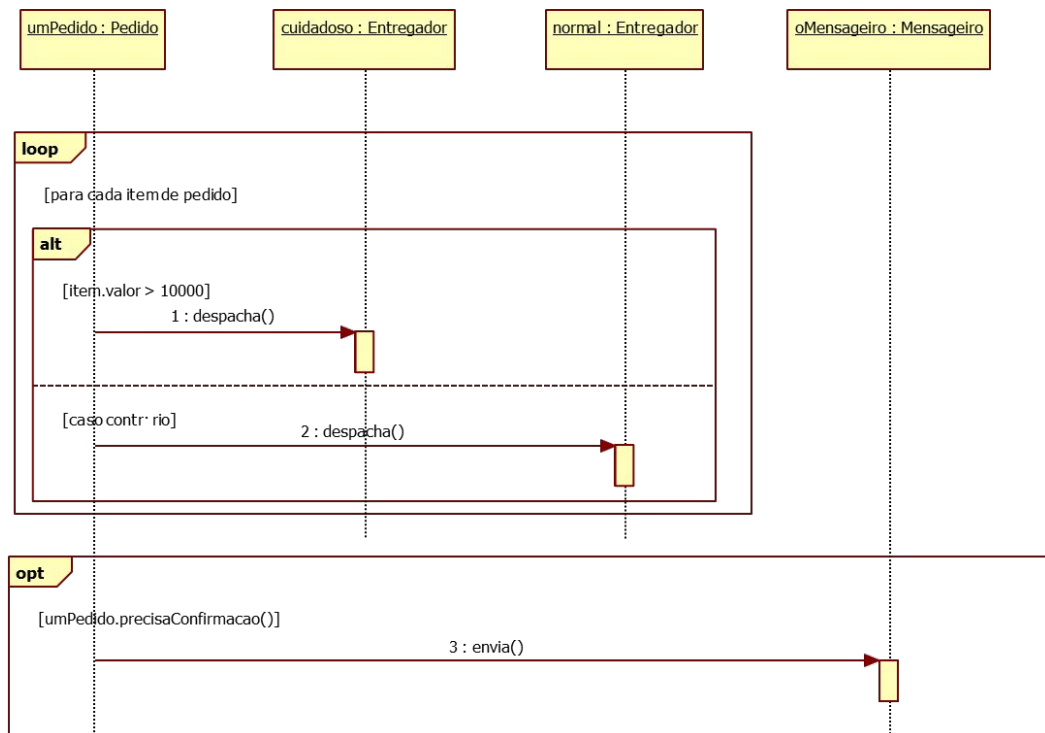


Decisões

- O diagrama de sequência permite que decisões sejam tomadas durante o fluxo
- Para isso são utilizados quadros (*frames*) do tipo *alt* ou *opt* com condições de guarda



Exemplo

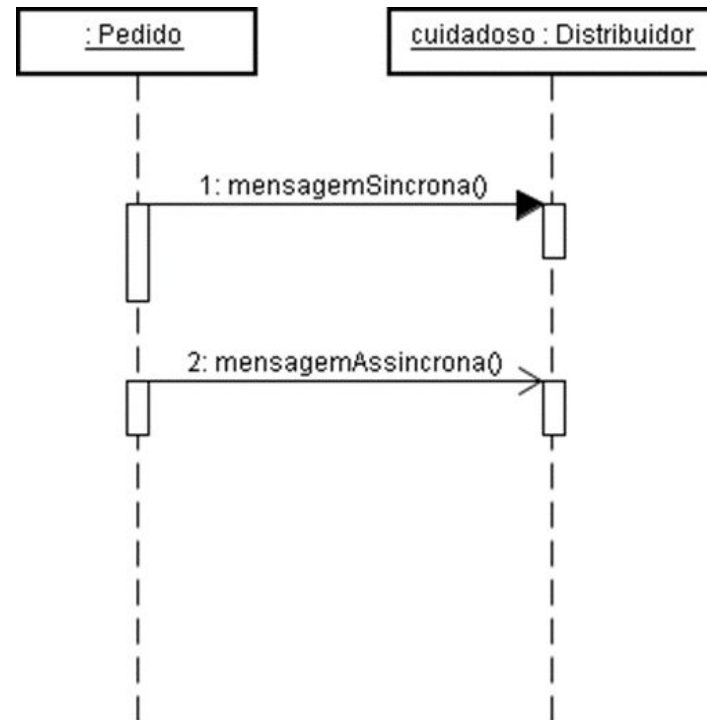


Outros quadros disponíveis

- Além dos quadros do tipo loop, opt e alt, existem outros tipos, entre eles:
 - par: Contém vários seguimentos e todos são executados em paralelo
 - region: Determina uma região crítica, que deve ter somente uma thread em execução em um dado momento

Chamada síncrona x assíncrona

- É possível utilizar dois tipos de chamada de métodos no diagrama de sequência:
 - Chamada síncrona (seta cheia): a execução fica bloqueada até o retorno do método
 - Chamada assíncrona (seta vazia): a execução continua em paralelo ao método que foi chamado (fork implícito)



Quando usar diagrama de sequência?

- Para representar a interação entre diferentes objetos visando atender a um caso de uso
 - Para ajudar a encontrar os métodos do diagrama de classes
 - Cuidado: não use diagrama de sequência...
 - Para métodos muito simples (ex.: get e set)
 - Para definição precisa de como será o código

Exercício

- Elabore um diagrama de sequência para o cenário de obter extrato de uma conta comum.
- Esse processo irá utilizar as classes Cliente, ContaComum, Historico, InterfaceCaixaEletronico

Modelagem de Software - Parte II

Prof. Pedro Henrique Dias Valle