



Projeto de Sistemas II

Faculdade Prof. Miguel Ângelo da Silva Santos

Material 3- Classes Modais e Alguns Padrões

Professor: Isac Mendes Lacerda, M.Sc., PMP, CSM
e-mail: isac.curso@gmail.com

Classes Modais

- ▣ ***Classes modais*** ou ***classes com estados*** são usadas para modelar variação de estado ao longo de sua existência;
- ▣ Destaque para três situações:
 - Transição estável
 - Transição monotônica
 - Transição não monotônica

Classes Modais: Transição Estável

- ▣ **Transição estável** não afeta a estrutura das instâncias, apenas os valores;
- ▣ Um exemplo seria o estado de um endereço:

Endereço
+cep
+rua
+número
+bairro
+complemento
+suspense : Booleano

Classes Modais: Transição Monotônica

- ▣ **Transição monotônica** muda de estado podendo ganhar novos atributos e associações;
- ▣ Um exemplo seria o de Pagamento:

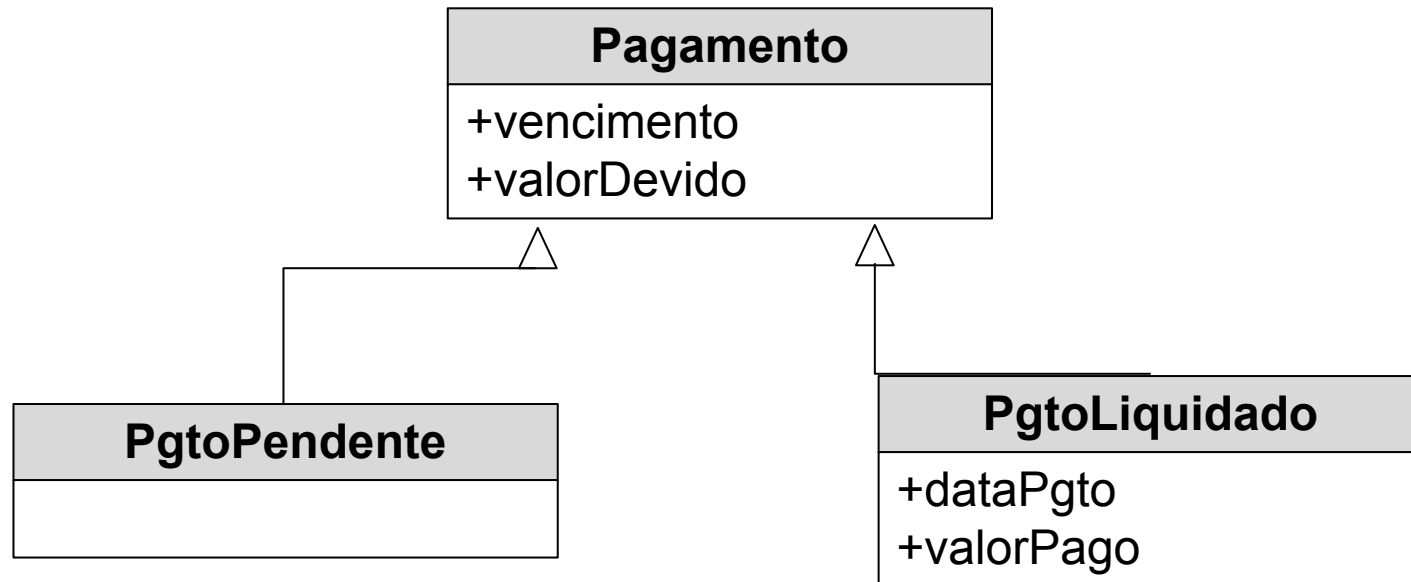
PgtoPendente
+vencimento +valorDevido



PgtoLiquidado
+vencimento +valorDevido +dataPgto +valorPago

Classes Modais: Transição Monotônica

- ❑ **Forma inconveniente** de resolver o problema seria **com herança** do tipo:



Classes Modais: Transição Monotônica

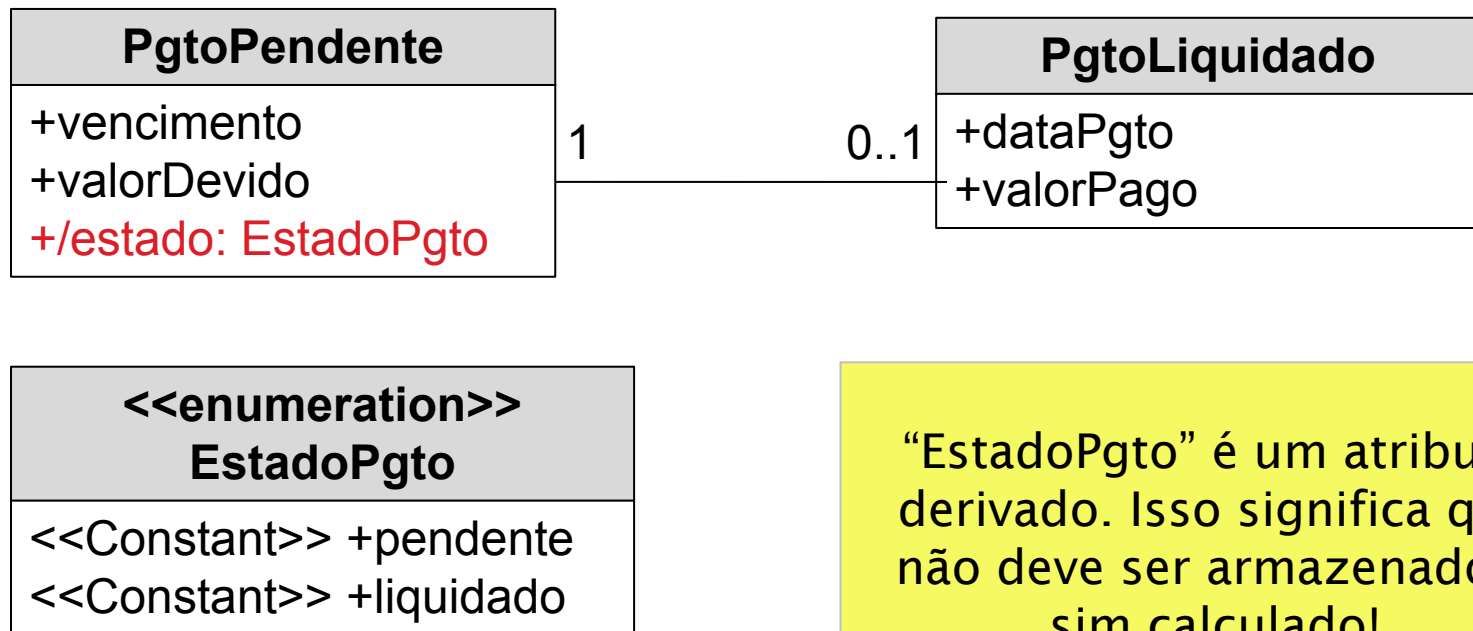
- Outra solução bastante usada, **mas não muito prática**, é ter todos os atributos em uma classe e controlar os nulos:

Pagamento
+vencimento +valorDevido +dataPgto +valorPago

Exige checagem frequente nas instâncias e regras complexas suscetíveis a erros

Classes Modais: Transição Monotônica

- Melhor seria desdobrar a classe em duas outras:



“EstadoPgto” é um atributo derivado. Isso significa que não deve ser armazenado e sim calculado!

Classes Modais: Transição Não Monotônica

- ▣ **Transição não monotônica** muda de estado podendo ganhar e perder atributos e associações;
- ▣ É raro querer perder alguma informação. No entanto, se ocorrer, estamos diante de uma transição não monotônica.

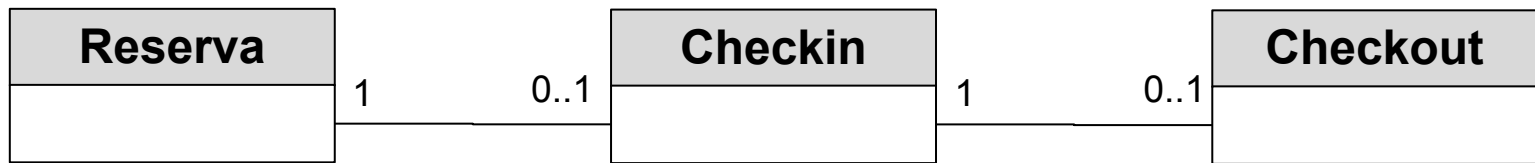
Classes Modais: Transição Não Monotônica

- Um exemplo (Sistema de reserva de hotel):
 1. Um hóspede pode fazer uma **reserva** indicando os dias previstos de chegada e saída, o tipo de quarto e o número de pessoas;
 2. Quando o hóspede faz o **checkin**, é registrado o dia de chegada. A data de saída prevista continua existindo, embora seu valor possa ser mudado no momento do *checkin*;
 3. Quando é feito o **checkout**, deixa de existir a data prevista de saída para existir a data de saída efetiva. Nesse momento, a conta deve ser paga.

Classes Modais: Transição Não Monotônica

□ Um exemplo (Sistema de reserva de hotel):

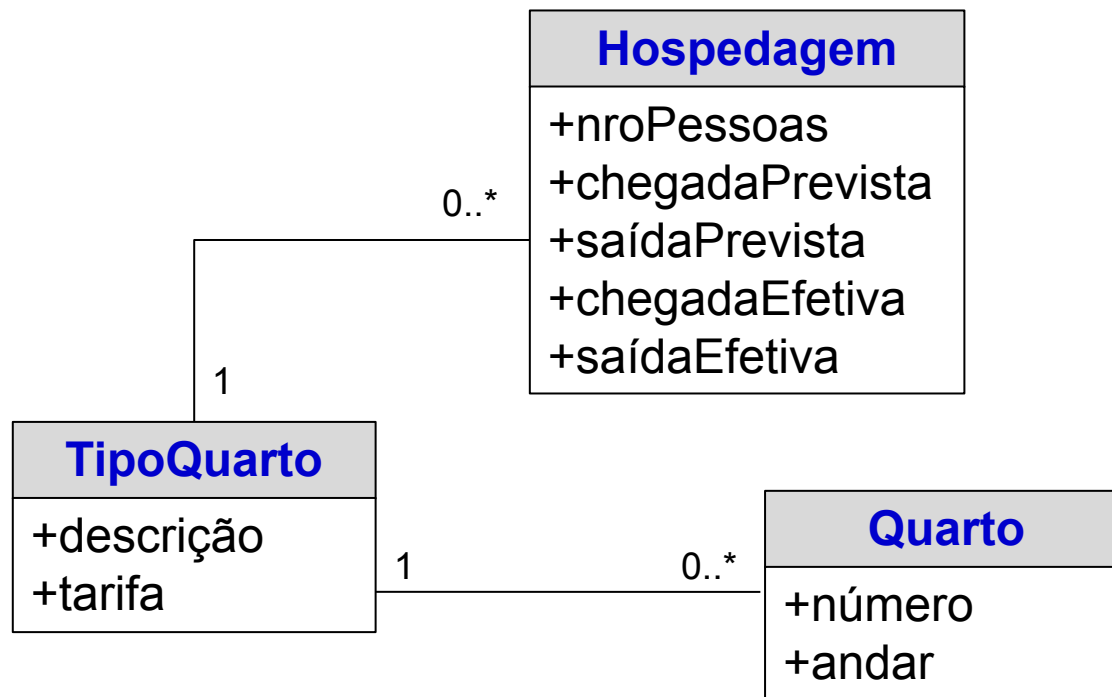
1. Considerando que os estados fossem cumulativos, poderíamos modelar como uma sequência de conceitos:



O problema aqui é que atributos como datas previstas são substituídas por datas reais, assim como tipo de quarto pode mudar no ato do checkin!

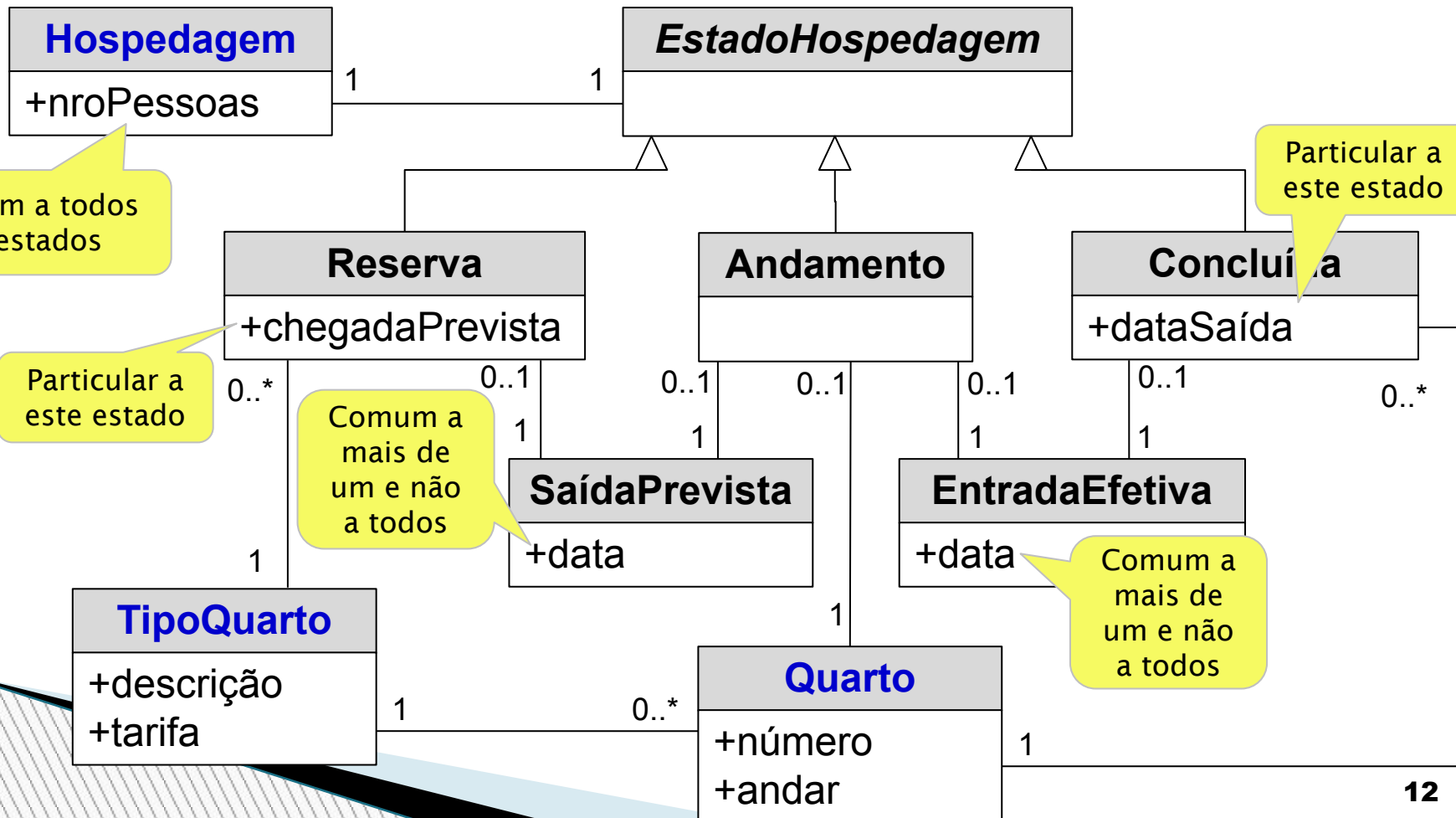
Classes Modais: Transição Não Monotônica

- Se os estados não fossem preocupação:



Classes Modais: Transição Não Monotônica

- Melhor seria tratar pelo padrão Estado (*state*):



Padrão: Estado (*State*)

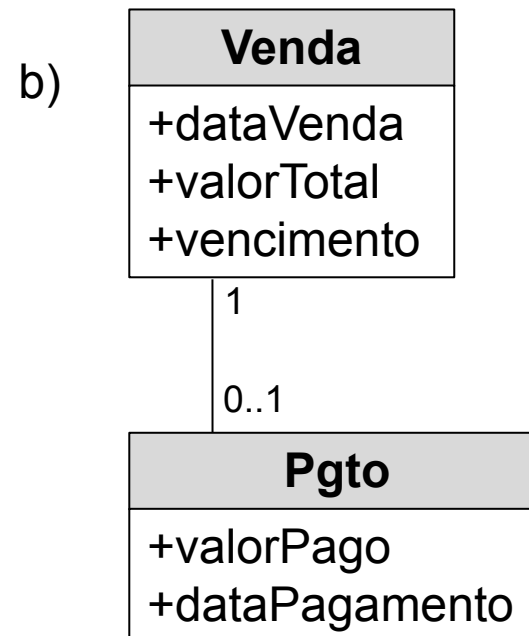
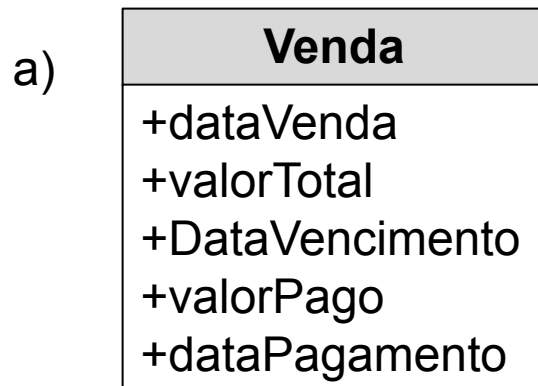
- Nesse padrão, o conceito (classe) deve ser separado dos estados (que devem virar classes também).
- Os atributos comuns aos estados devem ir para a classe original.
- Os estados devem ser classes especializadas de uma classe generalizada (abstrata).
- Os atributos comuns a mais de um estado, mas não a todos, tornam-se classes individuais também.
- Muito útil para tratamento de estado não monotônico.

Padrão: Alta Coesão

- O padrão de alta coesão propõe que atributos com pouca dependência entre si sejam separados em conceitos (classes) diferentes;
 - Exemplo de venda-pagamento;
 - Exemplo de Pessoa-RG-Endereço-Fone

Padrão: Coesão Alta

- O padrão de alta coesão propõe que atributos com pouca dependência entre si sejam separados em conceitos (classes) diferentes;
 - Exemplo de venda-pagamento;



Exercício 1



Modele com um diagrama de classes o minimundo seguinte, considerando o padrão **estado (state)**. Em um banco privado a abertura de contas pode incluir depósitos iniciais ou não (contas podem ter saldo maior que 0 ou exatamente 0, no ato da abertura). Cada uma das contas pode ter uma das três situações (conta positiva, conta negativa ou conta bloqueada). Nessas contas, podem ser realizadas operações de saque e depósito somente. Admita que as operações de saque e depósito mantêm lógicas diferentes entre os três estados possíveis para uma conta.

Exercício 2



Implemente em java as classes correspondentes ao minimundo seguinte, considerando o padrão **estado (state)**. Em um banco privado a abertura de contas pode incluir depósitos iniciais ou não (contas podem ter saldo maior que 0 ou exatamente 0, no ato da abertura). Cada uma das contas pode ter uma das três situações: positiva, negativa ou bloqueada. Nessas contas, podem ser realizadas operações de saque e depósito somente. Admita que as operações de saque e depósito mantêm lógicas diferentes entre os três estados possíveis para uma conta. As contas positivas podem realizar saques de até R\$-100,00 e devem ter o estado alterado (se for o caso). As contas negativas devem ser bloqueadas se houver uma tentativa de saque e o valor do saldo for menor R\$-100,00. As contas bloqueadas devem informar o estado de bloqueio, em qualquer tentativa de saque. Depósitos devem incrementar o saldo, alterar a situação da conta (se for o caso) e mostrar a situação depois da operação.

Exercício 3:



Modele classes de um sistema onde se possa aplicar transição de estado nos **modos estável**, **monotônica** e **não monotônica**. Considere ainda o uso dos padrões **alta coesão** e **estado (state)**. Para isso, utilize o seguinte minimundo:

Em uma rede de locação e venda de veículos, os clientes podem ter diversos endereços e telefones para contato em seus cadastros. Ao realizar uma reserva de locação o pagamento fica pendente. No início da locação o pagamento deve ser liquidado, incluindo o valor de caução (garantia por acidentes). A locação pode estar “Agendada”, “Em andamento” ou “Concluída”. Quando o carro for devolvido, sem acidentes, o valor de caução é estornado. Os veículos são administrados por diferentes estados, ou seja, “disponível para locação”, “disponível para venda”, “vendido” ou “manutenção”.

Exercício 4



Implemente o padrão estado de um sistema de hospedagem. Use para isso, a modelagem do slide 12 como partida. Você pode incluir novos atributos e métodos que considerar relevantes, mas não pode deixar de fora as classes e atributos indicados no slide 12. Com isso, simule um ciclo completo de variação de estados (todos os estados) para uma instância da classe de contexto.