

Projeto de Bloco

Engenharia Disciplinada de Softwares

Graduação em Engenharia de Software - 2020

Tarefas Solicitadas na Aula Passada - 13/08/2020

1. Revisar o Diagrama de Casos de Uso e o Diagrama de Classes, considerando o mapa mental de requisitos.
2. Iniciar a elaboração dos Protótipos de Tela para o sistema.

Etapa 3 Aula 1

Disciplined Agile Delivery - Inception

Modelo de Domínio

Como Identificar Classes?

- As classes de um sistema podem ser identificadas a partir de um diagrama de casos de uso e de suas descrições.
- Liste todas as **entidades** (tipos complexos) que forem encontradas nas descrições.
- Verifique se, entre estas entidades, não existe alguma relação, como por exemplo:
 - Elas são sinônimas?
 - Uma contém a outra?
 - Ambas têm muitos métodos e atributos em comum?

Classes de Um Sistema

- **Classes de Fronteira** - interfaces com o usuário e outros sistemas (Botões, *Checkboxes*, Telas).

Mais
sabe
do que
faz

- **Classes de Entidade** – estruturas de dados representando o domínio do problema (Cliente, Pedido, Item de Pedido, Produto).

Mais
faz do
que
sabe

- **Classes de Controle** – classes que representam processos e elementos dinâmicos (Datas e Tempo, Operações em Bancos de Dados, Gerenciador de Impressão, Leitura e Gravação de Arquivos).

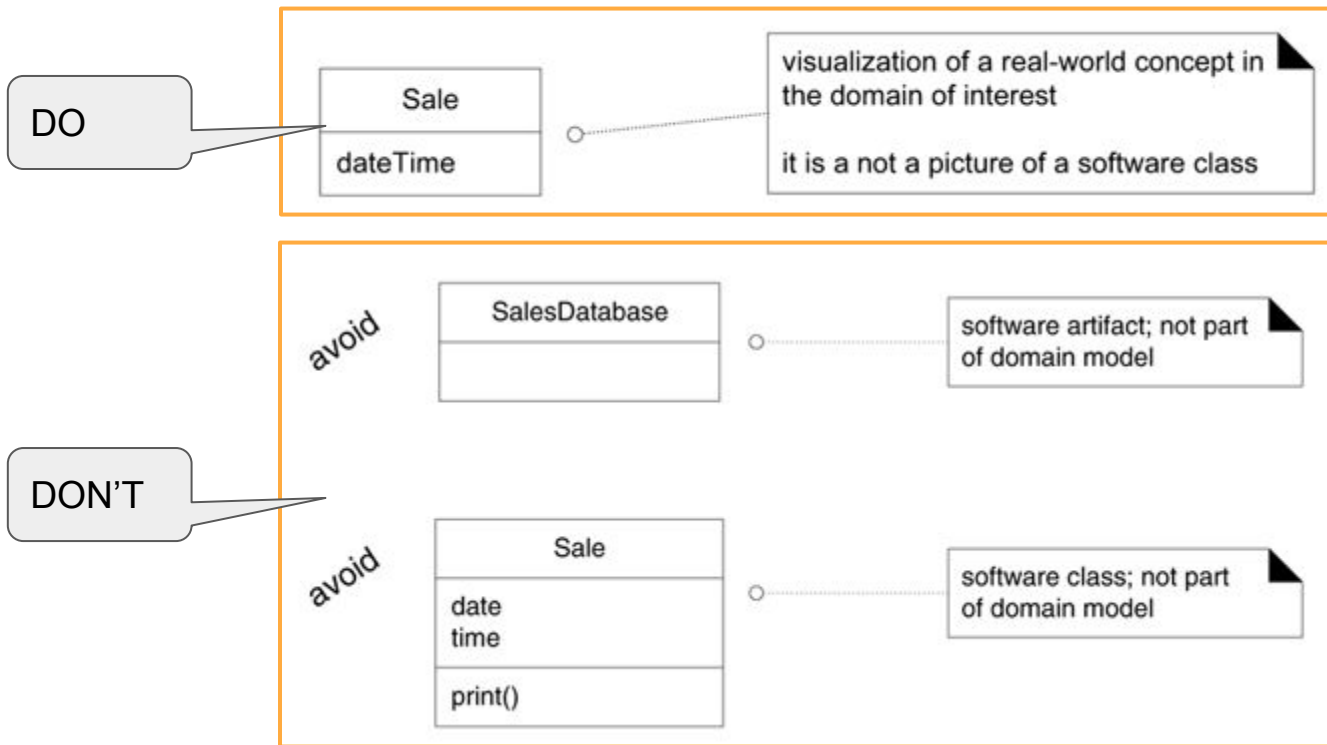
O que é um Modelo de Domínio?

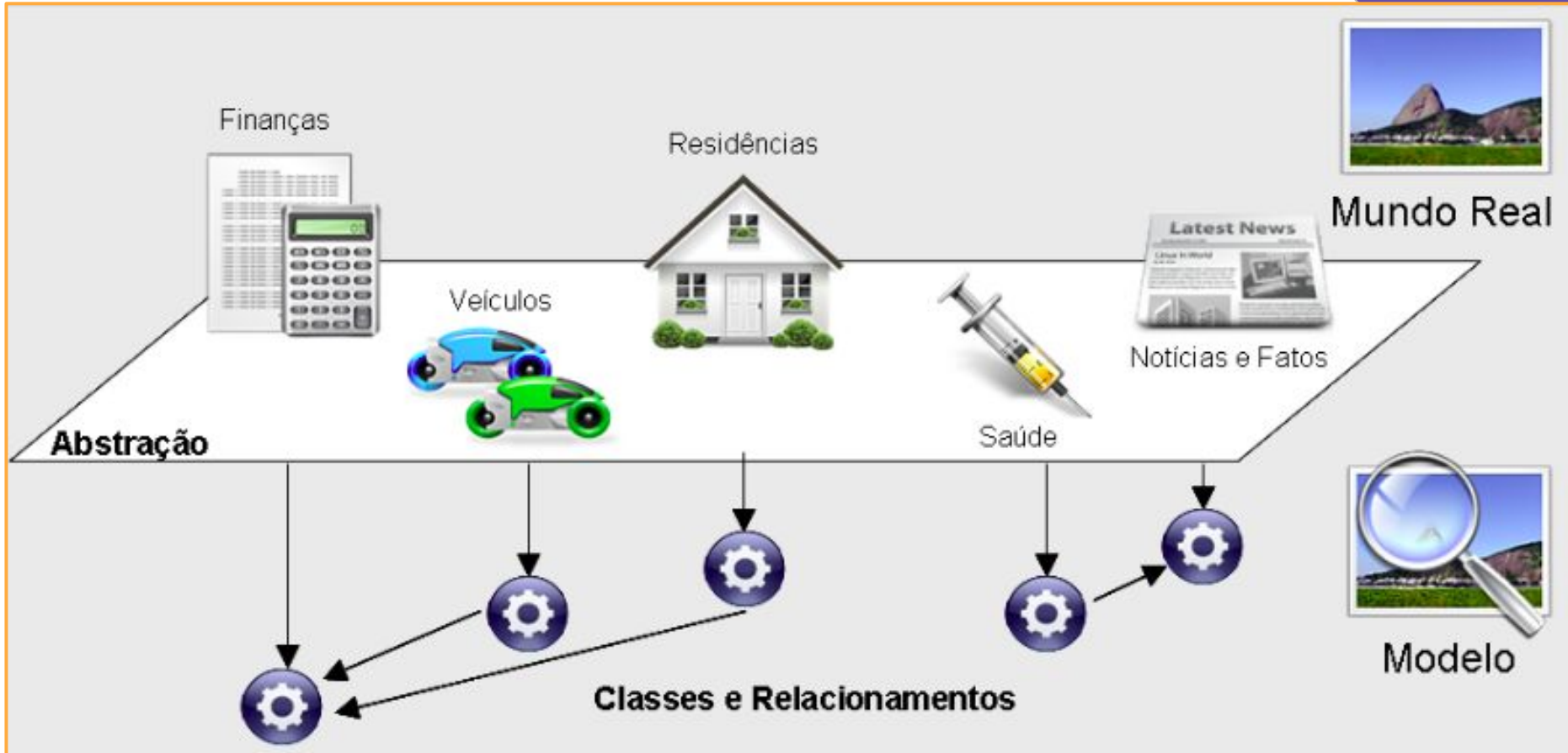
Um modelo de domínio é uma representação visual de **classes conceituais** ou objetos de situação real em um domínio [MO95,Fowler96].

Os modelos de domínio também são chamados de modelos conceituais, modelos de objeto de domínio e modelos de objeto de análise.

“Modelo de Domínio” significa uma representação de classes conceituais de situações reais, **não de objetos de software**.

O que é um Modelo de Domínio?





Abstração como um “plano”, separando o que é relevante para a solução do problema → Modelo de Domínio

Diretriz: como criar um modelo de domínio?

1. Encontre as classes conceituais.
 - Visto que um modelo de domínio mostra classes conceituais, uma questão central é: Como faço para encontrá-las?
 - i. Reutilize ou modifique os modelos existentes.
 - ii. **Use uma lista de categorias.**
 - iii. Identifique frases que contém substantivos relevantes.
2. Desenhe-os em um Diagrama de Classes UML.
3. Adicione associações e atributos.

Lista de Categorias

Categoria	Exemplos
Transações de negócio	Venda, Pagamentos, Reserva Aérea, Matrícula
Itens de linha de transação	Itens de venda, Itens da reserva
Produto ou serviço relacionado a uma transação ou item de linha de transação	Voo, Assento, Refeição
Onde a transação é registrada?	Nota-fiscal, Ticket de Embarque, Pauta de Chamada
Papéis de pessoas ou organizações relacionadas à transação; Atores dos casos de uso	Caixa, Cliente, Passageiro
Local de transação; Local de serviço	Loja, Aeroporto, Avião, Assento

Lista de Categorias

Categoria	Exemplos
Eventos notáveis, muitas vezes com uma hora ou lugar que precisamos lembrar	Venda, Pagamento, Voo
Objetos físicos	Item, Placa de registro, Peça, Avião, Viatura
Descrições de coisas	Descrição do Produto
Catálogos	Catálogo de Produtos, Tabela de Vendas
Registros de finanças, trabalho, contratos, questões jurídicas	Recibo, Livro-Razão, Balacete

Lista de Casos de Uso X Classes do VenturaHR

Manter = CRUD

Caso de Uso (Resumo)	Classes de Domínio
Usuário Mantém Conta	Usuário, Conta. (São sinônimos?)
Usuário Pesquisa Vagas	Usuário, Vaga
Empresa Publica Vaga	Usuário (Empresa), Vaga, Critérios da Vaga
Empresa Lista Vagas Publicadas	Usuário (Empresa), Vaga, Critérios da Vaga
Empresa Consulta Ranking de Vaga	Usuário (Empresa), Vaga, Usuário (Candidato), Resposta da Vaga
Candidato Responde à Vaga	Vaga, Critérios da Vaga, Resposta da Vaga, Respostas dos Critérios da Vaga
Candidato Lista Vagas Respondidas	Vaga, Critérios da Vaga, Resposta da Vaga, Respostas dos Critérios da Vaga

Modelo de Domínio do VenturaHR

Vamos
adicionar
atributos?

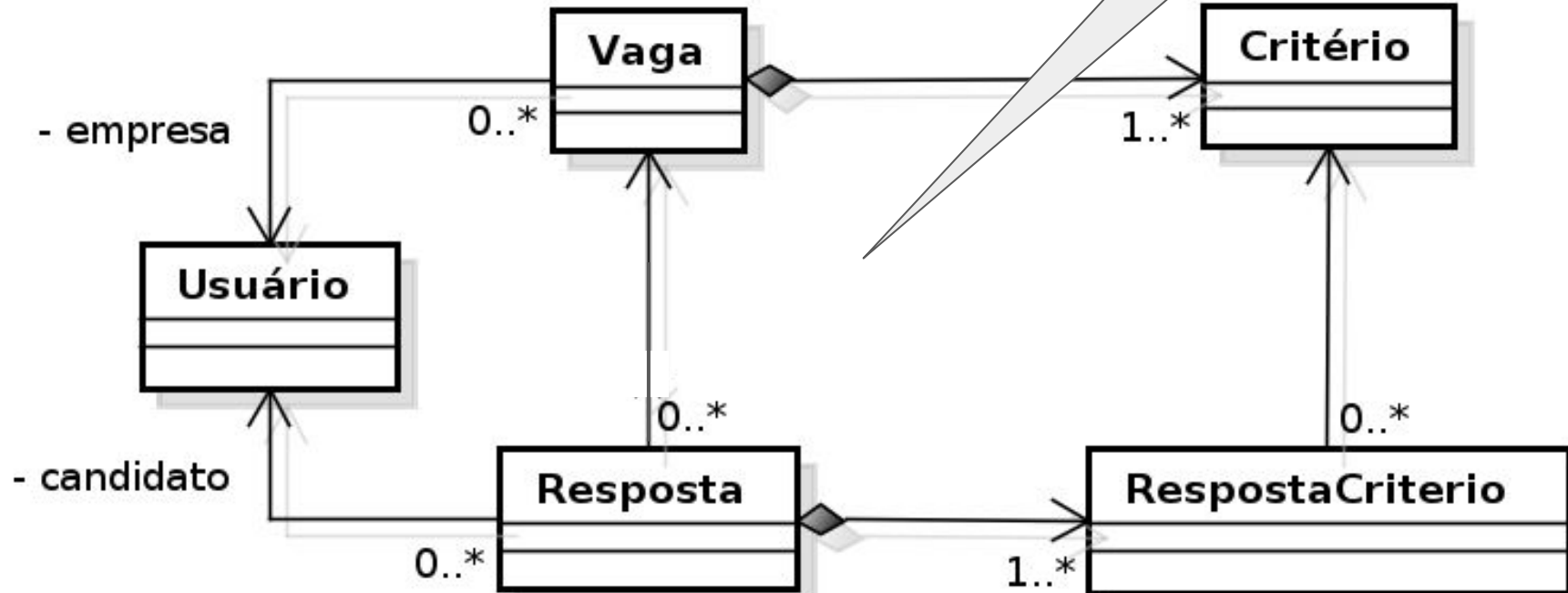


Diagrama de Estados

O que é o Diagrama de Estados ?

- Representa os estados de um objeto e a ordem pela qual ele passa por estes estados, desde sua criação até sua destruição.
- **Estuda o ciclo de vida de um objeto.**

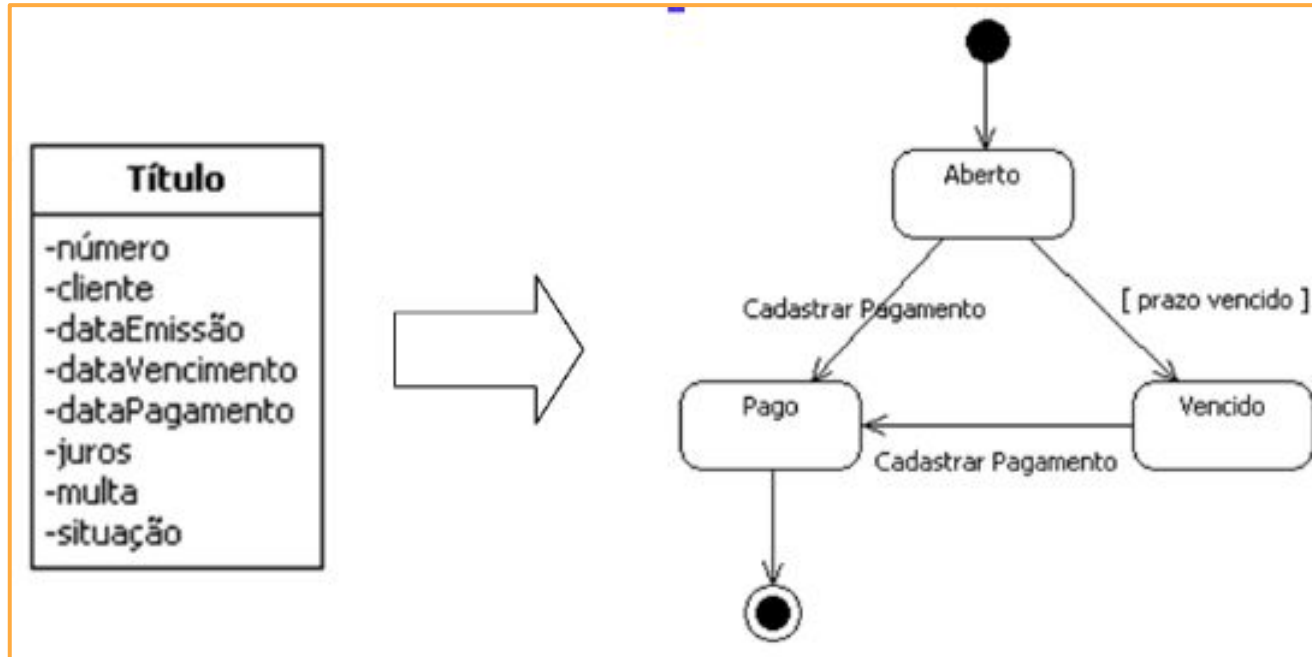
Componentes

- Estado.
- Transição.
 - Evento (caso de uso).
 - Ação (função elementar ou método de classe).
 - Condição de Guarda (aparece entre colchetes).
- Estados inicial e final.



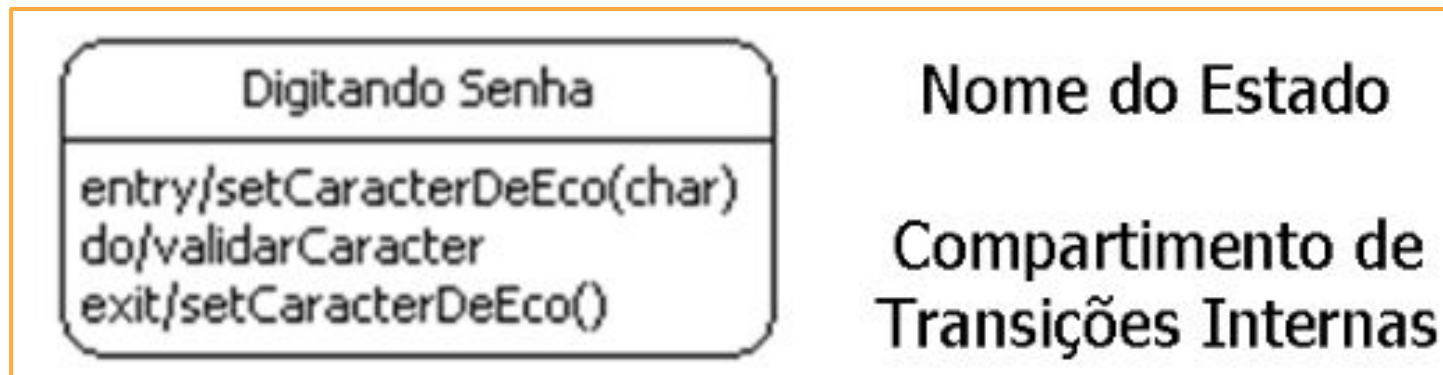
Componentes

- Estados são as situações de um objeto quando ele satisfaz alguma condição, aguarda um evento ou realiza alguma atividade.



Componentes

- Outra forma de desenhar um estado é separá-lo em dois compartimentos: um para o nome e outro para transições internas ao estado.



Essa forma só deve ser usada quando se deseja detalhar o estado

Componentes

- Transição:
 - evento / condição de guarda / ação (parâmetros)
 - Evento – é a especificação de uma ocorrência, que irá ocasionar a alteração de estado do objeto.
 - Ação – atividade elementar que não pode ser interrompida. É executada no início da transição.
 - Condição de Guarda – a transição será ativada apenas se a condição for verdadeira. Aparece entre colchetes '[']' na transição.

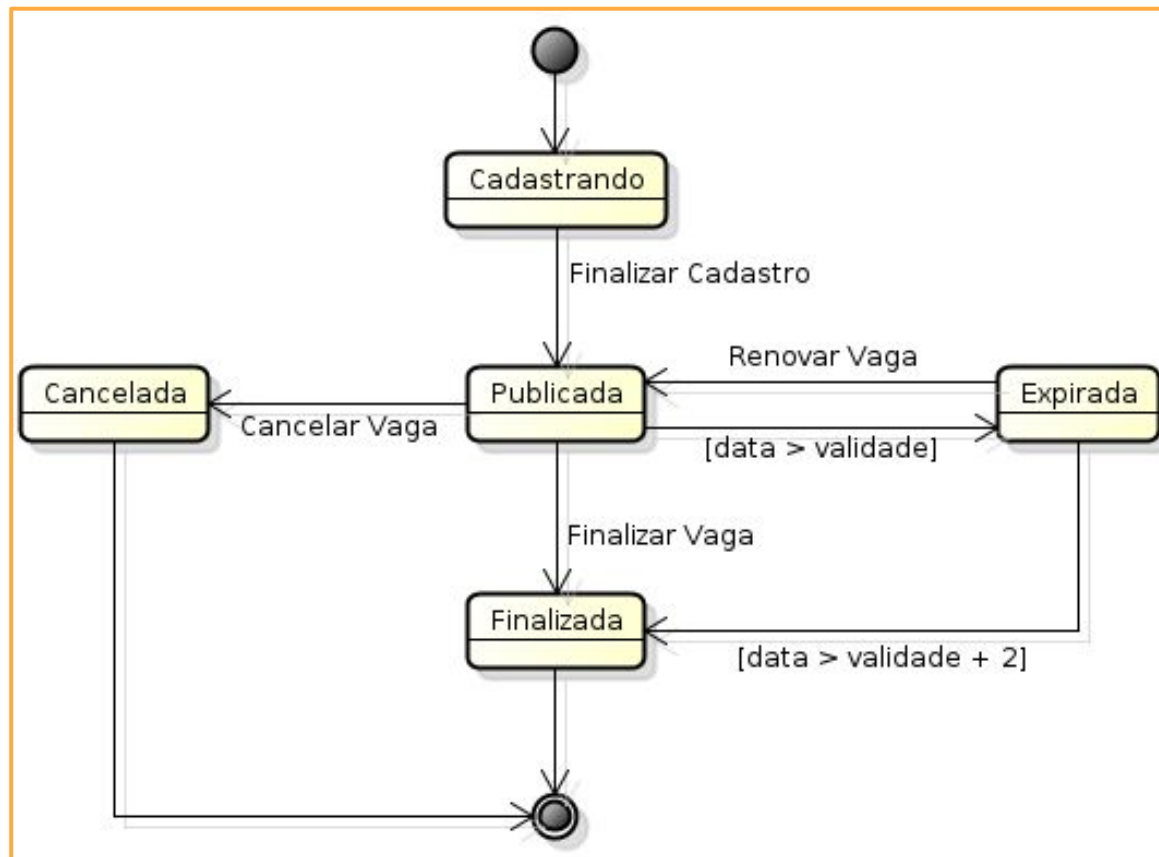
Componentes

- Estados:
 - Pronto, Em Manutenção, Checando, Finalizando, Efetuando Transação, Em Consulta, Em Internação.
- Eventos:
 - Cadastrar Empregado, Demitir Funcionário.
- Ações:
 - identificarCartao(), validarSenha().
- Condições de Guarda:
 - Senha inválida, falha detectada, Data expirada.

Quando Usar o Diagrama de Estados?

- Nem todas as classes podem ser representadas por um diagrama de estados, simplesmente por não possuírem estados a serem analisados.
- Se a classe tem um atributo *status* (ou situação), isso é um indicativo de que o objeto terá comportamentos diferentes – de acordo com os valores informados neste atributo.

Estados da Vaga



Arquitetura Lógica e Diagramas de Pacote UML

Modelo de Referência de Arquitetura

Presentation Tier

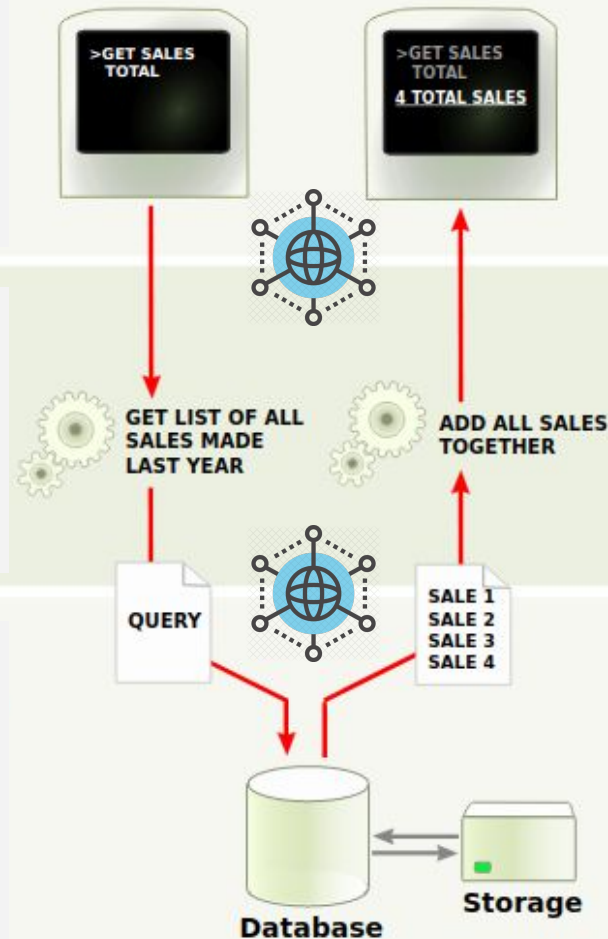
Nível mais alto da aplicação; Interface com o usuário. Sua principal função é traduzir as tarefas e resultados para algo que o usuário possa entender.

Logic Tier

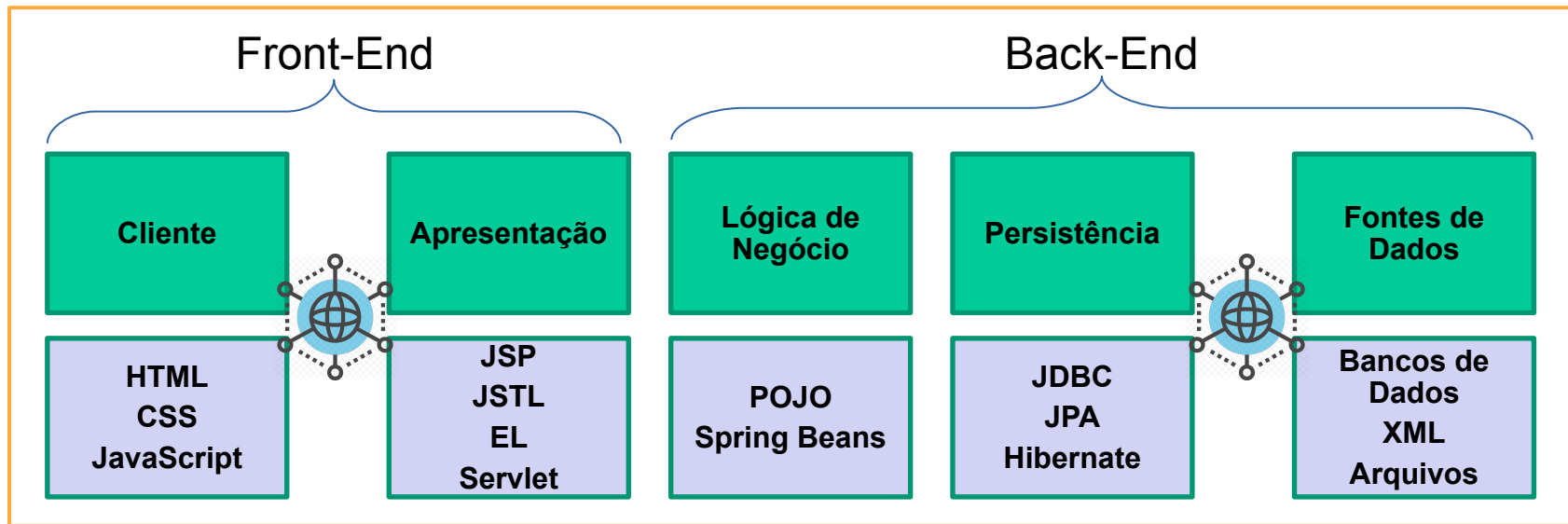
Essa camada coordena a aplicação, processa os comandos, toma as decisões lógicas, avalia e executa as ações. Além disso, processa e move os dados entre as outras camadas.

Data Tier

Aqui a informação é armazenada e recuperada do banco de dados, sistema de arquivos ou web services. A informação é então passada para a camada “lógica de negócio” e para o usuário.

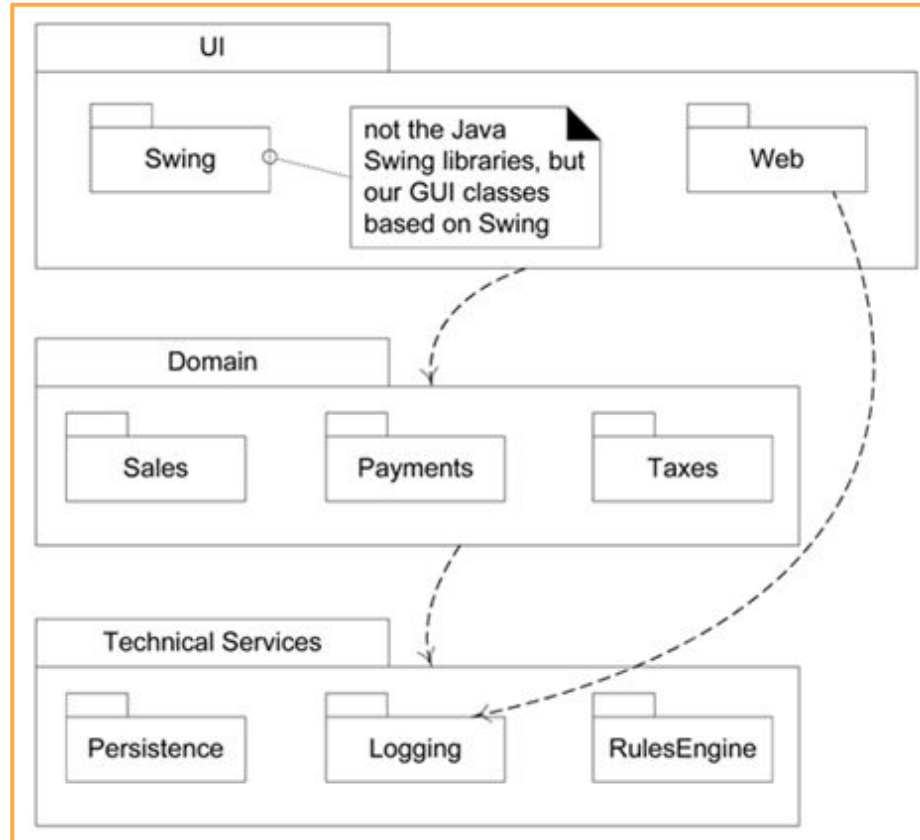


Modelo de Referência de Arquitetura



POJO = Plain Old Java Object

Exemplo de Arquitetura Lógica



O que é a arquitetura lógica? E camadas?

- A arquitetura lógica é a organização em grande escala das classes de software em pacotes (ou namespaces), subsistemas e camadas → **Modularização**.
- É chamada de arquitetura lógica porque não há decisão sobre como esses elementos são implantados em diferentes processos do sistema operacional ou em computadores físicos em uma rede (essas últimas decisões fazem parte da arquitetura de implantação).

O que é a arquitetura lógica? E camadas?

- Uma camada é um agrupamento muito granulado de classes, pacotes ou subsistemas que tem **responsabilidade coesa** por um aspecto principal do sistema.
- As camadas são organizadas de forma que as camadas "superiores" (como a camada de IU) solicitem serviços de camadas "inferiores", mas normalmente não o contrário.

O que é a arquitetura lógica? E camadas?

- Interface do Usuário / Experiência do Usuário.
- Lógica de aplicação (lógica de negócio) e objetos de domínio:
 - Objetos de software que representam conceitos de domínio (por exemplo, venda) que atendem aos requisitos do aplicativo, como o cálculo de um total de vendas.
- Serviços técnicos:
 - Objetos de uso geral e subsistemas que fornecem serviços técnicos de suporte, como interface com um banco de dados ou registro de erros.
 - Esses serviços geralmente são independentes do aplicativo e reutilizáveis em vários sistemas.

Aplicando UML: Diagramas de Pacote

Os diagramas de pacotes da UML são freqüentemente usados para ilustrar a arquitetura lógica de um sistema - as camadas, subsistemas, etc.

Uma camada pode ser modelada como um pacote UML; por exemplo, a camada de UI modelada como um pacote denominado UI.

Aplicando UML: Diagramas de Pacote

O diagrama de pacotes fornece uma maneira de agrupar elementos.

Um pacote UML pode agrupar qualquer coisa: classes, outros pacotes, casos de uso e assim por diante.

Pacotes aninhados são muito comuns.

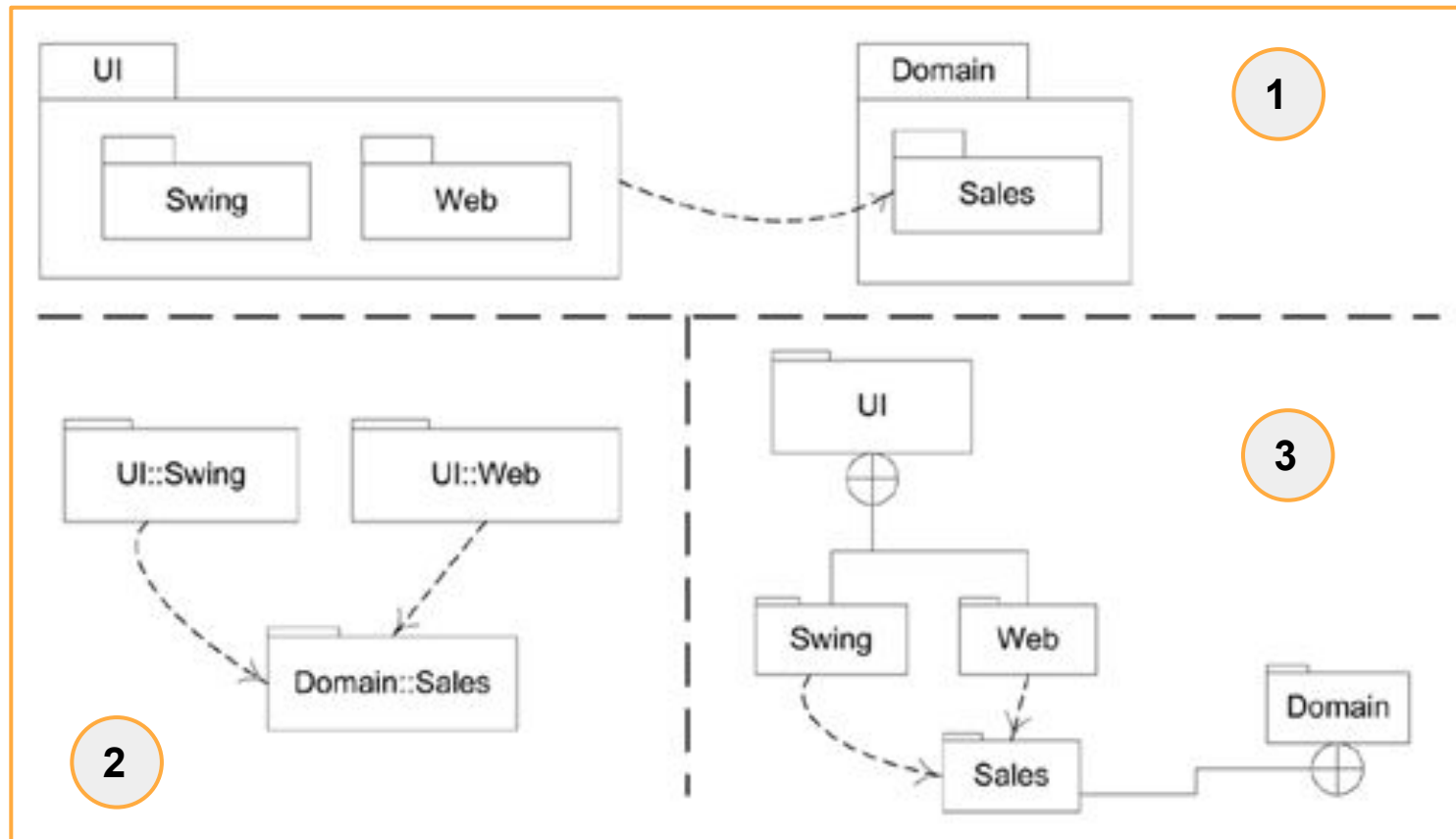
Um pacote UML é um conceito mais geral do que simplesmente um pacote Java ou namespace .NET, embora um pacote UML possa representá-los - e mais.

Aplicando UML : Diagramas de Pacote

É comum querer mostrar dependência (um acoplamento) entre pacotes para que os desenvolvedores possam ver o acoplamento em grande escala no sistema.

A linha de dependência UML é usada para isso: uma linha de seta tracejada com a seta apontando para o pacote do qual se depende.

Aplicando UML : Diagramas de Pacote



Pacotes em Java

```
// --- UI Layer

com.mycompany.nextgen.ui.swing
com.mycompany.nextgen.ui.web


// --- Camada DOMAIN

// pacotes específicos para o projeto NextGen
com.mycompany.nextgen.domain.sales
com.mycompany.nextgen.domain.payments


// --- Camada de SERVIÇOS TÉCNICOS

// nossa camada interna de acesso de persistência (banco de dados)
com.mycompany.service.persistence


// terceiro
org.apache.log4j
org.apache.soap.rpc


// --- Camada FOUNDATION

// pacotes
básicos que nossa equipe cria com.mycompany.util
```

Tarefas para Próxima Aula - 27/08/2020

1. Revisar os diagramas de Casos de Uso e Classes (modelo de domínio), incluindo os atributos no modelo de domínio.
2. Elaborar um esboço do Diagrama de Pacotes do VenturaHR.
3. Iniciar a elaboração do TP3 a ser entregue no dia 31/08 (o enunciado será publicado amanhã 21/08).