Aplicação de Linguagem de Programação Orientada a Objetos

LAB

Prof. Gustavo Molina

Agenda

- **Design Patterns**
- **Exemplo**
- **Esquema MVC / DAO / DTO**

Design Patterns

Design Patterns

Design Patterns – Padrões de Projetos:

No desenvolvimento de um sistema, espera-se que alguns requisitos sejam garantidos, como por exemplo: desempenho (eficiência e eficácia - que suporte a necessidade do Cliente e que funcione corretamente), compreensão, facilidade na manutenção na utilização e na reutilização.

Os Design Patterns surgiram da necessidade de criação, acerto ou manutenção de sistemas com agilidade, rapidez e preços competitivos.

Assim, à partir das *melhores práticas*, reuniram-se as melhores e mais eficazes soluções, transformando-as em padrões, e algumas até em Frameworks (pacotes prontos de códigos que fornecem alguma solução específica).

MVC

A Arquitetura MVC (Model-View-Controller) :

Essa arquitetura foi desenvolvida na década de 70 para ser utilizada em projetos de interface visual em Smalltalk (uma linguagem de programação que juntamente com o C++ ganhou grande reconhecimento na época), e após todos esses anos, ainda é importante nas mais variadas aplicações, principalmente em aplicações web.

MVC

A Arquitetura MVC (Model-View-Controller) :

Nesta arquitetura, o sistema é dividido em 3 camadas (pacotes) a fim de separar as Classes por "responsabilidades", de forma que:

- as Classes responsáveis pelas "telas" (interface com o Usuário) ficam na camada de Visualização.

- as Classes responsáveis pelo processo de Persistência (acesso a Banco de Dados) devem ficar na camada de Modelagem.

- na camada de Controle ficam as Classes responsáveis pelo controle das informações que tramitam entre as camadas de visualização e de modelagem. Nesta camada são realizadas as operações de validação, controle e encaminhamento daquelas informações.

MVC

Informações

View

Control

Model

Camada que contem:

- Classes que "constroem" telas.
- arquivos html (sistemas Web)
- imagens
- Etc.

Camada responsável pela intermediação das informações entre as camadas View e Model:

- validações
- cálculos
- Preenchimento de dados nas telas.
- Etc.

Camada que contem as Classes responsáveis pela Persistência:

- Classes que geram as conexões.
- Classes que geram e executam as queries.
- Etc.

DTO

O Pattern DTO (Data Transfer Object) :

Neste Pattern um objeto é utilizado para transferir dados de um local a outro na aplicação, comumente associado à transferência de dados entre uma camada de visão (view layer) e outra de persistência dos dados (model layer).

Assim, um Objeto encapsulado é preenchido com dados em seus atributos, e esse Objeto é transportado por entre as camadas de um MVC. Esse Objeto é considerado um JavaBean, ou seja, uma Classe Encapsulada simples, e representando uma entidade do Banco de Dados (uma Tabela, uma View, etc.).

Connection Factory

A Fábrica de Conexões :

A princípio uma Fábrica de Conexões com o Banco de Dados pretende administrar conexões que são solicitadas e geradas à medida que o sistema necessita acessar algum dado do BD.

A ideia é que ao solicitar uma Conexão ao Connection Factory, o mesmo fornecerá uma conexão (um objeto Connection), gerenciando a quantidade de conexões, e controlando aquelas que não estão mais sendo utilizadas.

DAO

O Pattern DAO (Data Access Object) :

Em geral, todo sistema necessita de persistência de informações.

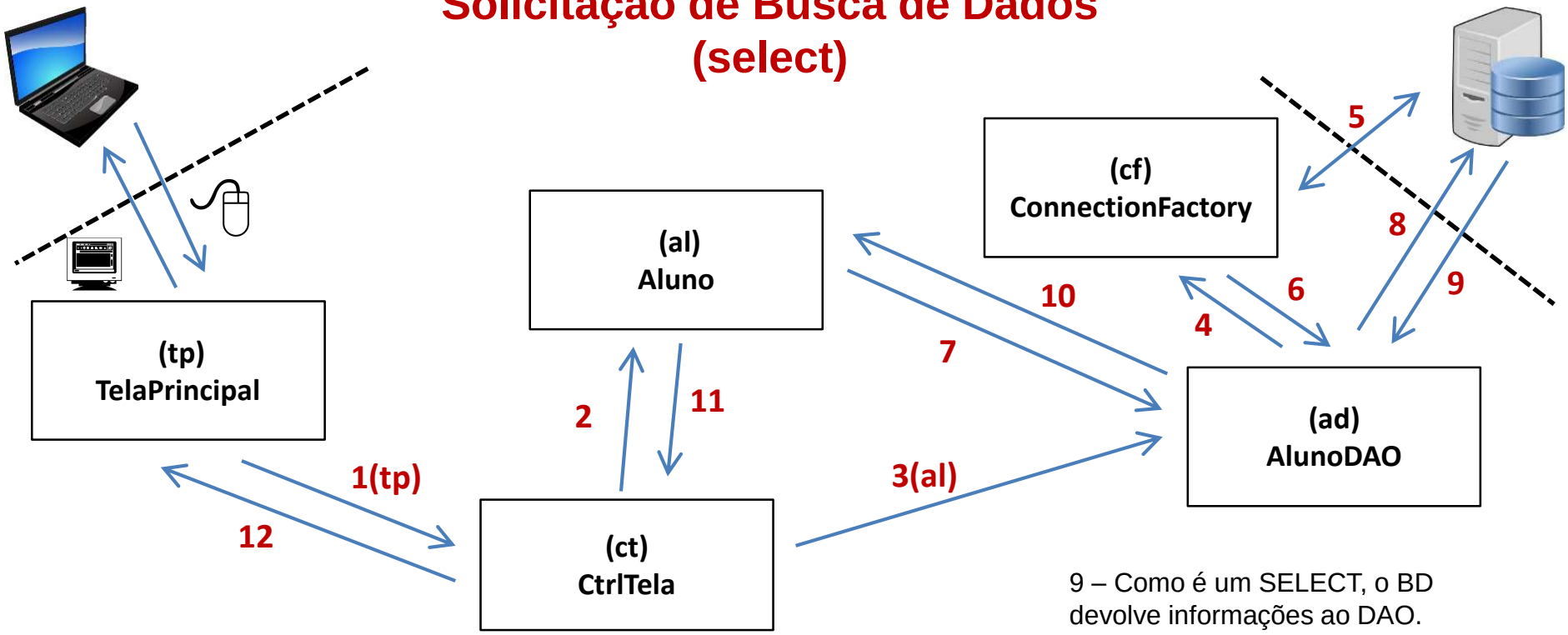
Seguindo o Pattern MVC cria-se a camada de Modelagem que será a responsável pelo transporte dos dados direto *do, e para o*, Banco de Dados.

Assim, as Classes do "model" que ficarão responsáveis por preparar diretamente esta comunicação direta com o BD, são as Classes DAO. Elas ficam responsáveis por criar corretamente as queries que o BD receberá para manipular os Dados.

Exemplo

Exemplo de Projeto

Solicitação de Busca de Dados (select)



INI – Usuário solicita dados.
1 – Tela chama Controle (enviando a própria tela como parâmetro).
2 – Controle cria um JavaBean (Aluno) e preenche o seu "Id" (para busca).
3 – Controle cria DAO (enviando Aluno – JavaBean).

4 – DAO solicita Conexão com BD.
5 – ConnectionFactory cria conexão com BD e...
6 – ...devolve esta conexão ao DAO.
7 – Com os dados do JavaBean, o DAO cria a query.
8 – DAO executa a query no BD.

9 – Como é um SELECT, o BD devolve informações ao DAO.
10 – O DAO preenche (popula) o JavaBean recebido com os dados lidos.
11 – O Controle pega os dados preenchido no JavaBean e...
12 – ...atualiza os campos da TelaPrincipal, preenchendo-os...
FIN - ...permitindo que o Usuário os visualize.

Esquema MVC / DAO / DTO

MVC / DAO / DTO

Solicitação de Busca de Dados (select)

VIEW

CONTROL

MODEL

1 - Gera ao controle uma solicitação de busca de informação.

2 - Valida os dados, e se estiver OK procede solicitação de busca.

3 - Gera um **JavaBean** com os dados necessários à solicitação.

4 - Envia o **JavaBean** à solicitação equivalente do **DAO** (para que seja encaminhada ao BD).

5 – Com o **JavaBean**, o **DAO** construirá a query necessária da solicitação.

6 - O **DAO** solicitará uma conexão com o BD ao **ConnectionFactory**.

7 - Com a conexão, o **DAO** enviará a query pelo canal de comunicação, executando-a (e receberá as informações solicitadas).

8 - O **DAO** populará o **JavaBean** com os dados recebidos e o disponibilizará à camada de controle.

9 - Com o **JavaBean** preenchido o Controle irá dispor os dados nos referidos campos da camada de visualização.

10 - Com os campos preenchidos o Usuário poderá ver e manipular os dados necessários.

MVC / DAO / DTO

Solicitação de Alteração de Dados (insert, update, delete)

VIEW

1 - Gera ao controle uma solicitação de alteração de informação.

CONTROL

2 - Valida os dados, e se estiver OK procede solicitação.

3 - Gera um **JavaBean** montando-o com os dados necessários à solicitação.

4 - Envia o **JavaBean** à solicitação equivalente do **DAO** (para que seja encaminhada ao BD).

MODEL

5 – Com o **JavaBean**, o **DAO** construirá a query necessária da solicitação.

6 - O **DAO** solicitará uma conexão com o BD ao **ConnectionFactory**.

7 - Com a conexão, o **DAO** enviará a query pelo canal de comunicação executando-a.

Obrigado !