



Programação Orientada a Objetos 2

GBC055

Prof. Henrique Fernandes



MVC

Padrão MVC

- **M**odelo... modelagem dos dados
- **V**isão... apresentação da aplicação
- **C**ontrole... processamento da aplicação

- Mas o que é esse padrão?

- Para entender...
- Vejamos os **problemas** que levaram a ele

Padrão MVC

- Um mesmo aplicativo...
 - Diversos tipos de equipamento



Processamento
Apresentação

Processamento
Apresentação

Processamento
Apresentação



Processamento

- Um mesmo aplicativo
 - Diversos tipos de equipamento

Processamento
Apresentação

Processamento
Apresentação



**Repetição
de Código!**



Processamento

- Um mesmo aplicativo
 - Diversos tipos de equipamento

Processamento
Apresentação

Processamento
Apresentação



Retrabalho!



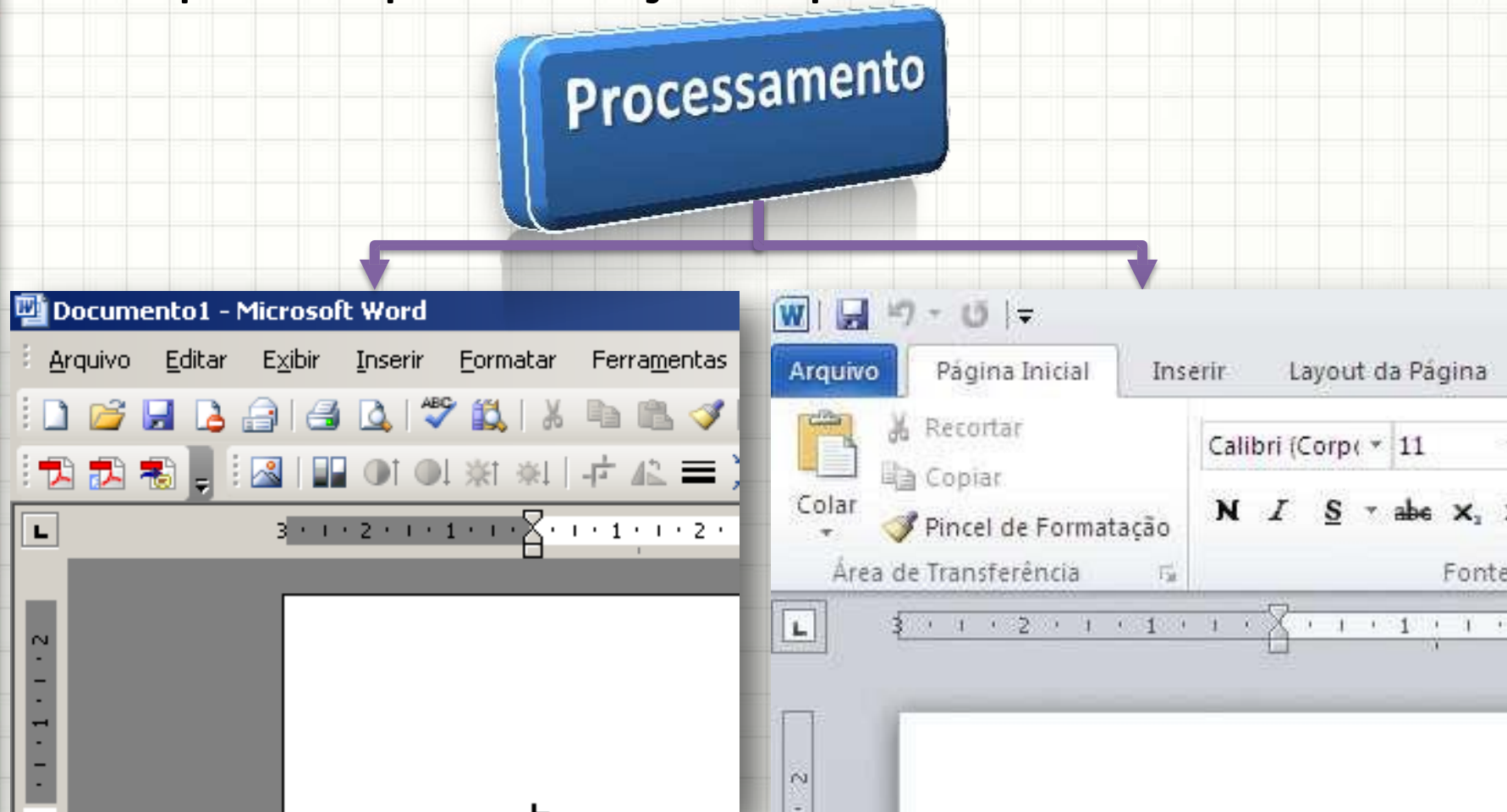
Padrão MVC

- Separar apresentação e processamento?



Padrão MVC

- Separar apresentação e processamento?



Padrão MVC

- Reaproveitar modelos de dados...
 - Cliente é usado em várias aplicações



Padrão MVC

- Separar modelo do restante



Padrão MVC

- **Objetivo:** facilitar desenvolvimento, manutenção e reaproveitamento de código
- Mudanças mais frequentes
 - Interface com o Usuário (visão)
- Mudanças comuns
 - Modelo de dados
- Mudanças menos comuns
 - Processo de negócio (processamento)

Padrão MVC

- **Objetivo:** facilitar desenvolvimento, manutenção e reaproveitamento de código
- Desejamos
 - Mudar a interface sem mudar o resto
 - Mudar os dados sem mudar o resto
 - Mudar o processo sem mudar o resto
- A separação em três conjuntos **facilita!**

Padrão MVC

- **Solução**
- **Modelo**
 - Componentes de Dados e ***Persistência***
 - Armazenam e **validam** os dados (não é na “janela”!)
- **Visão**
 - Componentes de interação com usuário
 - Janelas, formulários etc.
- **Controle**
 - Componentes de processamento
 - Coordenam todos os componentes para produzir o resultado desejado pelo usuário

Padrão MVC

- Visão
- Modelo
 - Controlador

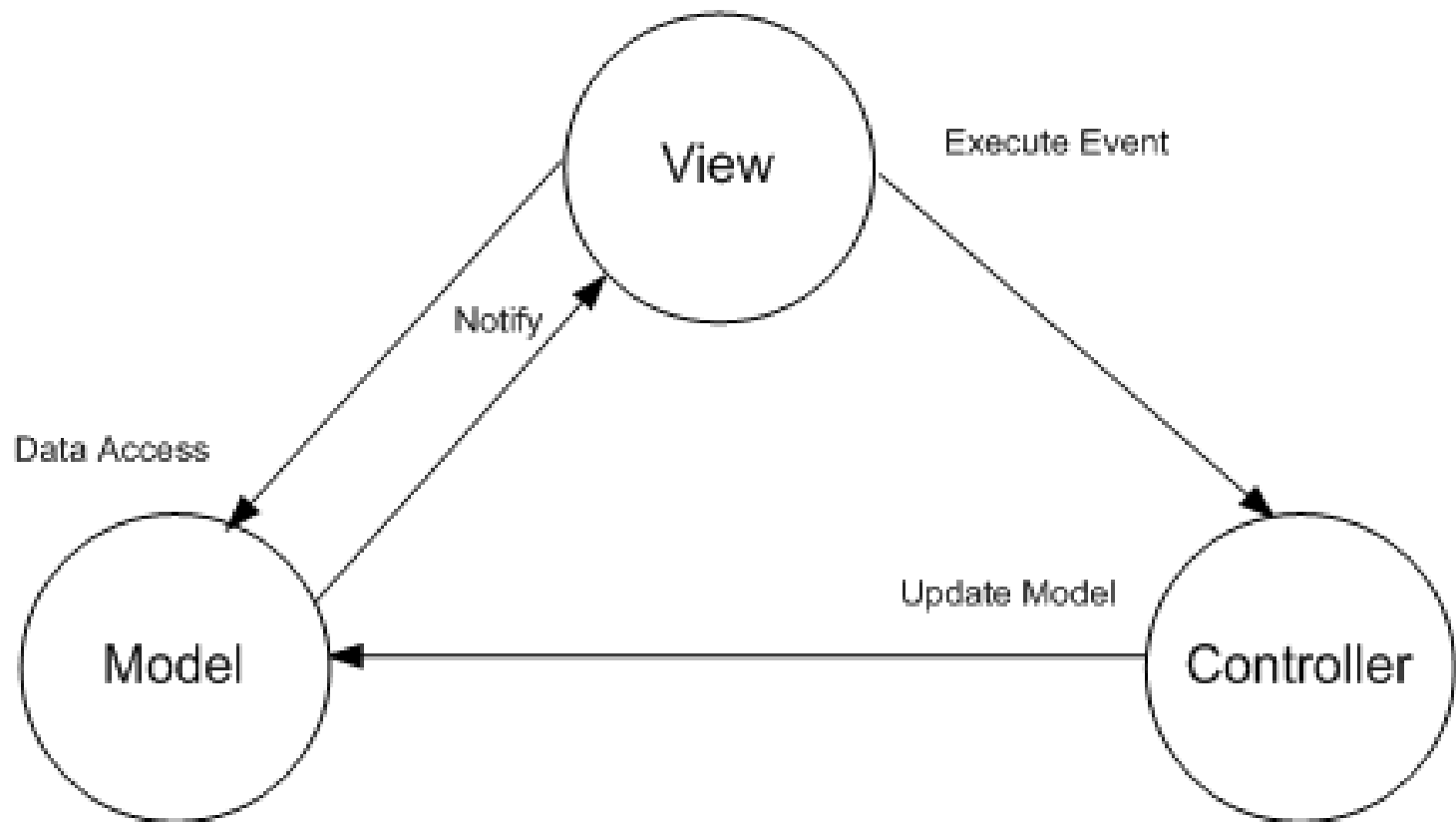
**Um não se
intromete na
função do outro**

Controle

- Componentes de controle
- Coordenam todos os componentes para produzir o resultado desejado pelo usuário

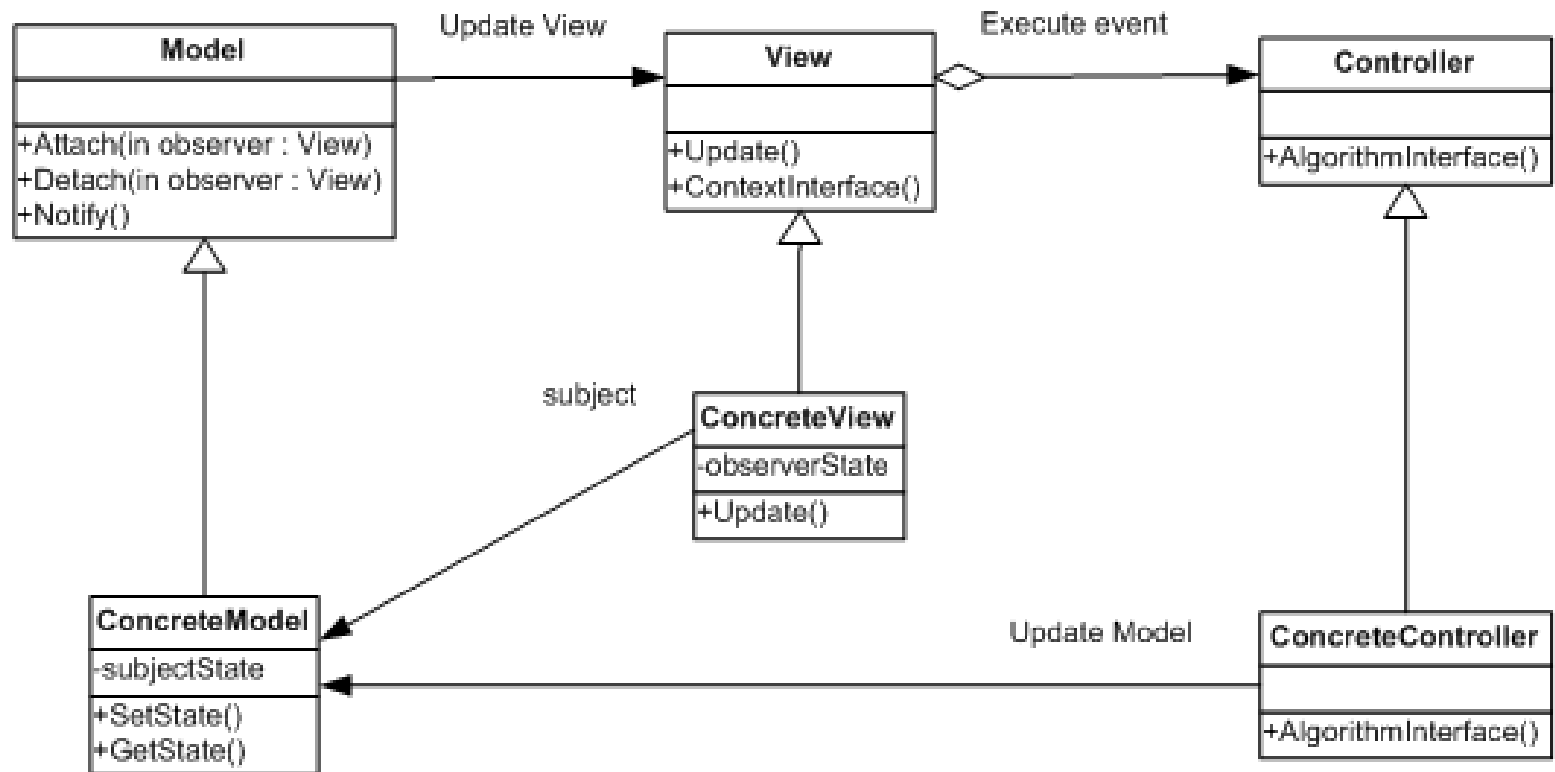
Padrão MVC

- Solução



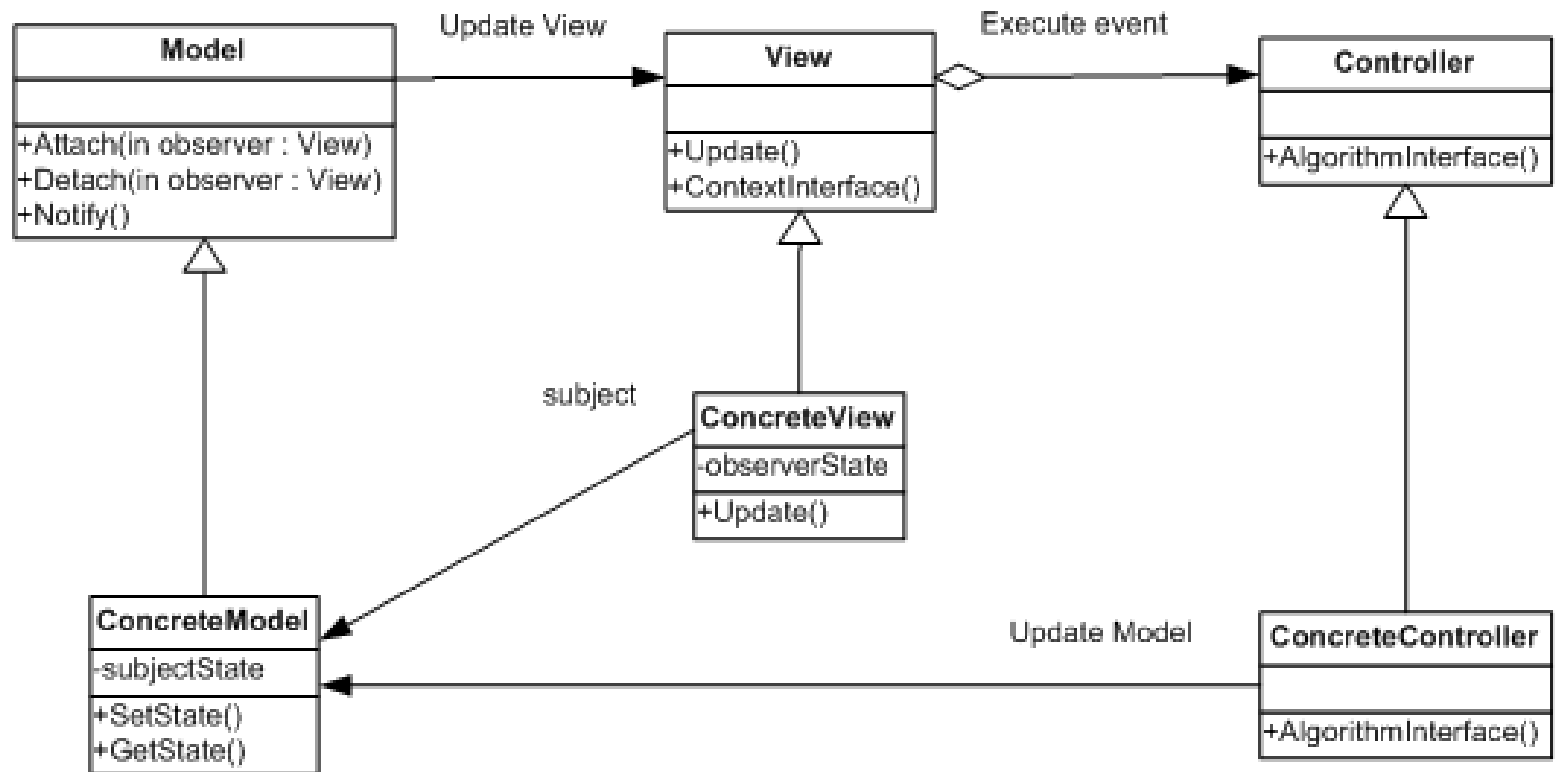
Padrão MVC

- Solução



Padrão MVC

- Solução



Você consegue encontrar algum padrão de projeto que já estudamos até agora dentro do padrão MVC?

Padrão MVC na Web

- No caso de Aplicações Web em Java...
- Visão: HTML e formulários
 - Usualmente JSPs
- Modelo: Classes de Entidade
 - Dados não nativos
- Controle: Processo de negócio
 - Usualmente Servlets

Padrão MVC na Web

- No caso de Aplicações Web em Java...
- Visão: HTML e formulário
– Usualmente Servlet
- Modelo:
– Dados e lógica de negócio
- Controle: Processa o clique do usuário
– Usualmente Servlet

E o banco de dados?



PERSISTÊNCIA

Padrão DAO

- Uso inicial do MVC no PC
 - MVC Nível 1
 - Entidades acessavam o banco de dados



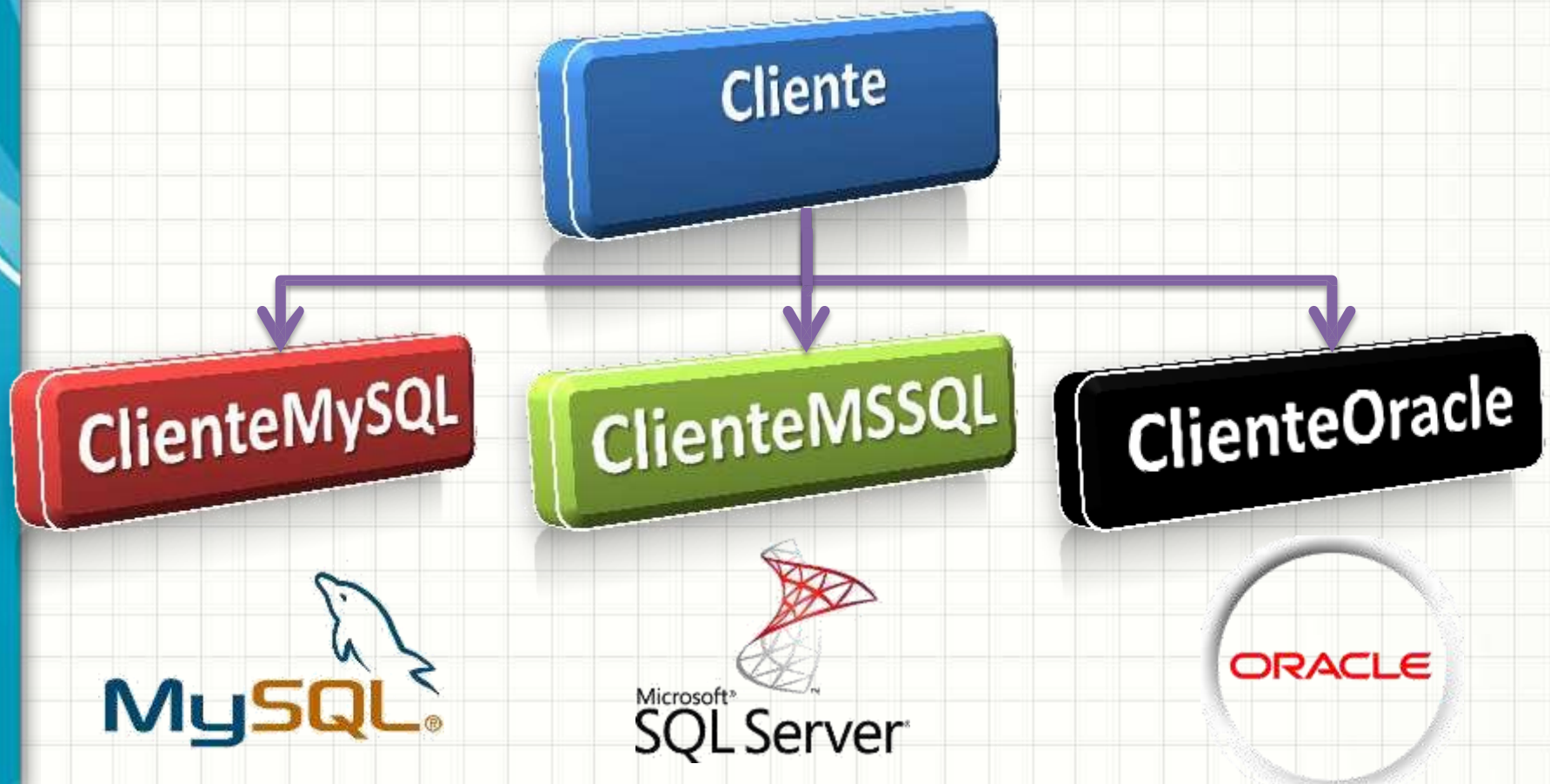
Padrão DAO

- Uso inicial do MVC no PC
 - MVC Nível 1
 - Mas cada contrato exigia o seu banco de dados



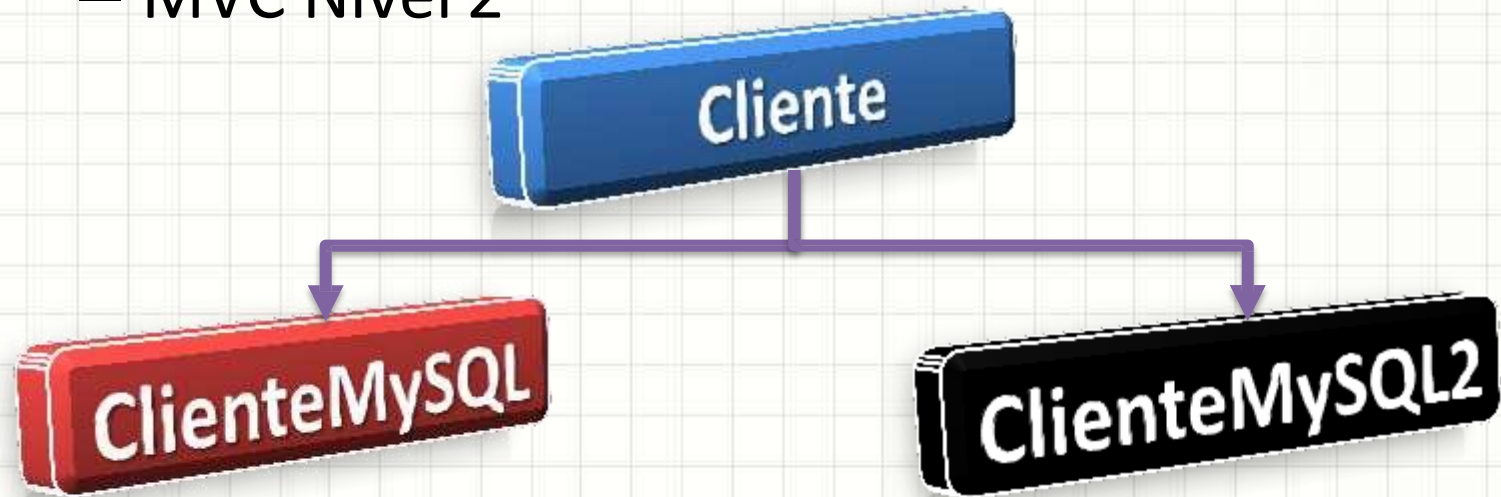
Padrão DAO

- Solução: Separar o acesso ao banco!
 - MVC Nível 2



Padrão DAO

- Solução: Separar o acesso ao banco!
 - MVC Nível 2



Cliente

id	nome	endereço

Cliente

id	nome

Endereco

id	endereço

Padrão DAO

- Esses caras...
 - São objetos de acesso a dados
- **Data**
- **Access**
- **Objects**
- Padrão criado pela Microsoft
- É um “construtor” de objetos com base nas informações do banco de dados



Padrão DAO

- Padrão de Nome
 - NomeDaEntidade**DAO**
- Cliente...
 - Cliente**DAO**
- Produto...
 - Produto**DAO**
- Como usar isso?



Padrão DAO

- Armazenar um cliente?

```
ClienteDAO.acidiona(objeto_cliente);
```

- Buscar um cliente?

```
Cliente c;
```

```
c = ClienteDAO.busca("012.345.678-90");
```

- Fácil!



OUTROS PADRÕES DE PERSISTÊNCIA

Outros Padrões de Persistência

- DAO é o único padrão?
- Claro que não!
- Mas, talvez nem precisemos nos preocupar...
 - Existem *frameworks* prontos para persistência!
 - Castor...
 - Hibernate...
 - Speeder...
 - TopLink...
 - **Dentre muitos outros...**

Outros Padrões de Persistência

- Cada *framework* funcionava de um jeito...
- Então a Sun criou o JPA
 - Java
 - **P**ersistence
 - **A**PI
- O que é uma API?
 - Quais “comandos” deve ter a biblioteca/*framework*
 - Define como esses “comandos” devem funcionar
 - O que fazem
 - Como são usados (nome, parâmetros etc.)

Outros Padrões de Persistência

- Maioria dos frameworks: implementa JPA
- Hibernate JPA
- OpenJPA
- TopLink
- Dentre outros...