



PROGRAMAÇÃO SERVIDOR EM SISTEMAS WEB

PADRÕES MVC E DAO

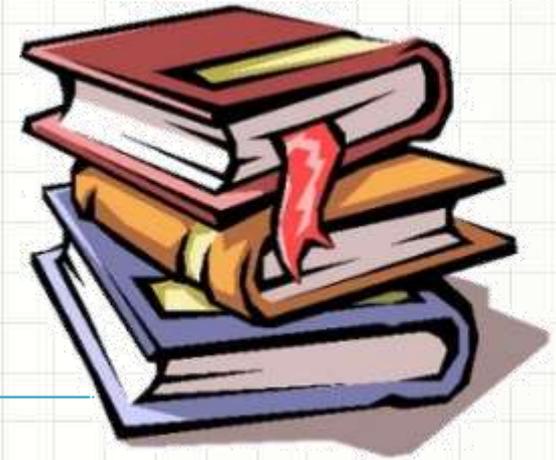
Prof. Dr. Daniel Caetano

2011 - 2

Visão Geral

- 1 • Padrão de Desenvolvimento?
- 2 • O Conceito de Padrão de Projeto
- 3 • Padrão MVC
- 4 • Persistência
- 5 • MVC Nível 1 e 2
- 6 • Padrão DAO
- 7 • Outros Padrões de Persistência

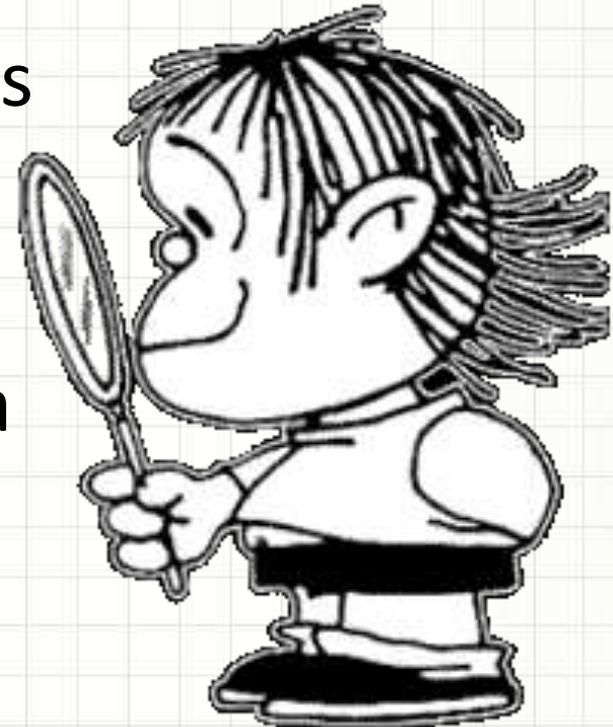
Material de Estudo

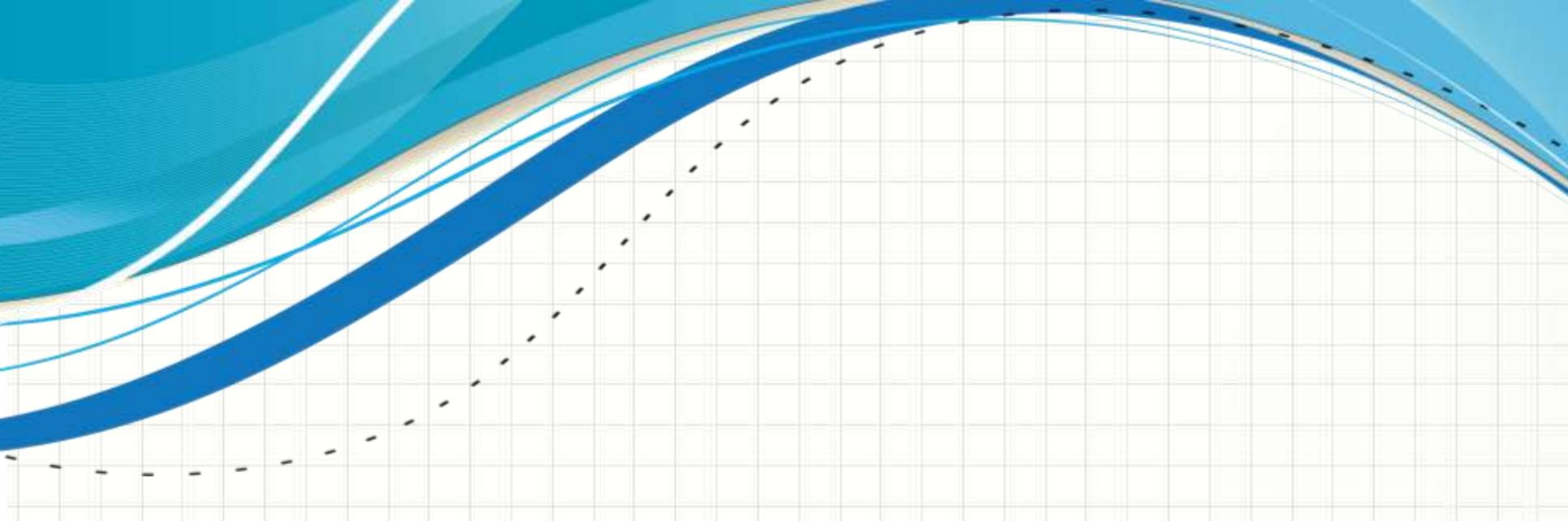


Material	Acesso ao Material
Notas de Aula	http://www.caetano.eng.br/aulas/psw/ (Aula 8)
Apresentação	http://www.caetano.eng.br/aulas/psw/ (Aula 8)
Material Didático	Padrões de Arquitetura de Aplicações Corporativas, páginas 37 a 44, 71 a 77 e 295 a 317
Biblioteca Virtual	Java: Como Programar (6ª Edição) Páginas 894 a 904 e 906 a 918 Padrões de Projeto
Google	“Design Patterns”

Objetivos

- Compreender o conceito de Padrões de Projeto
- Compreender o Padrão MVC
- Conhecer o princípio de alguns dos padrões de persistência, como o padrão DAO
- Tomar primeiro contato com a ideia da *Java Persistence API*





O QUE É PADRÃO DE DESENVOLVIMENTO?

Introdução

- Nos primeiros exemplos...
 - Tínhamos um Servlet único...
 - Que foi dividido: Processamento e Apresentação



Introdução

- Essa separação é uma ideia aleatória?
- Será que é a melhor forma de fazer?



Padrões de Desenvolvimento

- Desenvolvimento → Dúvida
- Como dividir o sistema em partes menores?
 - Partes pequenas o suficiente (para implementar)
 - Grandes o suficiente (para fazer algo significativo)
- Diretrizes
 - ↑ Manutibilidade (flexibilidade e extensibilidade)
 - ↓ Custos
 - ↑ Reusabilidade
- Como fazer?

Padrões de Desenvolvimento

- Muitas coisas para pensar...
- Como fazer?



Padrões de Desenvolvimento

- Ei, que tal fazer como nossos ancestrais têm feito?



- É! Vamos lá resolver essa parada...

Padrões de Desenvolvimento

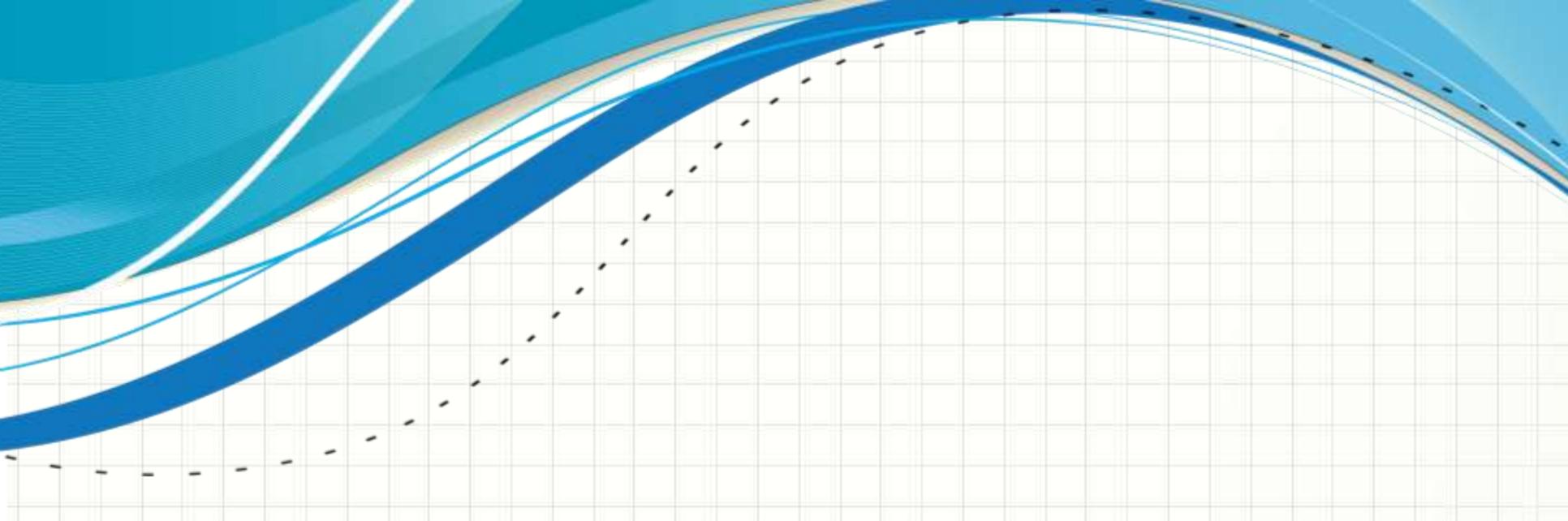
- Padrões de Desenvolvimento
 - Conjunto das “**Melhores Práticas**” conhecidas
- Isso significa que não vamos ter que pensar?



Padrões de Desenvolvimento

- Pensar no que é importante...
 - Vamos direto ao ponto!





PADRÕES DE PROJETO

O Que é um Padrão de Projeto

- Dividir e Organizar um Sistema...
 - ...de maneira elegante
- Solucionar problemas comuns...
 - ... de maneira elegante, mas não óbvia!

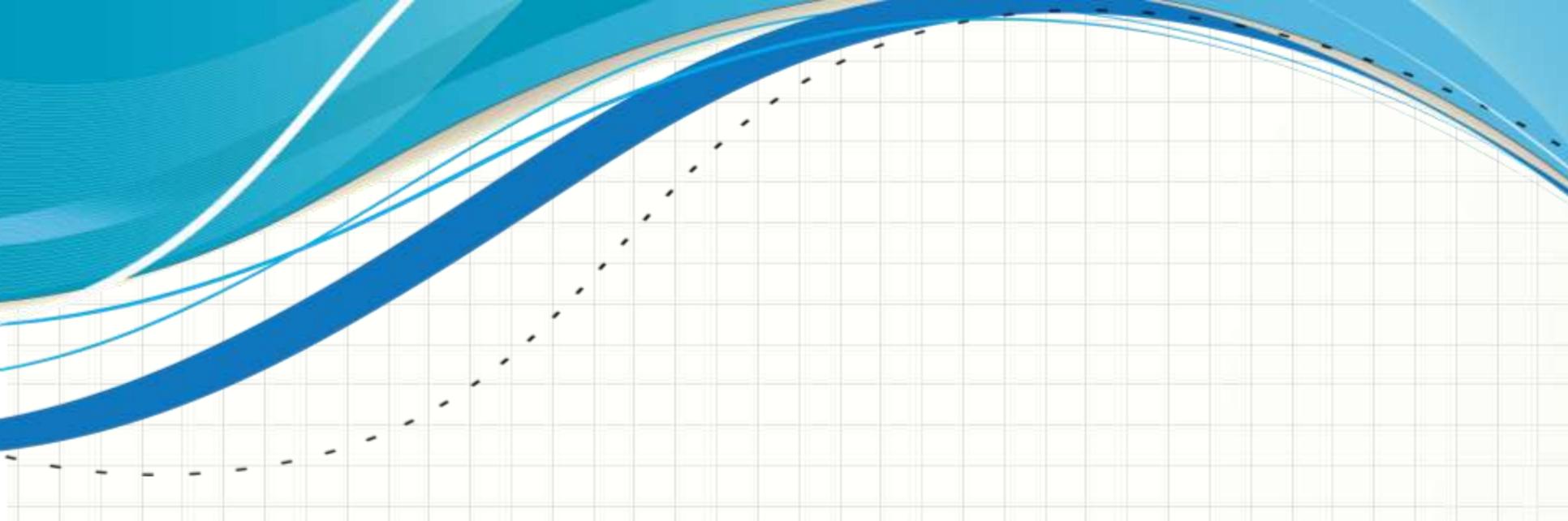


O Que é um Padrão de Projeto

- Há diversos níveis de “detalhamento”
- Há padrões “mais genéricos”
 - Diretrizes sobre organização
 - Padrões de Arquitetura
- Há padrões “mais específicos”
 - Como resolver um problema específico
 - “Padrões de Implementação”
 - Padrões de Criação
 - Padrões de Estrutura
 - Padrões de Comportamento

O Que é um Padrão de Projeto

- Todo padrão de projeto tem:
 - Um **nome**
 - Que identifique claramente o que o padrão faz
 - Um **objetivo**
 - Que descreva com detalhes a razão de ser do padrão
 - Um **problema**
 - Uma situação de projeto cuja solução não é óbvia
 - Uma **solução**
 - Detalhes de implementação da solução elegante
 - **Consequências**
 - Não existe almoço grátis...



PADRÕES MVC

Padrão MVC

- **Modelo...** modelagem dos dados
- **Visão...** apresentação da aplicação
- **Controle...** 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
Apresentação

Processamento
Apresentação

Processamento
Apresentação



**Repetição
de Código!**



Processamento
Apresentação

Processamento
Apresentação

Processamento
Apresentação



Retrabalho!



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 os dados
- **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

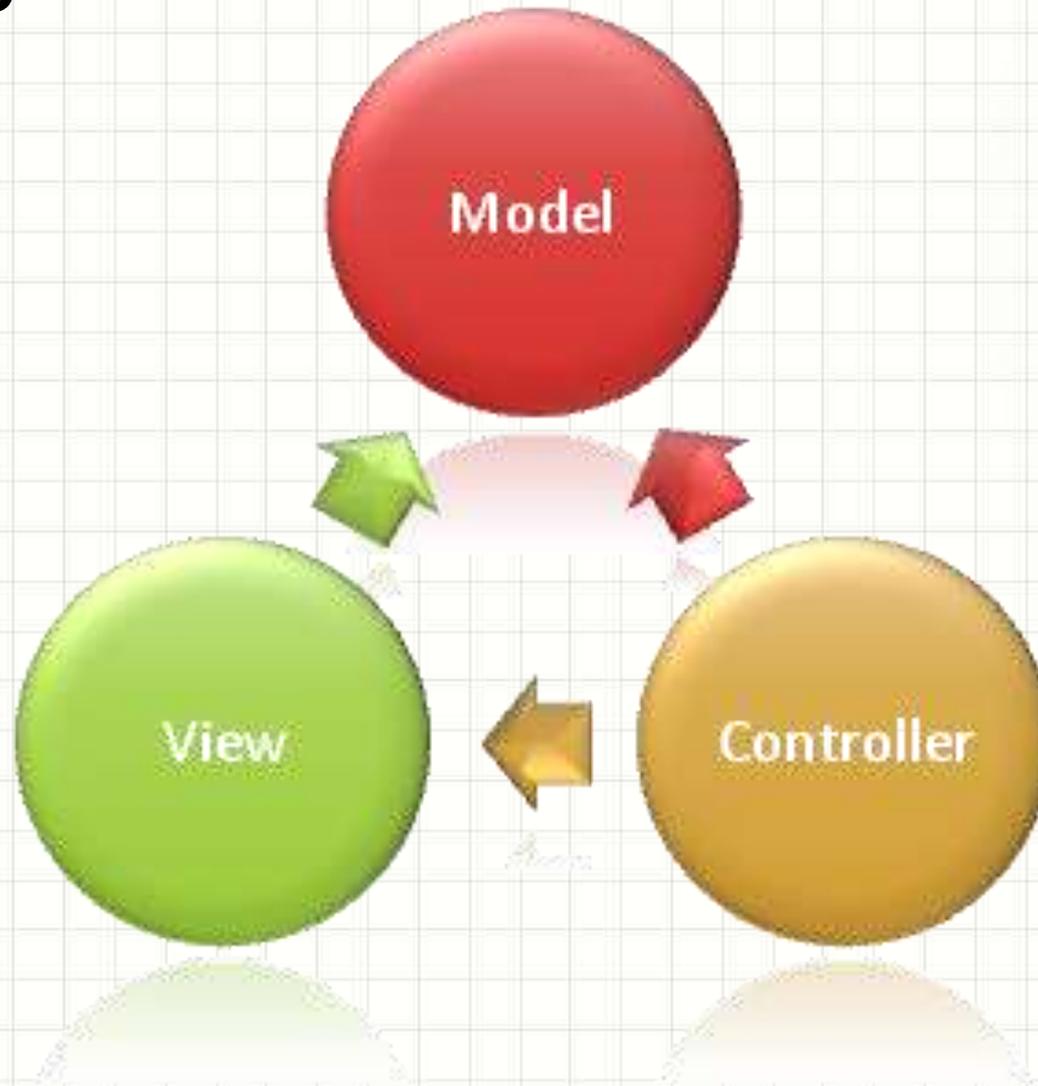
- Visão
- Modelo
- Controlador

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

- Controle
- Controlador
- Controladores coordenam todos os componentes para produzir o resultado desejado pelo usuário

Padrão MVC

- Solução

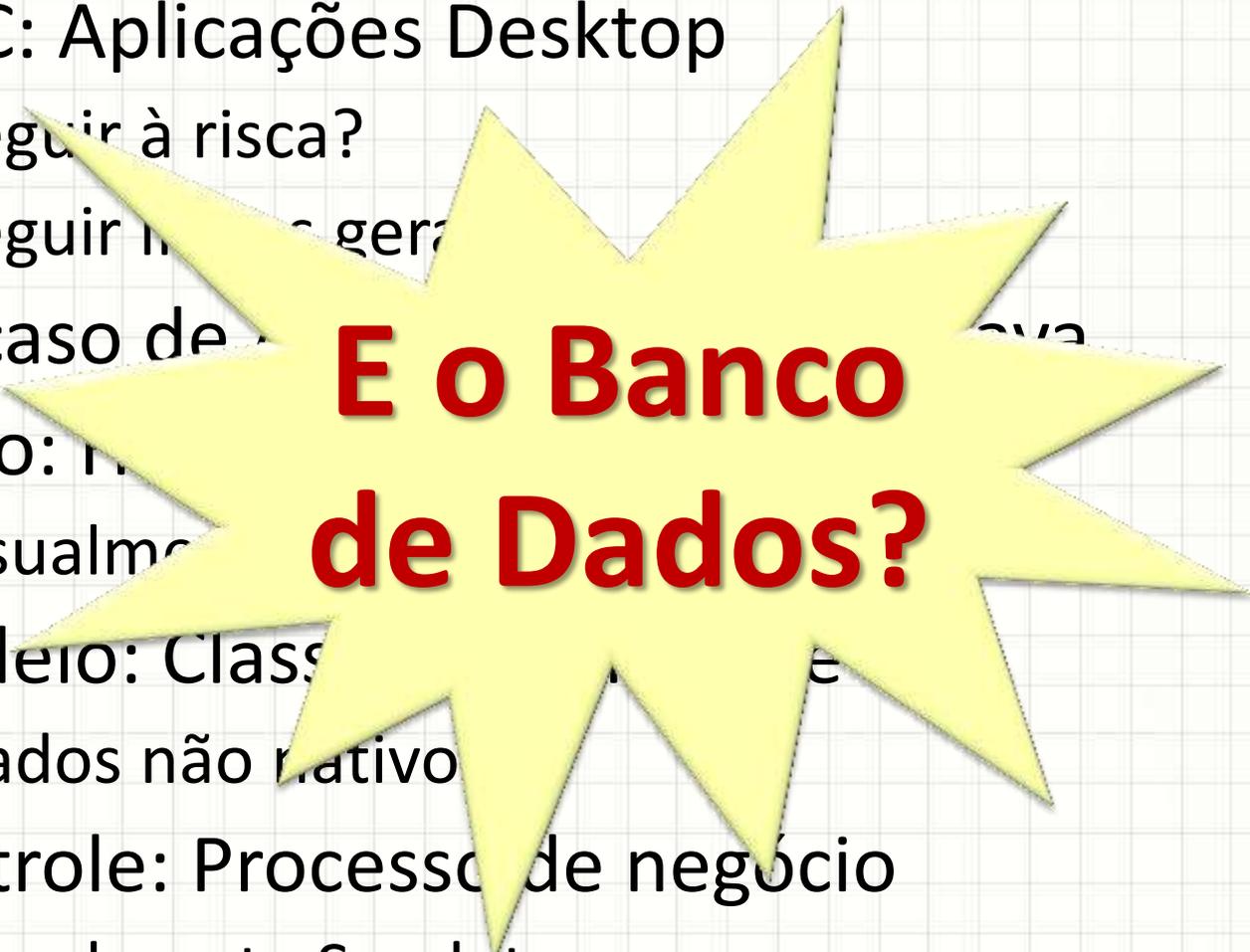


Padrão MVC na Web

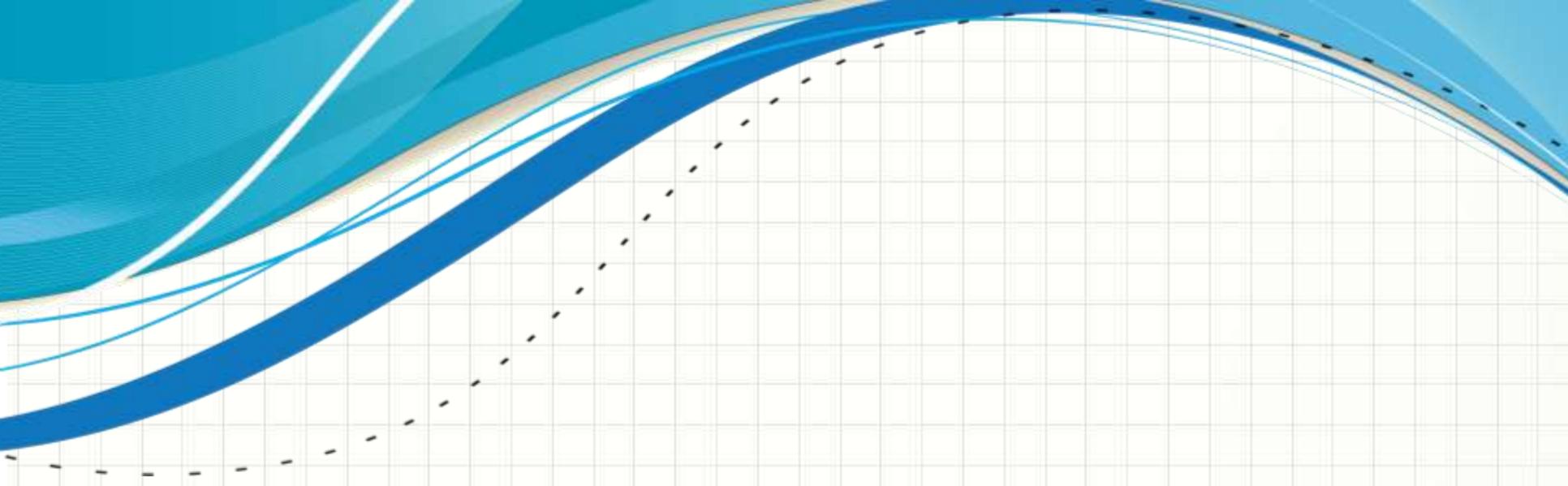
- MVC: Aplicações Desktop
 - Seguir à risca?
 - Seguir linhas gerais?
- No caso de Aplicações Web em Java...
- Visão: HTML e formulários
 - Usualmente JSPs
- Modelo: Classes de Entidade
 - Dados não nativos
- Controle: Processo de negócio
 - Usualmente Servlets

Padrão MVC na Web

- MVC: Aplicações Desktop
 - Seguir à risca?
 - Seguir modelo gerado
- No caso de ...
- Visão: ...
 - Usualmente ...
- Modelo: Classes ...
 - Dados não nativos
- Controle: Processos de negócio
 - Usualmente Servlets



**E o Banco
de Dados?**



PERSISTÊNCIA

Persistência

- Modelo MVC
 - Concebido para sistemas de memória persistente
 - Exemplo atual: celular com java
 - Não há disco x memória
 - O programa nunca para de executar... ele só é “congelado”
 - Se você criou um cliente, ele fica lá!
 - Para que armazenar?
- Isso é o mesmo que ocorre no PC?

Persistência

- No PC, a memória é **volátil!**
- Desligar: cliente da memória “vai pro brejo”
- Precisamos “simular” a **persistência**
 - Criar um elemento que armazene e recupere nossos clientes quando necessário
- A ideia é armazenar os dados não nativos em um banco de dados!
 - Dados Não Nativos: **Classes de Entidade ...**
 - ... a mesma do **Entidade-Relacionamento**

Persistência

- Existem duas formas básicas de implementar a persistência
- MVC Nível 1
 - A **própria entidade** cuida de sua persistência
 - Ela é quem acessa o banco de dados...
- MVC Nível 2
 - Um **elemento externo** cuida da persistência
 - Isola as classes de entidade do banco de dados



PADRÃO DAO

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

- 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?

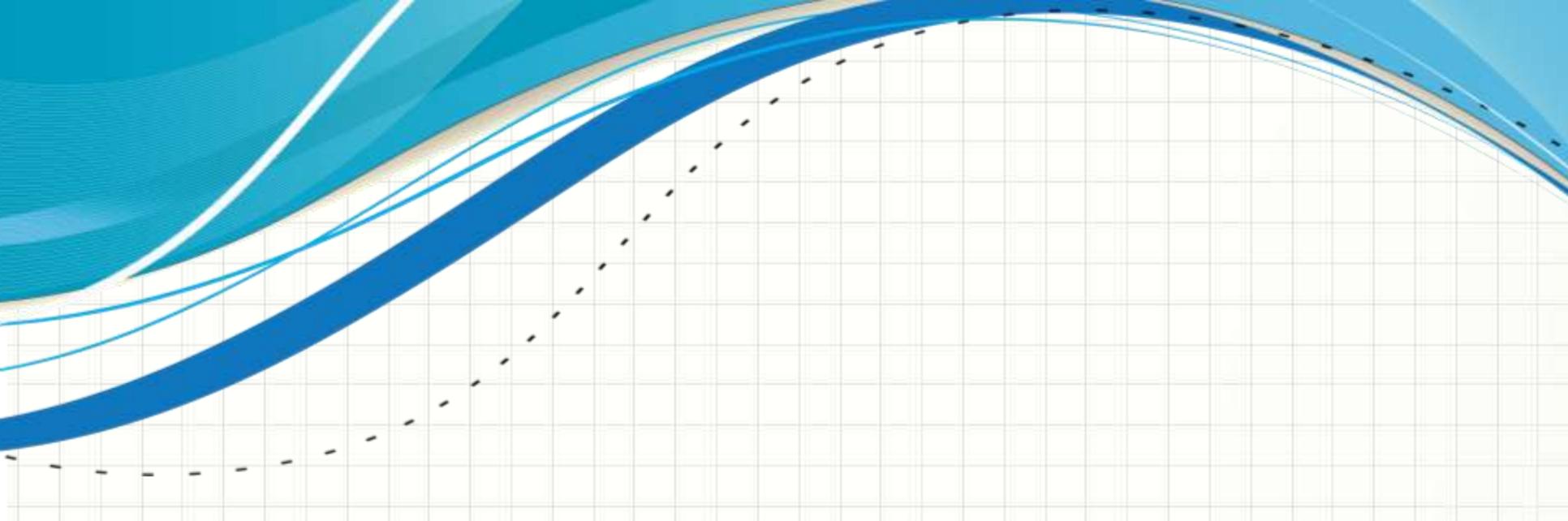
```
ClienteDAO.acidiona(objeto_cliente);
```

- Buscar um cliente?

```
Cliente c;
```

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

- Fácil!



OUTROS PADRÕES DE PERSISTÊNCIA

Outros Padrões de Persistência

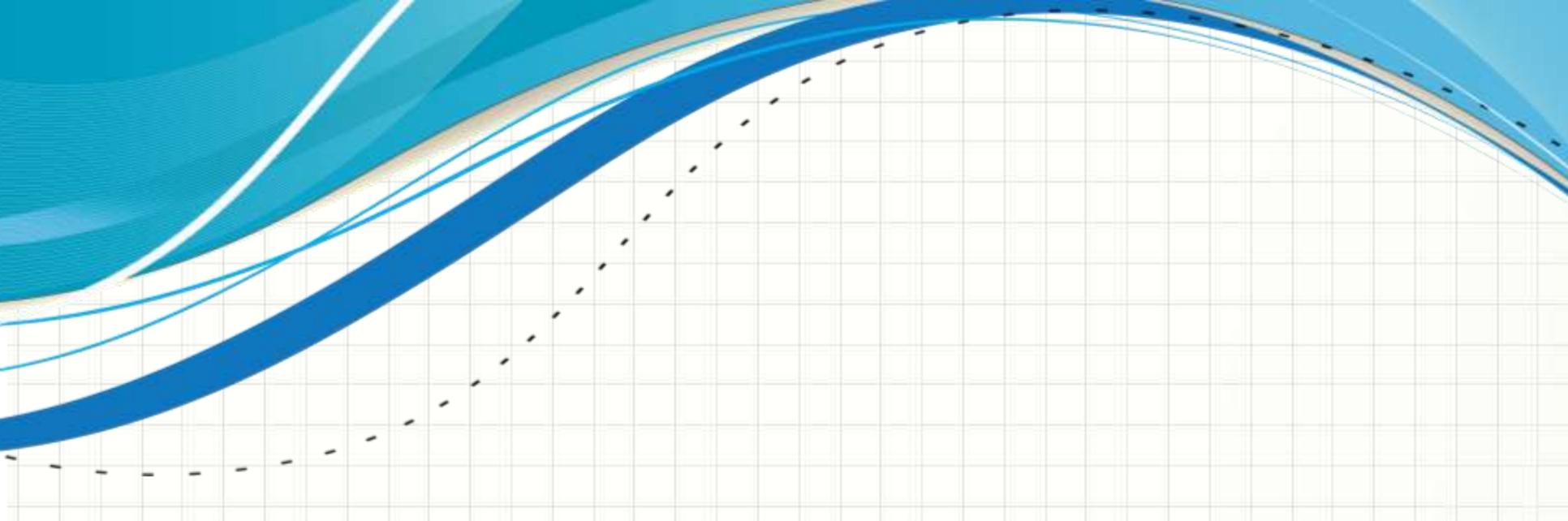
- DAO é o único padrão?
- Claro que não!
- Mas, talvez nem precisemos nos preocupar...
 - Existem *frameworks* prontos para persistência!
 - Castor...
 - Hibernate...
 - Speeder...
 - TopLink...
 - **Dentre muitos outros...**

Outros Padrões de Persistência

- Cada *framework* funcionava de um jeito...
- Então a Sun criou o JPA
 - Java
 - Persistence
 - API
- 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...
-
- Veremos JPA no final do curso!



CONCLUSÕES

Resumo

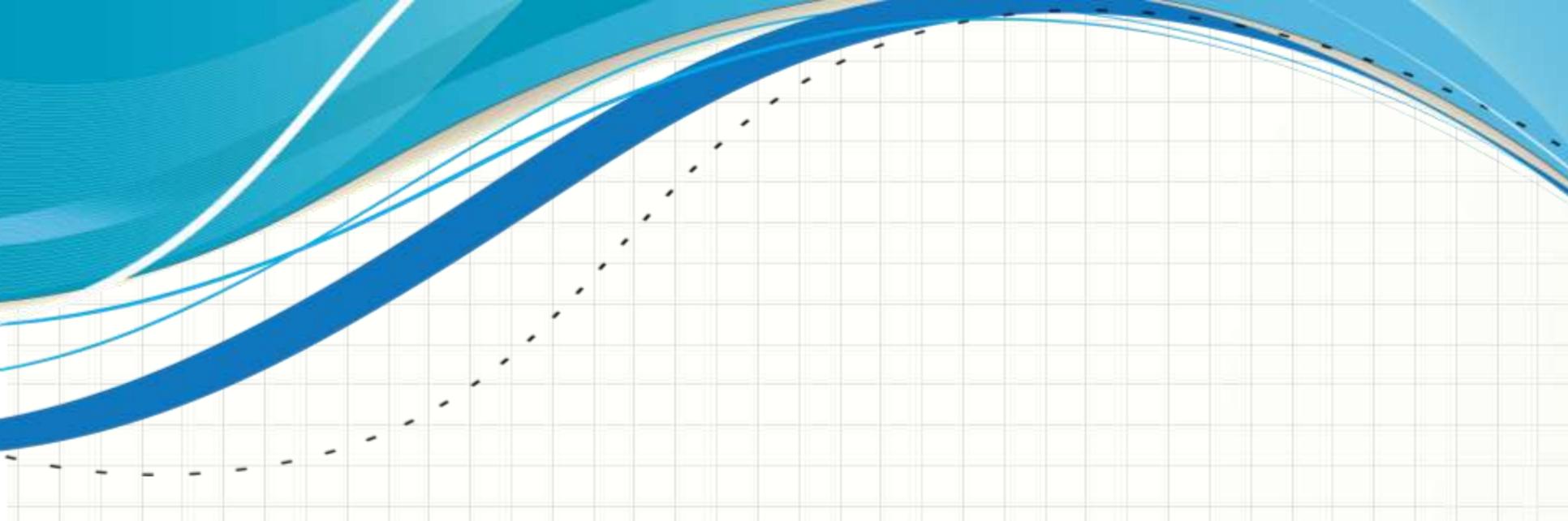
- Padrões de Desenvolvimento oferecem diretrizes ou soluções para resolver problemas comuns
- Não resolve a vida do programador, mas simplifica!
- Às vezes implementar um padrão **não** é o jeito mais fácil, mas é o **melhor a longo prazo**

- **TAREFA**
 - Trabalho 2a para turmas de 4ª!

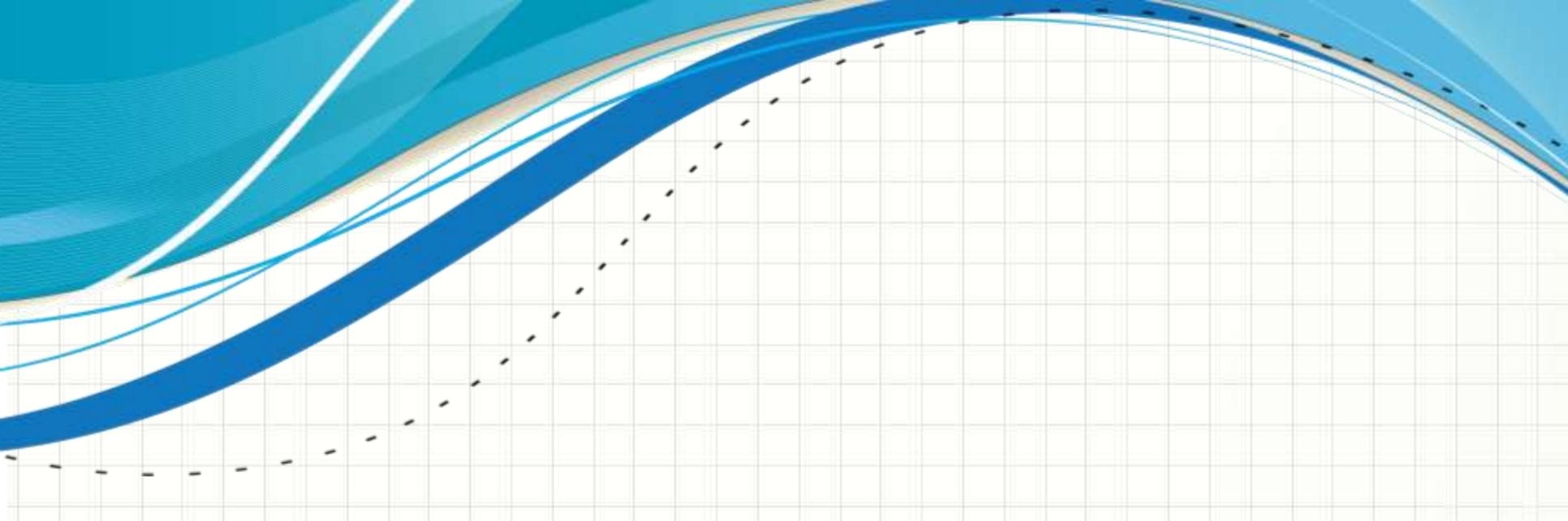
Próxima Aula



- Como implementar isso tudo?!
 - Como criar a entidade (bean)
 - MVC Nível 1



PERGUNTAS?



**BOM DESCANSO
A TODOS!**