



# **PROGRAMAÇÃO SERVIDOR EM SISTEMAS WEB**

## **MIDDLEWARE JDBC: CRIAÇÃO DE BEAN**

Prof. Dr. Daniel Caetano

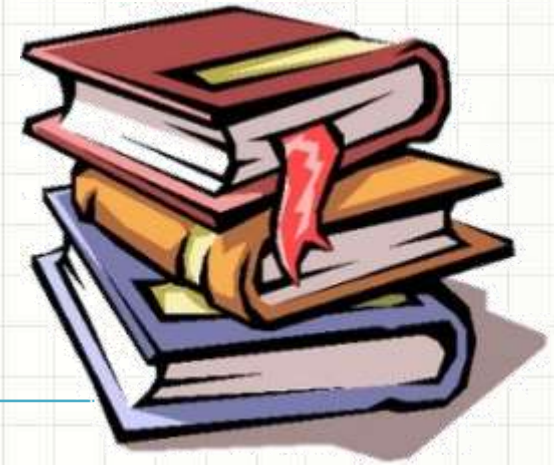
2012 - 1

# Objetivos

- Entender o conceito de Bean
- Capacitar para a implementação de um Bean simples, seguindo MVC Nível 1
- Capacitar para o uso de Java Beans do tipo desenvolvido
- **TAREFA!**
  - Atividade Estruturada 1!



# Material de Estudo



---

<b>Material</b>	<b>Acesso ao Material</b>
Notas de Aula	<a href="http://www.caetano.eng.br/aulas/psw/">http://www.caetano.eng.br/aulas/psw/</a> (Aula 9)
Apresentação	<a href="http://www.caetano.eng.br/aulas/psw/">http://www.caetano.eng.br/aulas/psw/</a> (Aula 9)
Material Didático	Padrões de Arquitetura de Aplicações Corporativas, páginas 295 a 313 (avançado)
Biblioteca Virtual	Java: Como Programar (6ª Edição) Páginas 894 a 918

---



**O QUE É UM BEAN?**

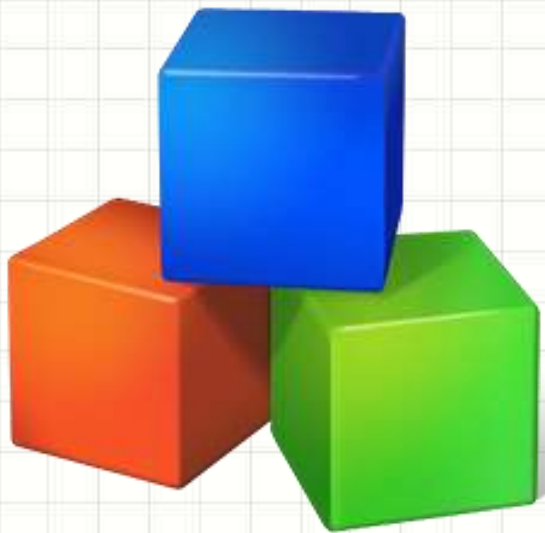
# O que é um Bean?

- “Bean”: termo para entidade
  - Um objeto que representa uma entidade



# O que é um Bean?

- “Bean”: termo para entidade
  - Um objeto que representa uma entidade
  - Entidades possuem atributos...



# O que é um Bean?

- É uma boa prática...
  - Manter os atributos privados!



# O que é um Bean?

- É uma boa prática...
  - Manter os atributos privados!





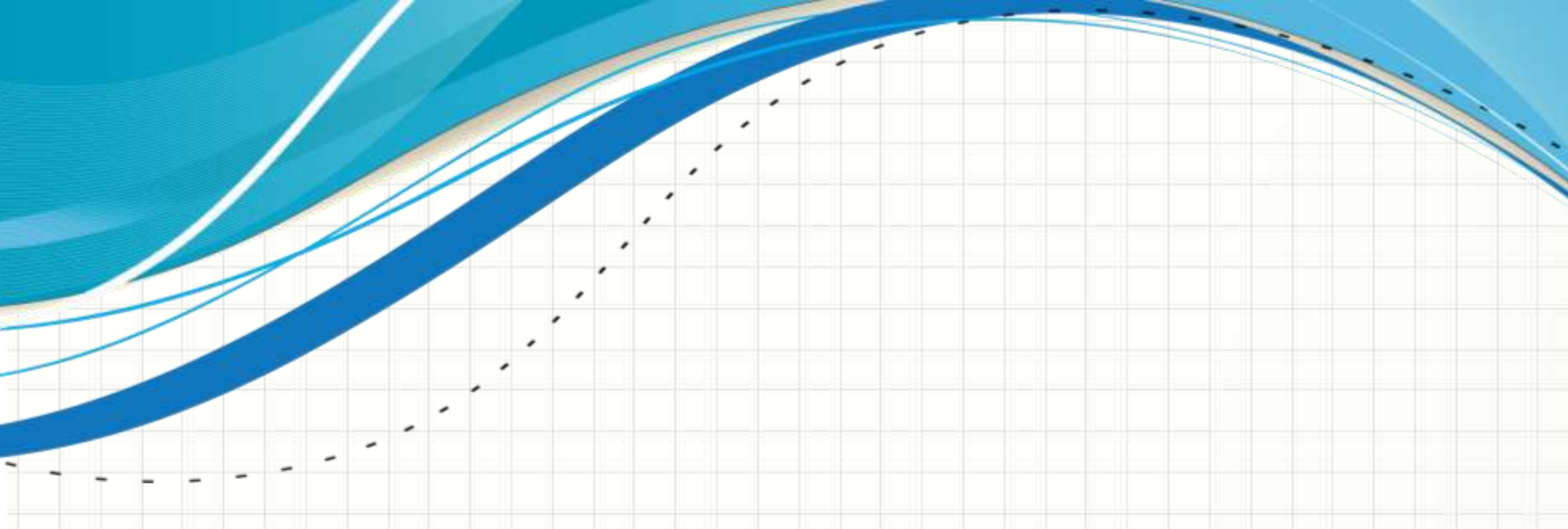
# O que é um Bean?

- É uma boa prática...
  - Manter os atributos privados!
  - Acessar através de *getter* e *setter*



`e.setCor("azul")`





# **CONSTRUINDO UM BEAN**

# Construindo um Bean

- Crie um projeto WProjeto5
- Crie um pacote **entidades**
- Construa um bean Cliente (classe tradicional)
  - **nome** (String)
  - **cpf** (String)
- *Getters e Setters* (com validações)
  - `getNome() / setNome()`
  - `getCpf() / setCpf()`
- Impressão
  - `toString()`
- Teste usando o **index.jsp!**



# MÉTODO PERSIST

# Método **persist**

- Persist: persistir, permanecer
- Uma entidade deve ser armazenada no BD



id (PK)	cor

An orange arrow points from the **persist()** method in the Entity class to the database table, indicating the flow of data from the object to the database.

# Método **persist**

- Persist: persistir, permanecer
- Uma entidade deve ser armazenada no BD



id (PK)	cor
1	“azul”

An orange arrow points from the 'persist()' method in the Entity class to the first row of the table, indicating the mapping of the object to the database record.

# Método **persist**

- Para isso precisamos criar uma tabela no DB
- Acompanhe o professor:
  - Criar o banco de dados **sisclientes**
  - Criar a tabela **cliente**
  - Criar o método **persist**
    - Liga o código ao driver do JavaDB
    - Conecta no banco de dados
    - Cria objeto de transação
    - Armazena os dados do próprio objeto
    - Fecha transação
    - Fecha conexão



# USANDO UM BEAN



# Usando o Bean – Passo 1

- Para que usar um Bean?
- Dados que precisam “persistir”
  - Permanecer depois que o sistema foi desligado
- Modifique o index.jsp para que redirecione para um Servlet com um link
- Crie o servlet **NovoCliente** que execute as tarefas:
  - Crie um objeto Cliente
  - Persista o objeto

# Usando o Bean – Passo 2

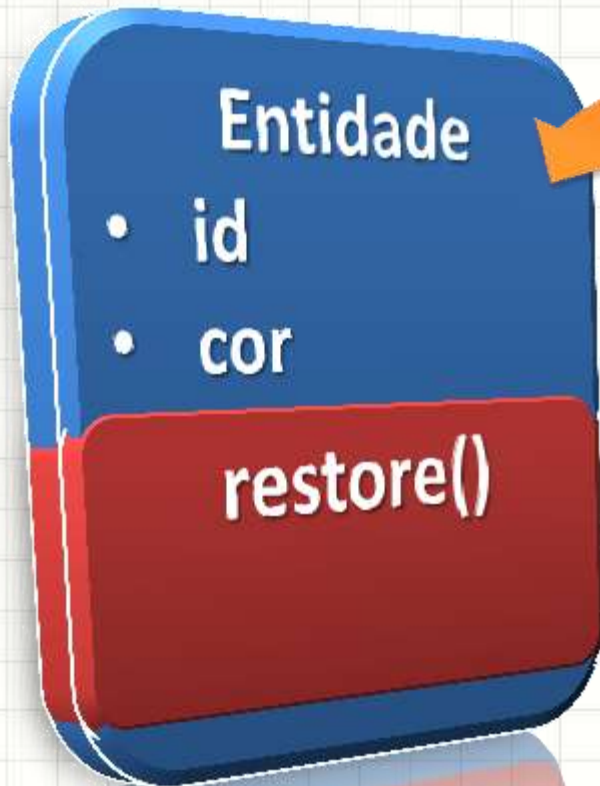
- Agora...
- Crie um JSP chamado **ClienteView.jsp**
  - Recebe um cliente na requisição
  - Imprime esse cliente
- Altere o NovoCliente para usar **ClienteView.jsp**



# O MÉTODO RESTORE

# Método **restore**

- Restore: restaurar
- Uma entidade deve ser recuperada do BD



id (PK)	cor
1	“azul”

# Método **restore**

- Criar o método **restore(String cpf)**
  - Liga o código ao driver do JavaDB
  - Conecta no banco de dados
  - Cria objeto de transação
  - Recupera os dados no próprio objeto
  - Fecha transação
  - Fecha conexão
  
- Teste no **NovoCliente**
  - Ao invés de configurar o objeto, use o restore

# Método **restore**

- Criar objeto e restaurar...?
- **Que tal criar já restaurando?**
- Como executar algo logo que o objeto é criado?
  - Método **construtor**
  - Criar construtor que recebe parâmetro **cpf**
  - Fazer o construtor chamar o método **restore**
  - Toda construção precisa fazer **restore**?
    - **Só aquelas que receberem um CPF**
- Experimente!



**CONCLUSÕES**

# Resumo

- Um bean é um objeto de entidade
- Usualmente desejamos que as informações de um bean possam persistir
- Uma das formas de fazer isso é por meio de um sistema MVC Nível 1, com métodos persist e restore em cada classe de entidade
- **TAREFA**
  - **Atividade Estruturada 1!**



# Próxima Aula



- MVC Nível 2
  - Separando o Bean do Acesso ao BD (Padrão DAO)
- Padrão Front Control / View Control
  - Múltiplas funções, menos Servlets



**PERGUNTAS?**



**BOM DESCANSO  
A TODOS!**