



Prog Web

Edson Orivaldo Lessa Junior

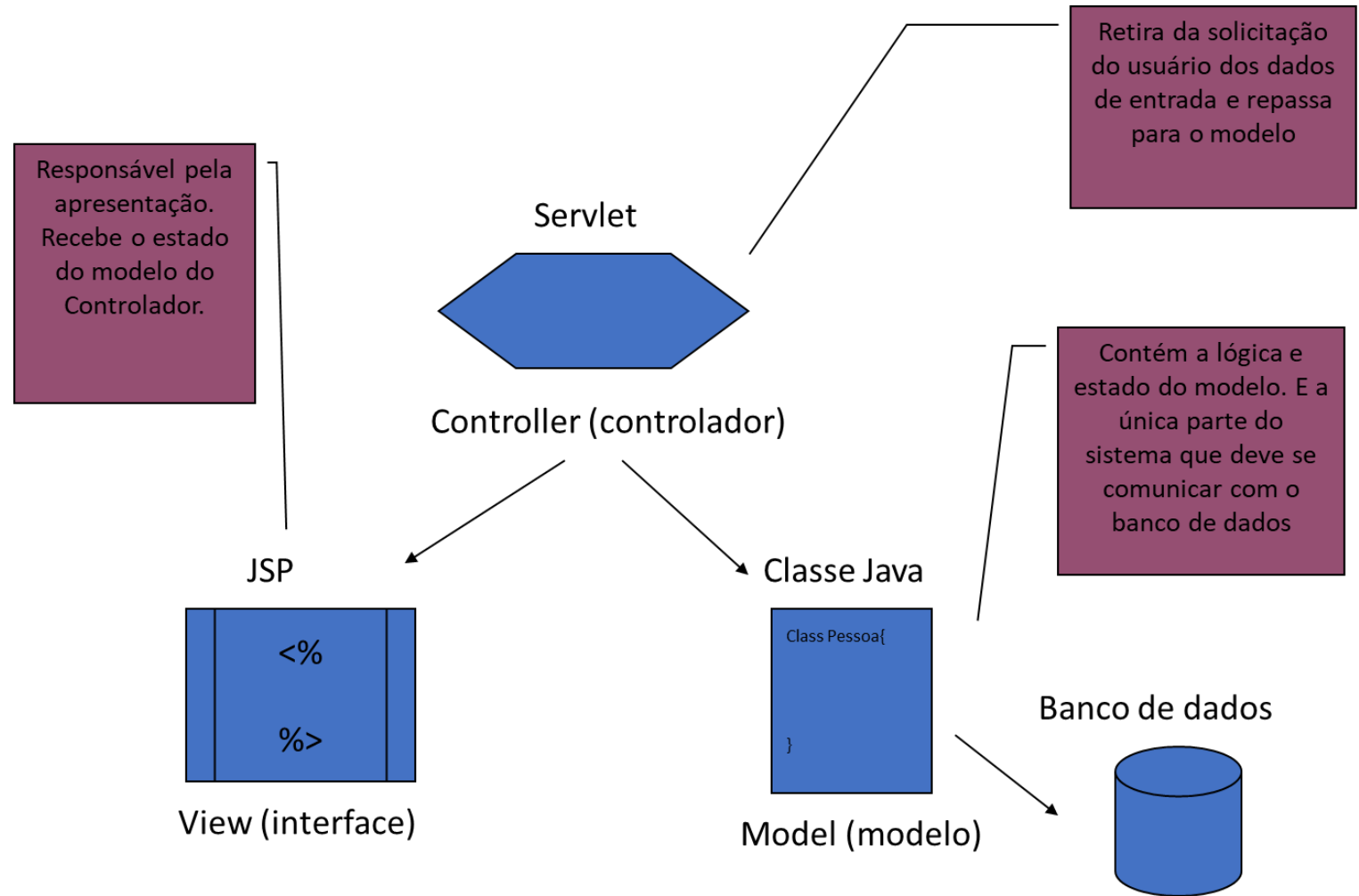
MVC - Model-View-Controller

- A essência do MVC é separar a lógica do negócio da apresentação (interface), de modo que essa lógica possa servir para qualquer forma de apresentação (HTML, Swing, Web service, etc).
- Quando as aplicações misturam códigos de apresentação, de acesso a dados e de lógica ou regra de negócios, tornam-se difíceis de manter.
- Primeiro por que a interdependência dos componentes faz com que alterações no código se propaguem.
- Segundo, por que o forte acoplamento torna difícil ou impossível a reutilização das classes;
- Aumenta a complexidade do projeto.

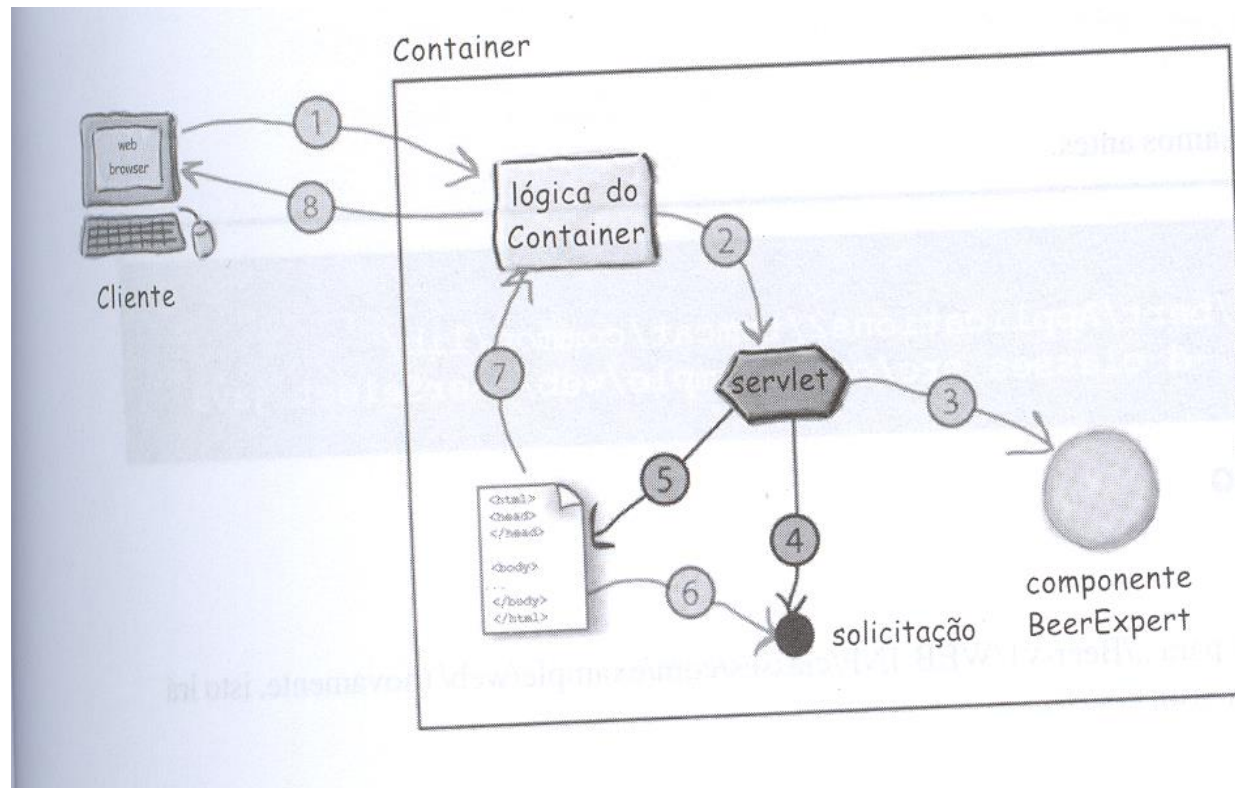
Passos para implementar MVC com Servlets e JavaServer Pages

- Crie beans para representar os dados;
- Crie um Servlet para tratar requisições, invocar regras de negócio e código de acesso a dados;
- Instancie os beans no Servlet;
- Armazene os beans na requisição, sessão ou contexto do Servlet
- Despache a requisição para um JSP
- Extraia os beans e exiba os dados

MVC



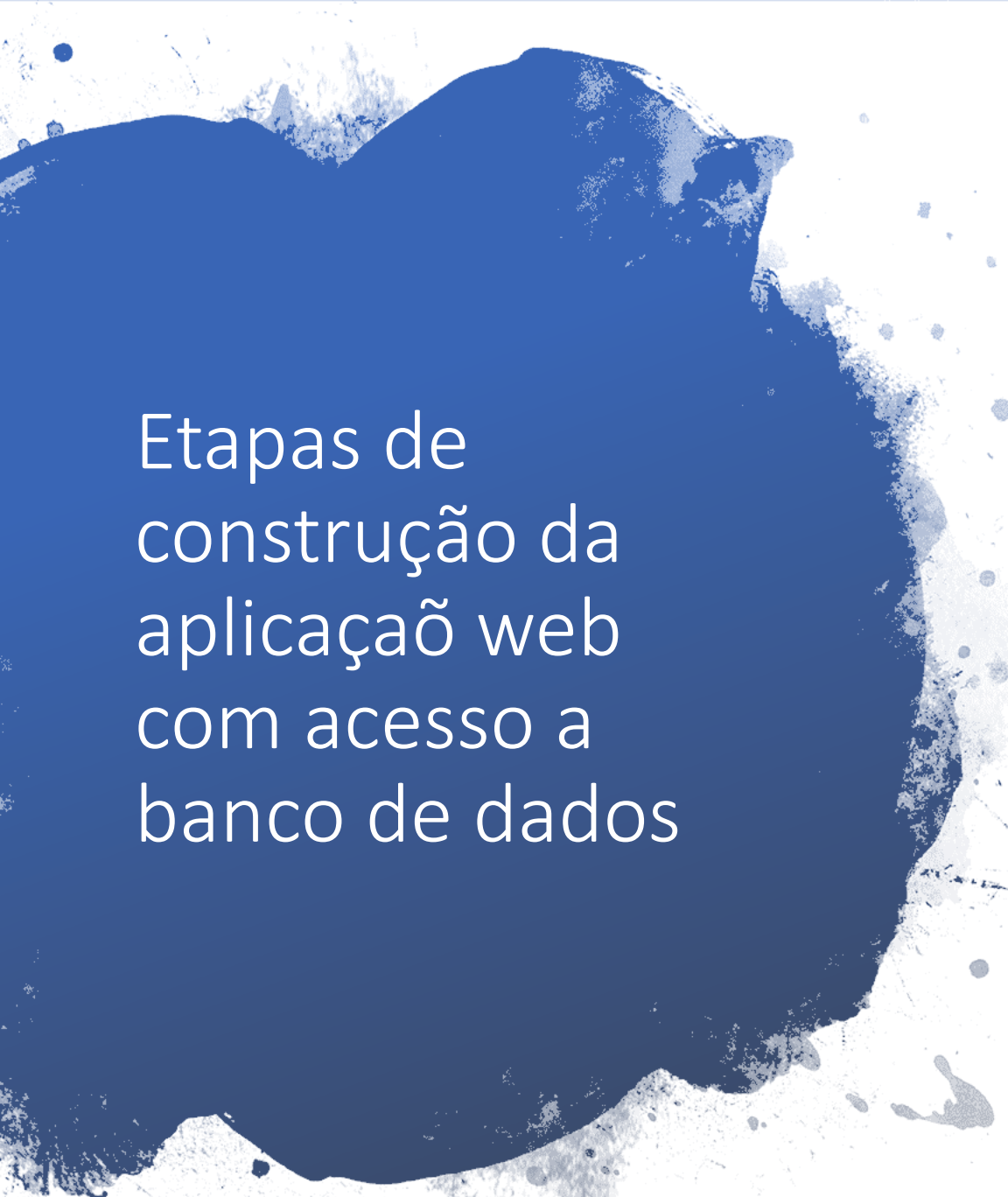
Fluxo de execução com MVC



- 1 - O browser envia os dados da solicitação para o container.
- 2 - O Container encontra o servlet correto baseado na URL e passa a solicitação ao servlet.
- 3 - O servlet chama o BeerExpert para ajudar.
- 4 - A classe responsável retorna uma resposta, que o servlet adiciona ao objeto solicitação.
- 5 - O servlet despacha para o JSP.
- 6 - O JSP recebe a resposta originada do objeto solicitação.
- 7 - O JSP gera uma página para o Container.
- 8 - O Container retorna a página para o usuário feliz.


Atividade

- Desenvolver uma aplicação JSP para controle de pessoas e projetos.
- A aplicação deve conter na primeira tela um “menu” onde o usuário poderá escolher uma das opções: Cadastrar ou Consultar
- Ao clicar na opção Cadastrar, uma tela com as opções Pessoa e Projeto deve ser exibida
- Ao clicar na opção Pessoa uma tela com formulário deve ser exibida. Esse formulário deve conter os seguintes campos: nome e cargo
- Ao clicar na opção Projeto uma tela com formulário deve ser exibida. Esse formulário deve conter os seguintes campos: nome do projeto, data de inicio, data prevista de término
- Ao clicar na opção Consultar, uma tela com as opções Pessoa e Projeto deverá ser exibida



Etapas de construção da aplicação web com acesso a banco de dados

- 1 - Desenvolver o modelo de dados da aplicação.
 - Diagrama Entidade-Relacionamento – Modelo lógico
 - Construção do modelo físico – Tabelas do Banco de dados e sua implementação no SGBD (Sistema Gerenciador de Banco de Dados)
- 2 - Desenvolver a interface (view). Nesse caso serão JSP's. Já foi feito em atividade anterior



Etapa 1 - Script para criação das tabelas no SGBD MySQL

- Criar o database
 - create database if not exists projeto;
- Acessar o database
 - use projeto;

Etapa 1 - Script para criação das tabelas no SGBD MySQL

- Criar a tabela de projetos

```
CREATE TABLE `projetos` (  
  `idProjetos` int(11) NOT NULL,  
  `nmDescricao` varchar(100) NOT NULL,  
  `dtInicio` date DEFAULT NULL,  
  `dtFinal` date DEFAULT NULL,  
  `nrPercConcluido` int(11) DEFAULT  
    NULL,  
  `nmSituacao` varchar(50) DEFAULT  
    NULL  
) ENGINE=InnoDB DEFAULT  
  CHARSET=utf8;
```

Etapa 1 - Script para criação das tabelas no SGBD MySQL

- Criar a tabela de participantes

```
CREATE TABLE `participantes` (  
  `idParticipantes` int(11) NOT  
  NULL,  
  `nmParticipante` varchar(50)  
  NOT NULL,  
  `nmCargo` varchar(50) DEFAULT  
  NULL  
) ENGINE=InnoDB DEFAULT  
  CHARSET=utf8;
```

Etapa 1 - Script para criação das tabelas no SGBD MySQL

- Criar a tabela de controle de projetos (participantes e seus projetos)

```
CREATE TABLE `controles` (  
  `idProjetos` int(11) NOT  
  NULL,  
  `idParticipantes` int(11) NOT  
  NULL  
) ENGINE=InnoDB DEFAULT  
CHARSET=utf8;
```

Etapa 1 - Script para criação das tabelas no SGBD MySQL

- Criar a tabela de controle de projetos (participantes e seus projetos)

```
ALTER TABLE `controles`  
  ADD PRIMARY KEY (`idProjetos`,`idParticipantes`),  
  ADD KEY `participantes_fk_idx` (`idParticipantes`);
```

```
ALTER TABLE `participantes`  
  ADD PRIMARY KEY (`idParticipantes`);
```

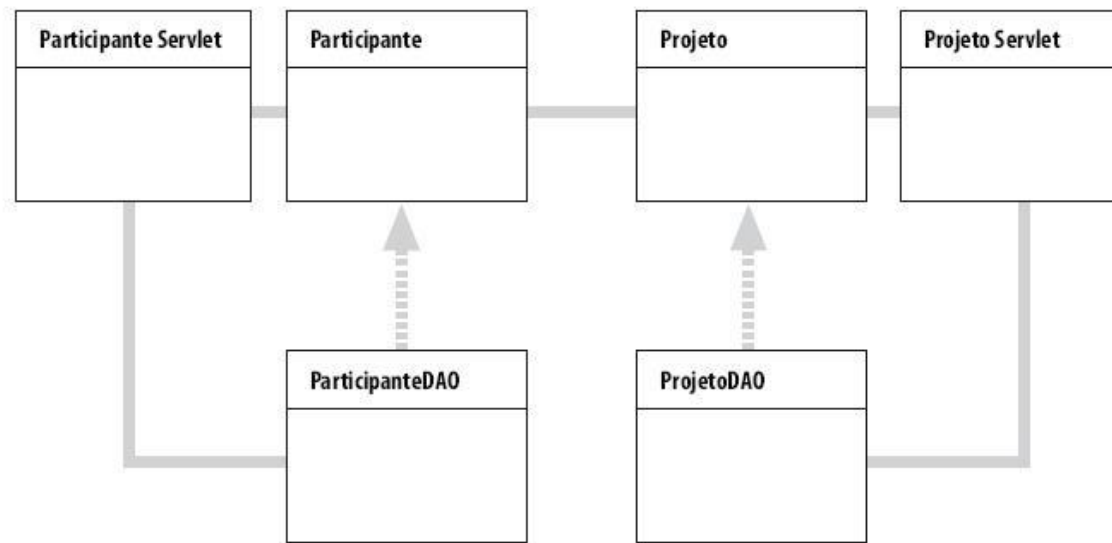
```
ALTER TABLE `projetos`  
  ADD PRIMARY KEY (`idProjetos`);
```

```
ALTER TABLE `participantes`  
  MODIFY `idParticipantes` int(11) NOT NULL AUTO_INCREMENT;
```

```
ALTER TABLE `projetos`  
  MODIFY `idProjetos` int(11) NOT NULL AUTO_INCREMENT;
```

```
ALTER TABLE `controles`  
  ADD CONSTRAINT `participantes_fk` FOREIGN KEY (`idParticipantes`) REFERENCES  
    `participantes` (`idParticipantes`) ON DELETE NO ACTION ON UPDATE NO ACTION,  
  ADD CONSTRAINT `projetos_fk` FOREIGN KEY (`idProjetos`) REFERENCES `projetos`  
    (`idProjetos`) ON DELETE NO ACTION ON UPDATE NO ACTION;  
COMMIT;
```

DAO - Data Access Object



- Padrão que isola as classes de negócio das classes que vão executar comandos SQL em alguma base de dados.

DAO Explicado

- As interfaces `ParticipanteDAO` e `ProjetoDAO` conterão todos os métodos abstratos que obrigatoriamente devem ser implementados pelas respectivas classes.
 - Por que isso? Dessa forma é possível ter várias implementações que acessarão diferentes bancos de dados, porém respeitando os métodos estabelecidos pela interface. Uma interface é uma espécie de contrato que deve ser respeitado.
 - Assim, é possível diminuir o nível de amarração da aplicação a um determinado banco de dados, ou seja, reduzir o acoplamento.
- As classes `ParticipanteDAO` e `ProjetoDAO` conterão métodos para serem utilizados no banco de dados: inserir, apagar, atualizar, recuperar.
- Criamos também dois Servlets (controller) que serão responsáveis por recuperar as informações fornecidas pelos usuários e preencher os objetos `Participante` e `Projeto` (model). Eles também enviarão as respostas para as views.

Atividade

- Utilizando a atividade de logar e cadastrar o aluno, acrescente um repositório de dados para persistir os usuários e os alunos, assim como a disciplina
- Podem usar JSP, JavaBen, Servlet