

Disciplina Regular 3

# Desenvolvimento Web com Java EE

Graduação em Engenharia de Software - 2020

# Etapa 1 Aula 2

## Servlets

Para acompanhar pelo Moodle você  
deve estudar as etapas 1 e 2

## Aula 2 - Competências e Checklist

- **Competências** Trabalhadas Nesta Etapa
  - Construir aplicativos Java utilizando Servlets.
  - Escrever interfaces simples com HTML e CSS.
- **Checklist** Desta Aula:
  - Montar o Ambiente de Desenvolvimento:
    - JDK 8.
    - Netbeans 11.
    - Tomcat 8.5.x.
  - Revisar os slides e fazer os exercícios da aula 1.
  - Ler os capítulos 1 e 2 do livro "Head First Servlets and JSP, 2nd Edition" disponível na O'Reilly Safari.

# Parâmetros de Requisição

# Parâmetros de Requisição

- Um servlet pode receber parâmetros vindos de um formulário HTML ou da inserção direta desses parâmetros em um link.
- São passados no formato nome=valor

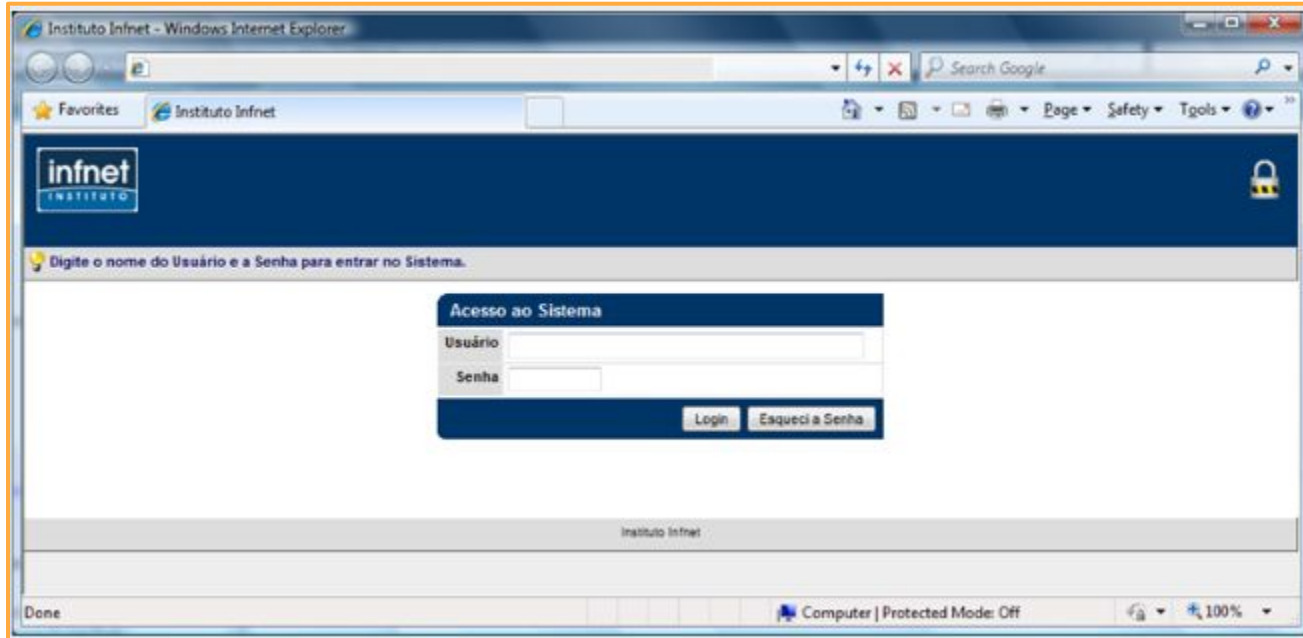
`produto=1201`

- E são separados por &

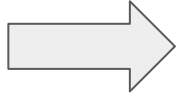
`produto=1201&quantidade=5`

# Formulários HTML

- Formulários HTML são usados para captar dados digitados pelos usuários e para enviar esses dados ao servidor.



# Requisição com Método GET X POST



**GET /servlet/com.infnet.exemplo?produto=1201 HTTP/1.1**

Accept: image/gif, image/jpeg, application/pdf

Accept-Language: pt

Accept-Charset: iso-8859-1

Connection: keep-alive

Host: localhost:80

Referer: http://www.infnet.com.br/teste.html

User-Agent: Mozilla/4.7 [en]

**Os dados digitados no formulário aparecem no endereço da requisição.**

# Requisição com Método GET X POST

## POST /app HTTP/1.1

Accept: image/gif, image/jpeg, application/pdf

Accept-Language: pt

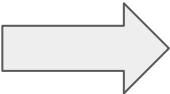
Accept-Charset: iso-8859-1

Connection: keep-alive

Host: localhost:80/

Referer: http://localhost:80/aluguel/cadastro.html

User-Agent: Mozilla/4.7 [en]



**<http://www.infnet.com.br/app?produto=1201&quantidade=5>**

Os dados digitados no formulário são transmitidos no corpo da requisição.



# Requisição com Método GET X POST

- Use GET:
  - Quando a quantidade de dados for pequena.
  - Quando os dados puderem ser exibidos na URL.
  - Quando desejar passar dados pela URL (link).
- Use POST:
  - Sempre que fizer um formulário.

# Processamento de Parâmetros

- Para recuperar os parâmetros são usados os métodos do objeto request:



**Atenção ao uso do método `getParameter`: ele retorna “String” ou “null” e deve ser convertido quando for necessário.**

# Exemplo

- Antes de consultar o saldo, o cliente deve efetuar o login entrando com os dados no formulário.

```
String login = request.getParameter("txtLogin");  
String senha = request.getParameter("txtSenha");
```

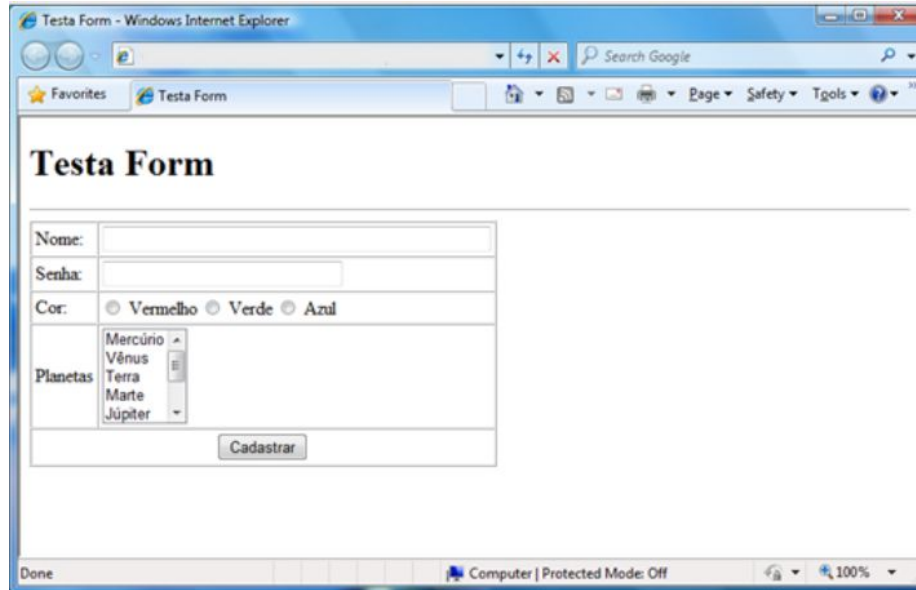
```
if (login == null || login.equals("") ||  
    senha == null || senha.equals(""))  
    gerarVisaoErroValidacao(request, response);
```

```
if (senha.equals("password"))  
    gerarVisaoMenu(request, response);
```

```
else  
    gerarVisaoAcessoNegado(request, response);
```

# Exercício

- Construir uma aplicação simples com o formulário:



The screenshot shows a Windows Internet Explorer browser window titled 'Testa Form - Windows Internet Explorer'. The address bar is empty, and the search bar contains 'Search Google'. The page title is 'Testa Form'. The form contains the following fields:

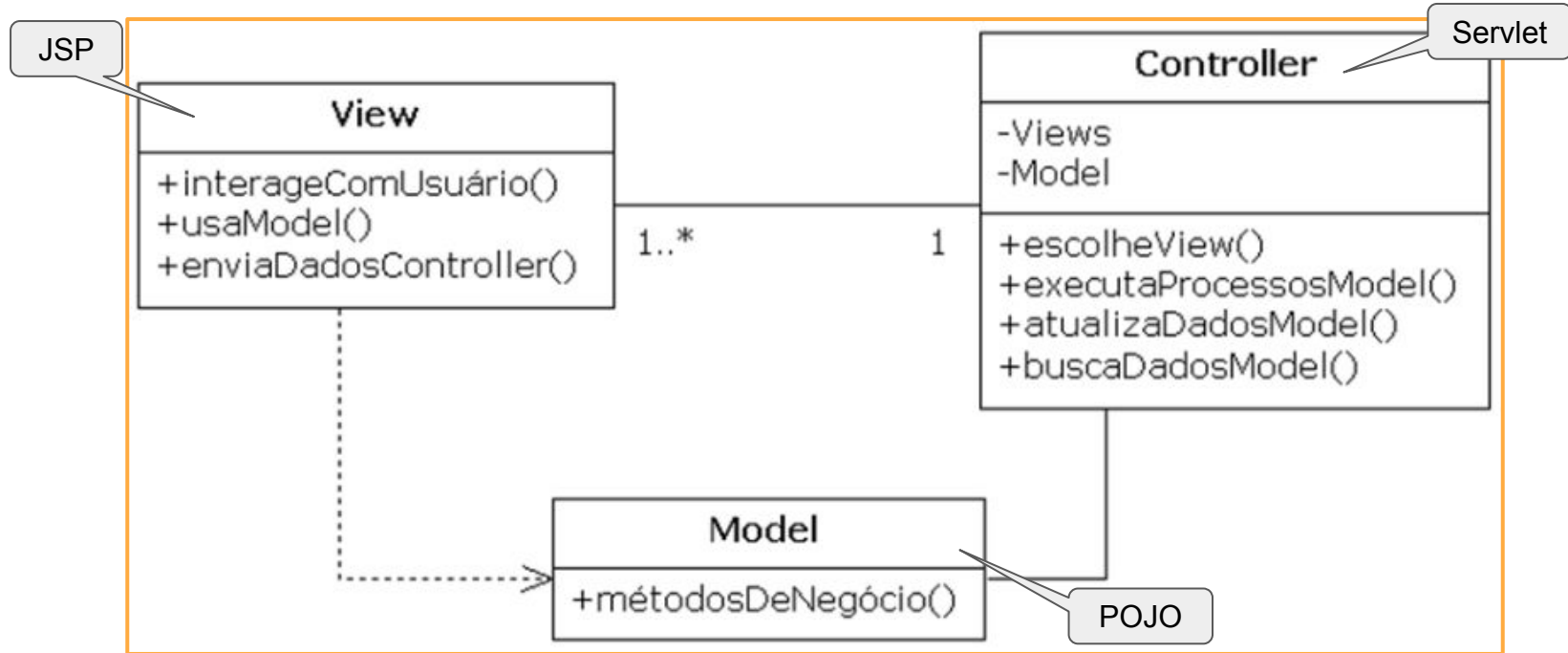
- Nome:
- Senha:
- Cor: ☐ Vermelho ☐ Verde ☐ Azul
- Planetas:
  - Mercúrio
  - Vênus
  - Terra
  - Marte
  - Júpiter

A 'Cadastrar' button is located at the bottom of the form.

- Construir um servlet que obtenha os dados passados pelo formulário e exiba o seu conteúdo.

# Servlet Controller

# Arquitetura MVC



**MVC é um Design Pattern de arquitetura.**

# Papel do Servlet no padrão MVC

- Funções:
  1. Recuperação dos parâmetros.
  2. Validação dos parâmetros.
  - 3. Processamento da requisição.**
  4. Inserção dos JavaBeans (POJO) resultantes do processamento na requisição.
  5. Redirecionamento da requisição para o JSP que apresentará o resultado.
- O Servlet usado dessa forma é chamado de **Servlet Controller**.

# Redirecionamento de Fluxo

- O container pode transferir o controle de um servlet para outro componente.
- Objetos podem ser inseridos no objeto de requisição e recuperados no destino.
- Essa operação é feita com um objeto da classe **RequestDispatcher** que recuperado a partir da requisição.
- Para repassar o controle para outro servlet, usa-se o método **forward**.



# Exemplo

- Validação de senha:

```
RequestDispatcher rd =  
    request.getRequestDispatcher("index.jsp");  
if (senha.equals("password")) {  
  
    ContaCorrente cc = new ContaCorrente(login);  
    request.setAttribute("conta", cc);  
} else {  
  
    request.setAttribute("msgErro", "Senha Inválida");  
}  
rd.forward(request, response);
```

# Exercício

- Construir uma aplicação simples com um formulário da entidade Contato, com nome, email e fone.
- Construir um servlet que obtenha os dados passados pelo formulário, valide os dados conforme as regras abaixo e redirecione de volta para o formulário:
  - nome → obrigatório.
  - email → obrigatório.
  - fone → obrigatório, numérico.
- Quando redirecionar, exibir uma mensagem de “Sucesso” ou uma lista de mensagens de erro.