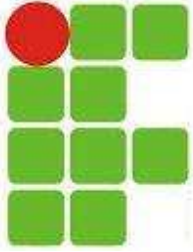

Programação Web

Professor: Diego Oliveira



**Conteúdo 14:
Model, View and Controller**

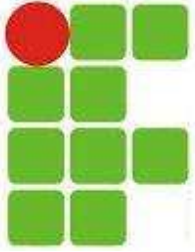




MVC

- MVC (Model-View-Controller) é um padrão de projeto de software
- Um padrão de projeto é uma solução para um problema dentro de um contexto
- Ao lidar com um cliente observa-se que os requisitos do software mudam constantemente
- Desta maneira separar a lógica de negócios da apresentação facilita a manutenção e modificação do sistema

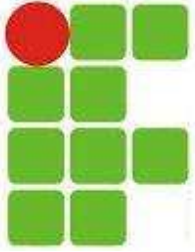




MVC

- **Modelo:**
 - Representa a lógica do negócio
 - Armazena as entidades do sistema
 - Pode interagir com a base de dados
- **Visão:**
 - Responsável por exibir as informações ao usuário
- **Controle:**
 - Interliga a Visão e o Modelo





MVC

- JSP e Servlets com MVC:
 - MVC retira a lógica de negócios e a interface de dentro dos Servlets
 - A lógica de negócios é colocada em classes simples e se torna a camada Modelo
 - A interface é colocada dentro dos JSP e se torna a camada de Visão
 - O que sobra nos Servlets se torna a camada de controle do sistema





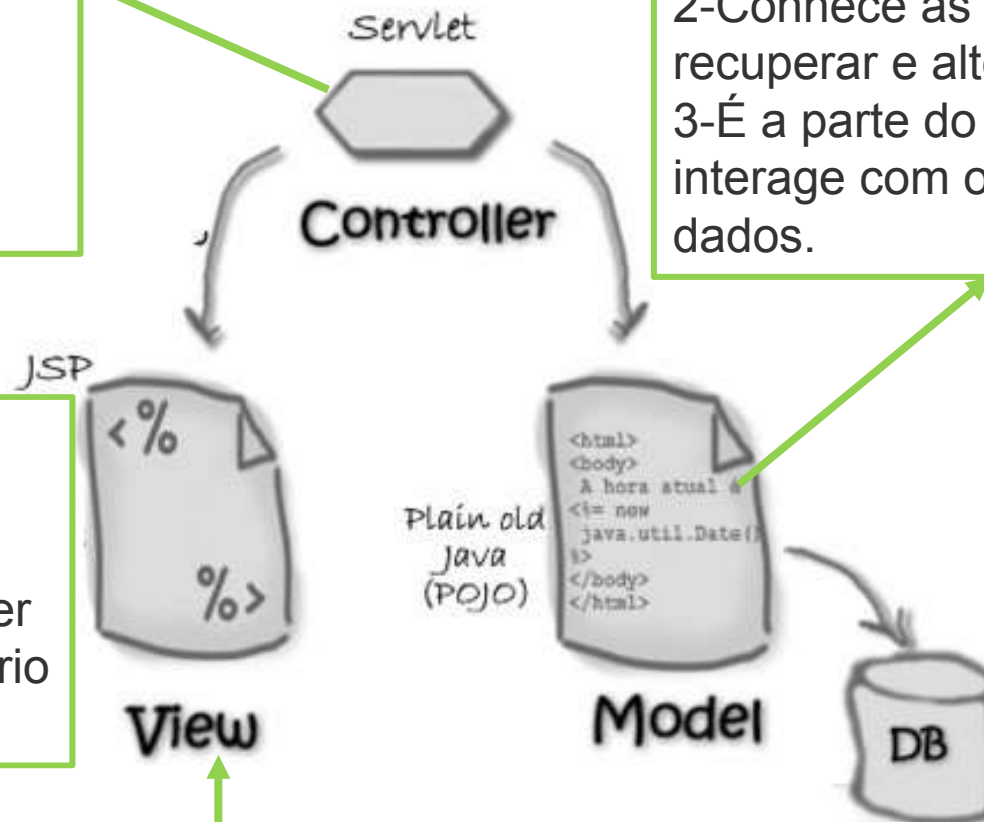
MVC

- **Resumo do JSP com MVC:**

1-Recupera a entrada do usuário a partir da requisição e traduz para o modelo.
2-Diz ao modelo para se atualizar e disponibiliza um novo estado para a visão.

1-Responsável pela apresentação.
2-Recupera o estado e o modelo através do controller
3-Obtém entradas do usuário e repassa para o controller

1-Guarda a lógica e o estado do negócio.
2-Conhece as regras para recuperar e alterar o estado.
3-É a parte do sistema que interage com o banco de dados.



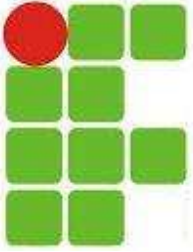


Exemplo de JSP com MVC

- Vamos criar um sistema utilizando MVC que recebe o login de um usuário e mostra um menu, como na aula anterior.
- Pra isso faremos uma página JSP, um Servlet e utilizaremos o banco de dados + a classe Usuário.
- Ao final o Servlet escreverá uma nova JSP com a resposta, que será o menu de acordo com o nível de acesso do usuário logado.



- Vamos aos códigos!



MODEL

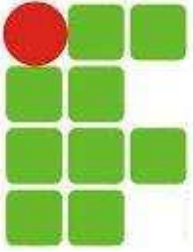
- Método de recuperação do Usuário pelo 'login' no Banco de Dados:

```
public Usuario getUsuario(String login) throws SQLException{
    ResultSet rs;
    String SQL = "select * from usuario where login = '" +
                login + "'";
    Usuario user = new Usuario();
    rs = stmt.executeQuery(SQL);
    while (rs.next()){
        user.setLogin(login);
        user.setId(rs.getInt("id"));
        user.setSenha(rs.getString("senha"));
        user.setNivelAcesso(rs.getInt("nivelAcesso"));
    }
    return user;
}
```

```
public class Usuario {
    private int id;
    private String login;
    private String senha;
    private int nivelAcesso;
```

```
//GETs e SETs
```





VIEW

- JSP inicial (tela de login):

```
<body>
  
  <%
    session.setAttribute("login", null);
    session.setAttribute("id", null);
  %>
  <h1>Sistema de Empréstimo de Livros</h1>
  <form action="VerificacaoUsuarios" method="processRequest">
    Login: <input type="text" name="login"><br><br>
    Senha: <input type="password" name="senha"><br><br>
    <input type="submit" value="LOGIN">
  </form>
  <br><a href="CadastrarUsuario.jsp">Não sou cadastrado ainda!</a>
</body>
```





CONTROLLER

- Servlet:

```
@WebServlet(name = "VerificacaoUsuarios", urlPatterns = {"/VerificacaoUsuarios"})
public class VerificacaoUsuarios extends HttpServlet {
    protected void processRequest(HttpServletRequest request,
        HttpServletResponse response)
        throws ServletException, IOException, SQLException {
        response.setContentType("text/html;charset=UTF-8");

        //1-RECUPERAR DADOS DO JSP (VIEW)
        String login = request.getParameter("login");
        String senha = request.getParameter("senha");

        //2-ACESSAR O MODELO PARA VERIFICAR OS DADOS DO USUÁRIO
        Banco b = new Banco();
        Usuario user = b.getUsuario(login);
    }
}
```





CONTROLLER

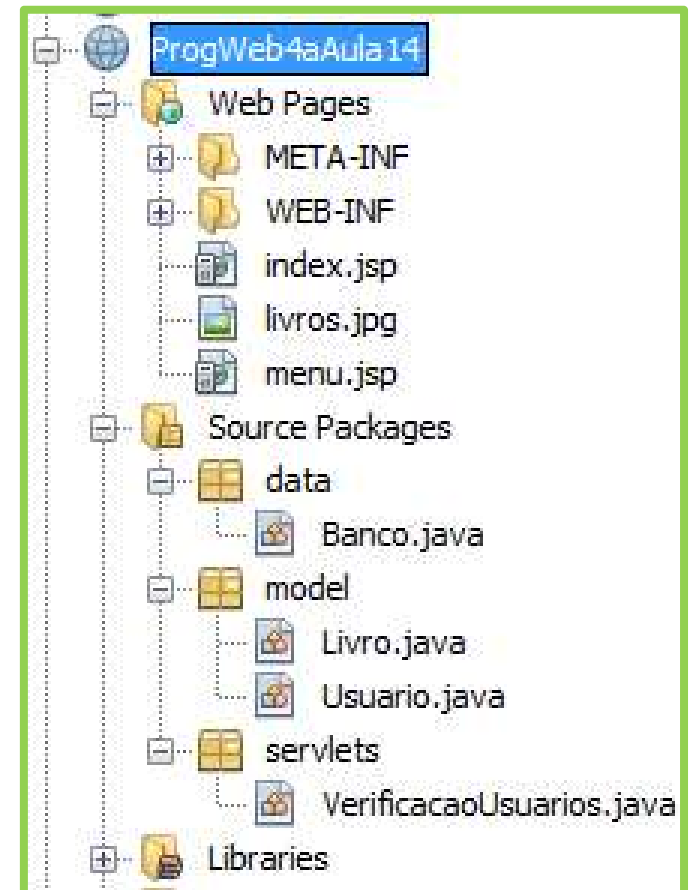
- Servlet:

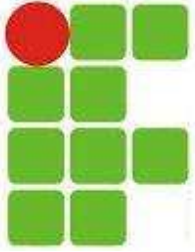
```
//3-ESCREVER A RESPOSTA UTILIZANDO O PRÓPRIO CONTROLLER (ESTE SERVLET)|
try (PrintWriter out = response.getWriter()) {
    out.println("<!DOCTYPE html>");
    out.println("<html>");
    out.println("<body>");
    if (!senha.equals(user.getSenha())) {
        out.println("<h1>SENHA ou LOGIN incorretos!</h1>");
    } else {
        out.println("<table border='1'>");
        out.println("<tr><td><ul>");
        if (user.getNivelAcesso() >= 1) {
            out.println("<li><a href='ConsultarLivro.jsp' >Consultar Livro</a></li>");
        }
        if (user.getNivelAcesso() >= 2) {
            out.println("<li><a href='EmprestarLivro.jsp' >Pegar Livro</a></li>");
            out.println("<li><a href='DevolverLivro.jsp' >Devolver Livro</a></li>");
            out.println("<li><a href='VerEmprestimosLivro.jsp' >Ver Empréstimos</a></li>");
        }
        if (user.getNivelAcesso() == 3) {
            out.println("<li><a href='AtualizarLivro.jsp' >Atualizar Livro</a></li>");
            out.println("<li><a href='CadastrarLivro.jsp' >Cadastrar Livro</a></li>");
            out.println("<li><a href='RemoverLivro.jsp' >Remover Livro</a></li>");
        }
    }
}
```



Exercício

- A estrutura do exemplo da aula ficou como a do lado:
- Baseando-se no exemplo da aula, faça um sistema onde é digitado o login do usuário e de acordo com o nível de acesso dele mostre uma página com título, cor de fundo e links diferentes





Perguntas?

