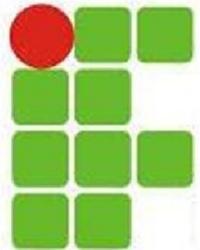
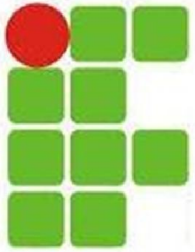

Programação Web

Professor: Diego Oliveira



**Conteúdo 11:
Acesso a BD**



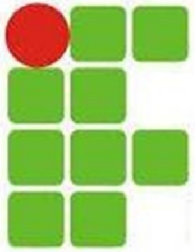


Acesso a Banco de Dados

- O acesso a banco de dados via JSP se dá da mesma maneira que o acesso de uma aplicação desktop
- Utilizaremos JDBC para fazer a conexão com banco MySQL
- A configuração do banco pode ser feita utilizando-se o MySQL Workbench, porém recomendo o HeidiSQL por ser muito mais simples e leve:

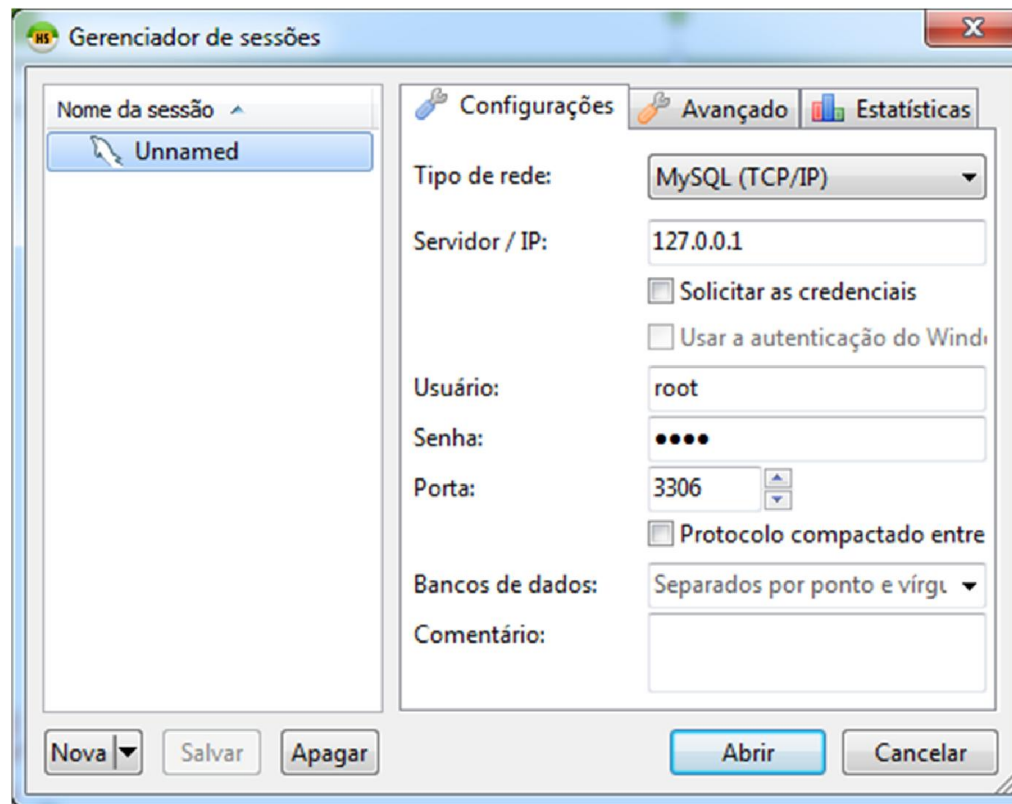


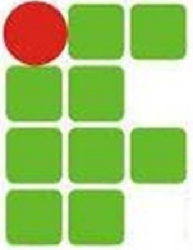
<http://www.heidisql.com/>



GUI do HeidiSQL

- Conectando a um banco de dados:





GUI do HeidiSQL

- Criando o banco manualmente:

The screenshot shows the HeidiSQL interface with the following details:

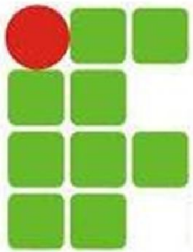
- Window title: Unnamed\livros\livro\ - HeidiSQL Portable 8.3.0.4694
- Menu bar: Arquivo, Editar, Pesquisar, Ferramentas, Ajuda
- Toolbar: Standard database management icons.
- Left sidebar: Tree view showing database structure: Unnamed > information_schema, estoque, livros (16,0 KiB) > livro (16,0 KiB), memo, mysql, performance_schema, sgl (112,0 KiB), sgl2, test.
- Right pane: Server: 127.0.0.1, Banco de dados: livros, Tabela: livro. Tabs: Básico, Opções, Índices, Chaves-estrangeiras.
- Form fields: Nome: livro, Comentário: (empty).
- Columns table:

#	Nome	Tipo de dados	Tamanho/Itens
1	id	VARCHAR	10
2	titulo	VARCHAR	50
3	autor	VARCHAR	50
4	edicao	VARCHAR	11
- Buttons: Ajuda, Descartar, Salvar.
- SQL Editor:

```
76 UPDATE `livros`.`livro` SET `codigo`='6552' WHERE `id`='5' AND `titulo`='Principios de Analise e Projeto'
```

```
77 SELECT `id` `titulo` `autor` `edicao` `editora` `ano` `codigo` `exemplares` FROM `livros`.`livro` W
```
- Status bar: Connected: 00:05 h, MySQL 5.6.20, Ativo durante: 00:13 h, Ocupado.





GUI do HeidiSQL

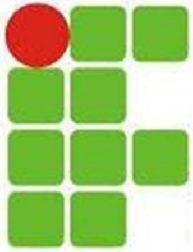
- Consultando informações via SQL:

The screenshot shows the HeidiSQL interface with the following details:

- Window Title:** Unnamed\livros\livro - HeidiSQL Portable 8.3.0.4694
- Menu Bar:** Arquivo, Editar, Pesquisar, Ferramentas, Ajuda
- Toolbar:** Standard database management icons.
- Left Panel (Database Tree):** Shows a tree view with folders like 'information_schema', 'estoque', 'livros' (16,0 KiB), 'memo', 'mysql', 'performance_schema', 'sgl' (112,0 KiB), 'sgl2', and 'test'.
- Right Panel (Query Editor):** Shows the server '127.0.0.1', database 'livros', and table 'livro'. The query entered is: `1 select * from livro where id < 10`
- Table View:** Displays the results of the query in a table with columns 'id', 'titulo', and 'autor'.

id	titulo	autor
1	Concepcao de Circuito Integrado	REIS, Ricardo Augusto
2	Sistemas de Software em Tempo Real	SHAW, Alan
3	Computacao Grafica: Teoria e Pratica	AZEVEDO, Eduardo
4	Algoritmos Estruturados	BACKER, Christiano Goncalv
5	Principios de Analise e Projeto	BEZERRA, Eduardo
6	Grafos: Teoria, Modelos e Algoritmos	BOAVENTURA NETO, Paulo
7	UML: Guia do Usuario: O Mais Avancado	BOCH, Grady
8	Multi-agent Programming: Languages	BORDINI, Rafael H
9	Mobile Agents: Basic Concepts, Mobility	BRAUN, Peter
- Status Bar:** Shows '81 /* Affected rows: 0 Registros encontrados: 9 Avisos: 0 Duração de 1 query: 0,000 sec. */', connection status 'Connected: 00:06 h', MySQL version 'MySQL 5.6.20', and 'Ativo durante: 00:14 h'.





GUI do HeidiSQL

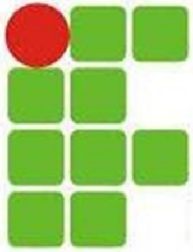
- Alterando dados no banco manualmente:

The screenshot shows the HeidiSQL interface with the following details:

- Window title: Unnamed\livros\livro\ - HeidiSQL Portable 8.3.0.4694
- Menu bar: Arquivo, Editar, Pesquisar, Ferramentas, Ajuda
- Toolbar: Standard database management icons.
- Left sidebar: Tree view showing database structure. The 'livros' database is selected, and the 'livro' table is highlighted.
- Right pane: Table view for 'Tabela: livro'. It shows a list of books with columns 'id', 'titulo', and 'autor'. Row 6 is selected.
- Status bar: 81 /* Affected rows: 0 Registros encontrados: 9 Avisos: 0 Duração de 1 query: 0,000 sec. */
- Bottom bar: Connected: 00:07 h, MySQL 5.6.20, Ativo durante: 00:15 h, Ocupado.

id	titulo	autor
1	Concepcao de Circuito Integrado	REIS, Ricardo A
2	Sistemas de Software em Tempo Real	SHAW, Alan
3	Computacao Grafica: Teoria e Pratica	AZEVEDO, Edua
4	Algoritmos Estruturados	BACKER, Christi
5	Principios de Analise e Projeto	BEZERRA, Edua
6	Grafos: Teoria, Modelos e Algoritmos	BOAVENTURA N
7	UML: Guia do Usuario: O Mais Avancado	BOCH, Grady
8	Multi-agent Programming: Languages	BORDINI, Rafea
9	Mobile Agents: Basic Concepts, Mobility	BRAUN, Peter
10	Ciencia da Computacao: uma visao	BROOKSHEAR, .
11	Interligacao em Rede com TCP/IP VOL1	COMER, Dougla
12	Interligacao em Rede com TCP/IP VOL2	COMER, Dougla



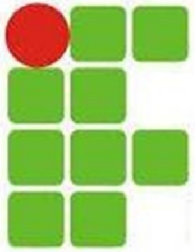


Conexão JDBC

- Após a configuração do banco de dados, é preciso estabelecer uma conexão do programa com ele, para isso utilizaremos JDBC
- Utilizaremos o MySQL Server 5.6
- Para conectar via JDBC, é preciso colocar a biblioteca do MySQL como dependência no projeto. Utilizaremos o seguinte JAR:

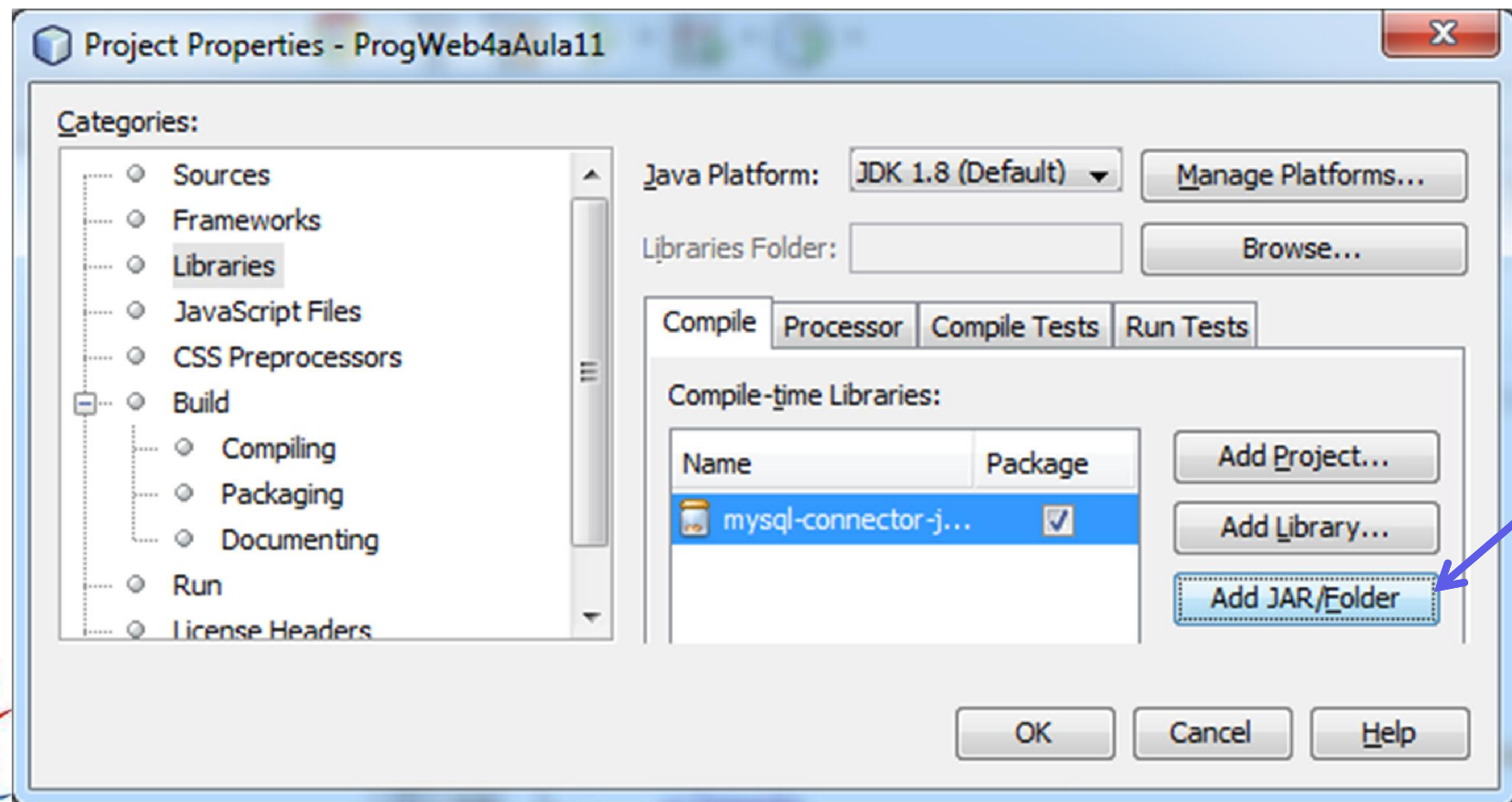


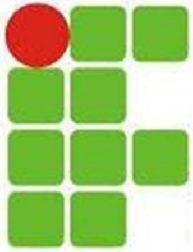
mysql-connector-java-5.1.32-bin.jar



Adicionando JAR ao Projeto

- Botão direito no projeto, PROPRIEDADES:

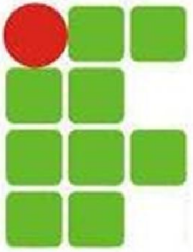




Conectando com o Banco

- Configuração da conexão para MySQL:

```
8 public class Banco {
9     private Statement stmt;
10    private ResultSet rs;
11    public Connection conn;
12    private static Banco b = new Banco();
13    public Banco() {
14        String url = "jdbc:mysql://localhost:3306/biblioteca";
15        String usr = "root";
16        String pas = "root";
17        try {
18            Class.forName("com.mysql.jdbc.Driver");
19            conn = DriverManager.getConnection(url, usr, pas);
20            stmt = conn.createStatement(ResultSet.TYPE_SCROLL_SENSITIVE,
21                                     ResultSet.CONCUR_READ_ONLY);
22        } catch (Exception e) {
23            System.out.println("Erro" + e.getMessage());
24        }
25    }
}
```

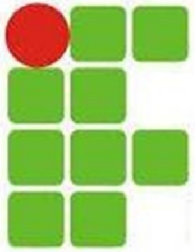


Conectando com o Banco

- Configuração da conexão para PostgreSQL:

```
9 public class Banco {
10     private Statement stmt;
11     private ResultSet rs;
12     public Connection conn;
13     private static Banco b = new Banco();
14     public Banco() {
15         String url = "jdbc:postgresql://localhost:5432/biblioteca";
16         String usr = "postgres";
17         String pas = "postgres";
18         try {
19             Class.forName("org.postgresql.Driver");
20             conn = DriverManager.getConnection(url, usr, pas);
21             stmt = conn.createStatement(ResultSet.TYPE_SCROLL_SENSITIVE,
22                                     ResultSet.CONCUR_READ_ONLY);
23         } catch (Exception e) {
24             System.out.println("Erro" + e.getMessage());
25         }
26     }
}
```

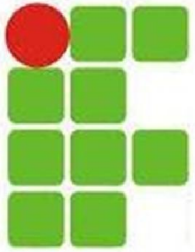




Integrando JSP e MySQL

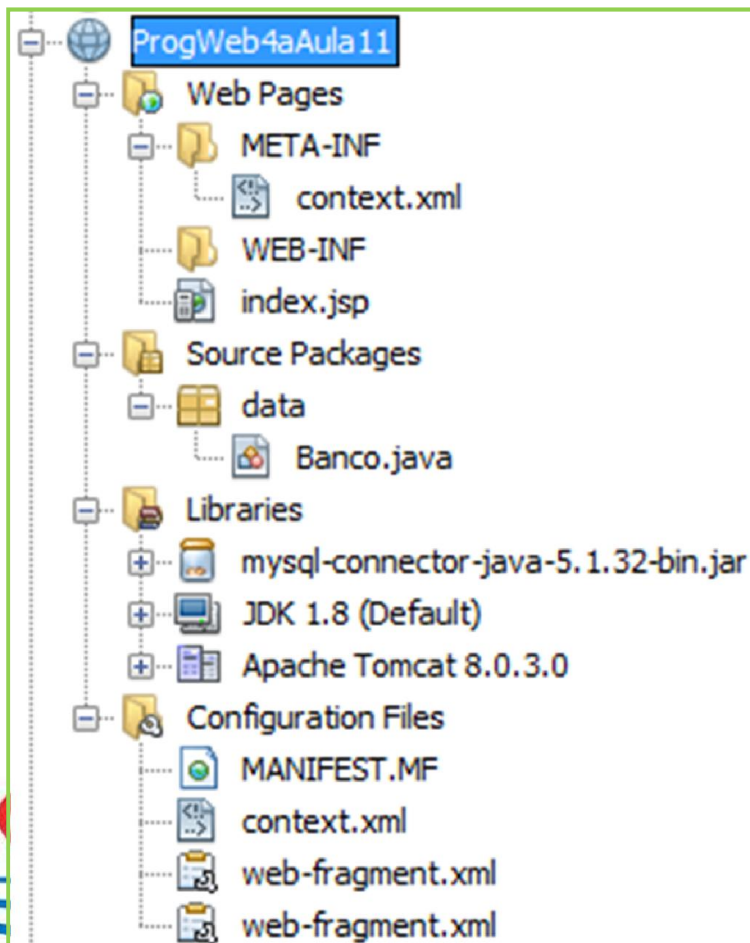
- Agora que o projeto já está conectado com o banco de dados, resta executar os comandos SQL desejados
- Os comandos de consulta retornarão valores, que serão armazenados em um ResultSet
- Este ResultSet pode ser percorrido para jogar as informações diretamente na página JSP ou montar relatórios em PDF



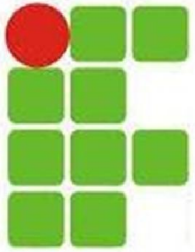


Integrando JSP e MySQL

- Atenção à estrutura do projeto!



- 1-As páginas JSP devem estar dentro de WEB PAGES
- 2-O JAR do MySQL Connector deve aparecer na pasta LIBRARIES
- 3-A página principal deve se chamar index.jsp, caso contrário nada aparecerá
- 4-Classes que serão usadas nos JSP devem ser importadas, obrigatoriamente
- 5-Não é recomendável fazer consultas nem código extenso dentro dos JSP, uma classe deve ser criada dentro de um pacote e ser utilizada no JSP através de importação
- 6-Objetos utilizados ao longo de todo o projeto devem ser criados uma única vez, utilizando o padrão SINGLETON
- 7-Sempre feche as conexões ao encerrar



Integrando JSP e MySQL

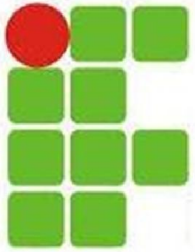
- Lendo todos os livros e jogando no JSP:

```
7 <%@page import="java.sql.ResultSet"%>
8 <%@page import="data.Banco"%>
9 <%@page contentType="text/html" pageEncoding="ISO-8859-1"%>
10 <!DOCTYPE html>
11 <html>
12 <head>
13 <meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
14 <title>JSP + MySQL</title>
15 <%
16 Banco b = new Banco();
17 ResultSet livros = b.selectAllBooks();
18 %>
19 </head>
```



```
26 public ResultSet selectAllBooks() throws SQLException{
27     ResultSet rs = stmt.executeQuery("select * from livro");
28     return rs;
29 }
```

Banco.java

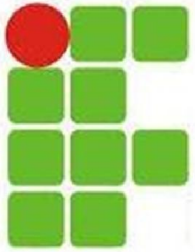


Integrando JSP e MySQL

- Lendo todos os livros e jogando no JSP:

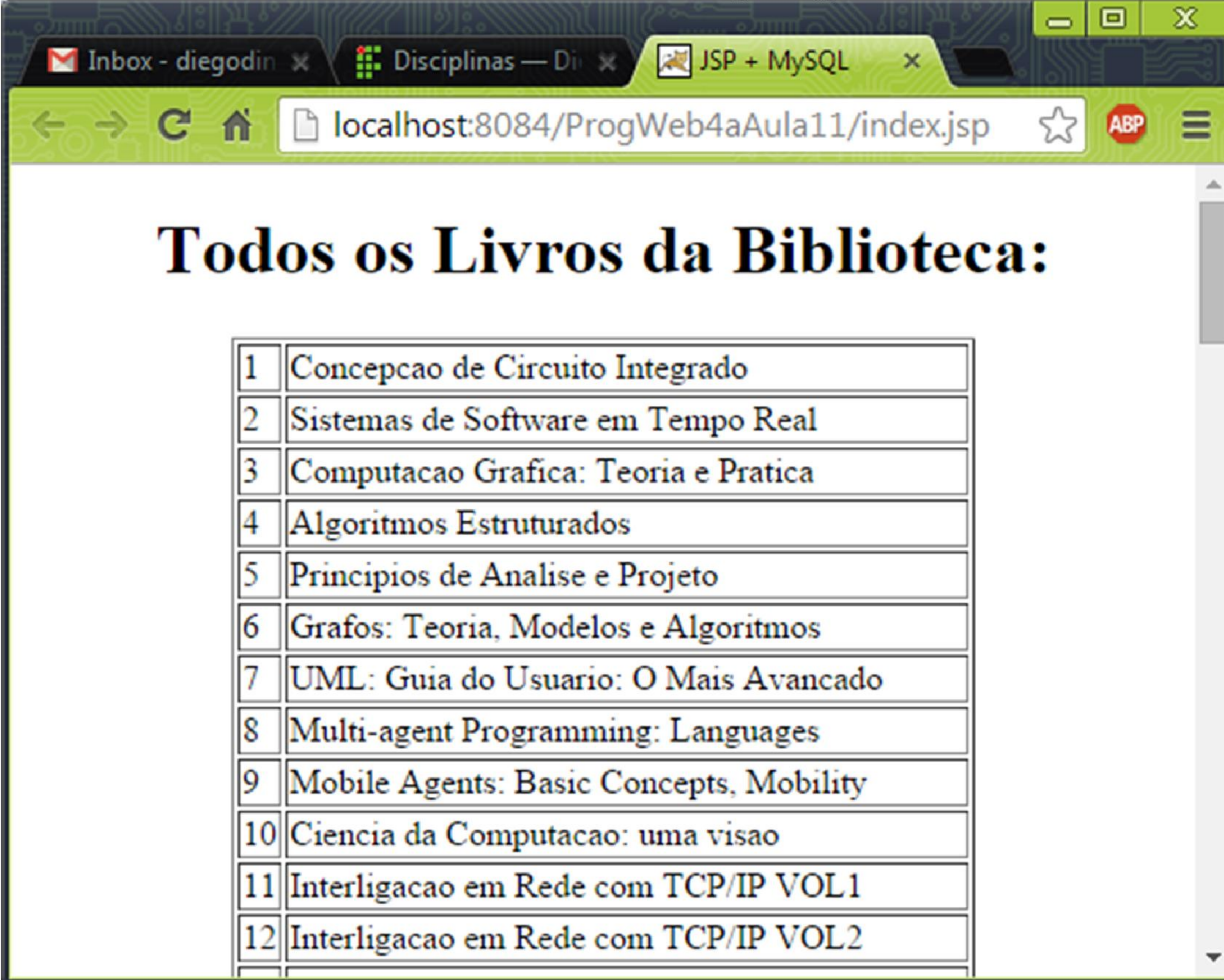
```
20 <body>
21 <center>
22 <h1>Todos os Livros da Biblioteca:</h1>
23 <table border="1">
24 <%
25     while(livros.next()){
26 <%>
27 <tr>
28 <td> <%=livros.getString("id")%> </td>
29 <td> <%=livros.getString("titulo")%> </td>
30 </tr>
31 <%=}%>
32 </table>
33 </center>
34 </body>
```





Integrando JSP e MySQL

- Lendo todos os livros e jogando no JSP:

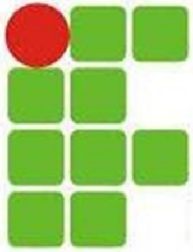


The screenshot shows a web browser window with the following details:

- Address bar: localhost:8084/ProgWeb4aAula11/index.jsp
- Page Title: Todos os Livros da Biblioteca:
- Table content:

1	Concepcao de Circuito Integrado
2	Sistemas de Software em Tempo Real
3	Computacao Grafica: Teoria e Pratica
4	Algoritmos Estruturados
5	Principios de Analise e Projeto
6	Grafos: Teoria, Modelos e Algoritmos
7	UML: Guia do Usuario: O Mais Avancado
8	Multi-agent Programming: Languages
9	Mobile Agents: Basic Concepts, Mobility
10	Ciencia da Computacao: uma visao
11	Interligacao em Rede com TCP/IP VOL1
12	Interligacao em Rede com TCP/IP VOL2

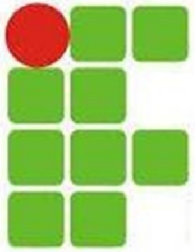




Exercício

- Na área do professor baixe os arquivos:
 - **livros.sql e mysql-connector-java-5.1.32-bin.jar**
- Crie um projeto WEB
- Crie o banco de dados “biblioteca”
- Conecte seu projeto ao Banco
- Realize as seguintes consultas, mostrando o resultado em uma página JSP:
 - **Todos os livros que contenham “java” no título**
 - **Todos os livros da editora “Campus”**
 - **Todos os livros com ano inferior a 2000**
 - **Todos os livros com mais de 3 exemplares**





Perguntas?

