**Matéria** : Tópicos especiais em Sistemas de Informações

**Alunos**:  
Ana Laura Rocha Gonçalves – R.A 010624042  
Ana Paula Silva Morais – R.A 010624061

Tema: Locação de Vestidos

**CÓDIGO COMENTADO:**

// Declara o pacote onde esta classe está localizada

package com.gerenciarvestidos;

// Importa as anotações para web service

import javax.jws.WebService;

import javax.jws.WebMethod;

import javax.jws.WebParam;

// Importa classes para manipulação de banco de dados

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.SQLException;

// Importa classes para manipulação de listas (não usado neste código)

import java.util.ArrayList;

import java.util.List;

// Define que essa classe é um web service

@WebService(

name = "GerenciarVestido", // Nome do serviço

serviceName = "GerenciarVestidosService" // Nome do serviço exposto

)

public class GerenciarVestidos {

// URL para conexão com o banco de dados

private static final String URL = "jdbc:derby://localhost:1527/laura";

// Usuário para acesso ao banco de dados

private static final String USUARIO = "laura";

// Senha para acesso ao banco de dados

private static final String SENHA = "laura";

// Variável para armazenar a conexão com o banco

private Connection conexao;

// Define método web para verificar se um vestido está alugado

@WebMethod(operationName = "verificarEmprestimo")

public String verificarEmprestimo(

// Recebe o ID do vestido como parâmetro

@WebParam(name = "id") int id)

{

try {

// Abre a conexão com o banco usando URL, usuário e senha

conexao = DriverManager.getConnection(URL, USUARIO, SENHA);

// Define comando SQL para buscar se o vestido está alugado

String sql = "SELECT alugado FROM vestidos WHERE id = ?";

// Prepara o comando SQL para execução segura com parâmetro

PreparedStatement localizar = conexao.prepareStatement(sql);

// Define o valor do parâmetro "id" na consulta SQL

localizar.setInt(1, id);

// Executa a consulta e obtém os resultados

ResultSet resultado = localizar.executeQuery();

// Variável para armazenar status do aluguel, padrão "false"

String estaAlugado = "false";

// Se existir resultado, atualiza a variável com valor da coluna "alugado"

if (resultado.next()) {

estaAlugado = resultado.getString("alugado");

}

// Fecha o resultado para liberar recursos

resultado.close();

// Fecha o statement para liberar recursos

localizar.close();

// Fecha a conexão com o banco

conexao.close();

// Retorna "Augado" (provavelmente quis dizer "Alugado")

return "Alugado";

} catch (SQLException e) {

// Em caso de erro na consulta, imprime o erro no console

e.printStackTrace();

// Retorna mensagem indicando que não está alugado

return "Não alugado";

}

}

// Define método web para incluir um vestido no banco

@WebMethod(operationName = "incluirVestidos")

public boolean incluirVestido(

// Recebe os parâmetros do vestido a ser incluído

@WebParam(name = "id") int id,

@WebParam(name = "cor") String cor,

@WebParam(name = "tamanho") String tamanho,

@WebParam(name = "modelo") String modelo,

@WebParam(name = "alugado") boolean alugado) {

try {

// Abre conexão com o banco de dados

conexao = DriverManager.getConnection(URL, USUARIO, SENHA);

// Comando SQL para inserir um novo vestido

String sql = "INSERT INTO vestidos (id,cor,tamanho,modelo,alugado) VALUES (?, ?, ?, ?, ?)";

// Prepara o comando para execução com parâmetros

PreparedStatement incluir = conexao.prepareStatement(sql);

// Define o valor dos parâmetros da consulta SQL

incluir.setInt(1,id);

incluir.setString(2, cor);

incluir.setString(3, tamanho);

incluir.setString(4, modelo);

incluir.setBoolean(5, alugado);

// Executa a inserção e retorna o número de linhas afetadas

int linhasAfetadas = incluir.executeUpdate();

// Fecha o statement para liberar recursos

incluir.close();

// Fecha a conexão com o banco

conexao.close();

// Retorna true se pelo menos uma linha foi inserida, false caso contrário

return linhasAfetadas > 0;

} catch (SQLException e) {

// Em caso de erro, imprime o erro no console

e.printStackTrace();

// Retorna false indicando falha na inserção

return false;

}

}

// Define método web para alterar o status de aluguel de um vestido

@WebMethod(operationName = "alterarVestidos")

public boolean alterarVestido(

// Recebe o ID do vestido a ser alterado

@WebParam(name = "id") int id,

// Recebe o novo status de aluguel (como int, 0 ou 1 provavelmente)

@WebParam(name = "alugado") int alugado){

try {

// Abre conexão com o banco de dados

conexao = DriverManager.getConnection(URL, USUARIO, SENHA);

// Comando SQL para atualizar o status de aluguel do vestido com o ID informado

String sql = "UPDATE vestidos SET alugado=? WHERE id=?";

// Prepara o comando para execução com parâmetros

PreparedStatement alterar = conexao.prepareStatement(sql);

// Define os valores dos parâmetros

alterar.setInt(1,alugado);

alterar.setInt(2, id);

// Executa a atualização e retorna o número de linhas afetadas

int linhasAfetadas = alterar.executeUpdate();

// Fecha o statement para liberar recursos

alterar.close();

// Fecha a conexão com o banco

conexao.close();

// Retorna true se pelo menos uma linha foi atualizada

return linhasAfetadas > 0;

} catch (SQLException e) {

// Em caso de erro, imprime o erro no console

e.printStackTrace();

// Retorna false indicando falha na atualização

return false;

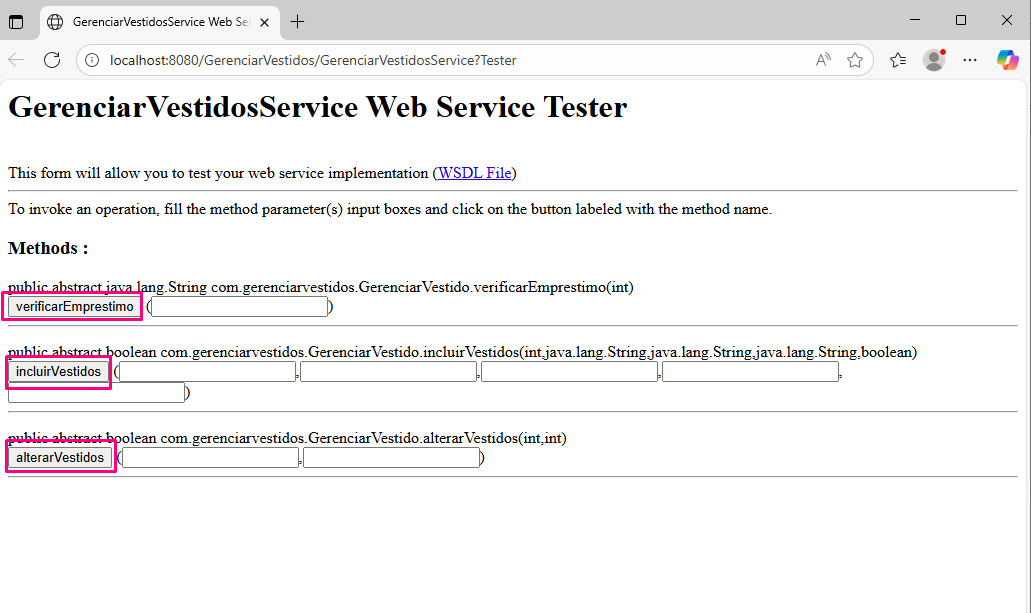
}

}

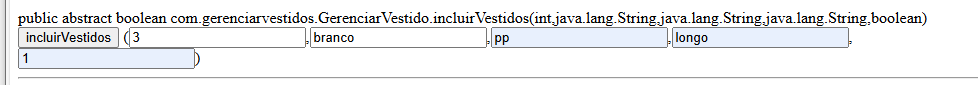
}

TELAS RODANDO

Página inicial:

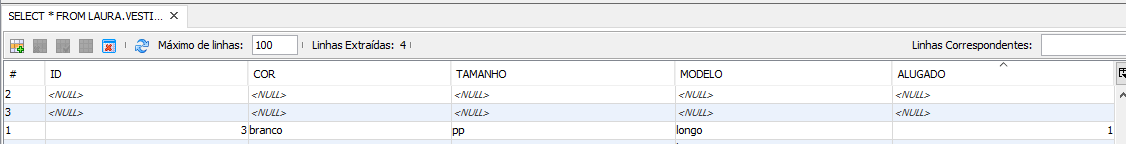


Incluindo:



Retorno em XML:



Banco de Dados:  


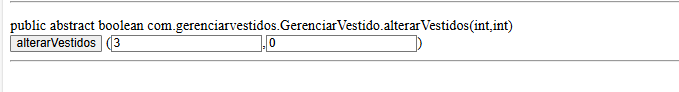
Verificando:

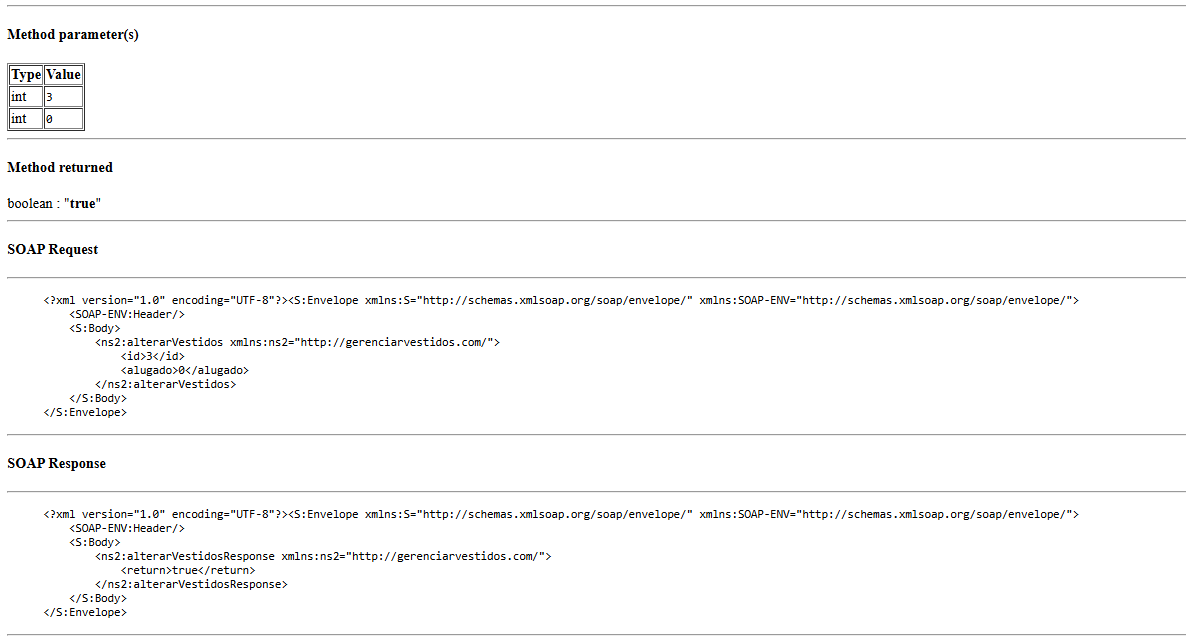


Retorno XML:



Alterando:



Retorno XML:  


Banco de dados:

