paquete

import tabla del BO

@WebServlet(name = "TablaServlet", urlPatterns = {"creartabla", "/tabla(plural)", "/editartabla, "/, "/borrartabla""})

public class TablaServlet extends HttpServlet {

/\*\*

\* Handles the HTTP

\* <code>GET</code> method.

\*

\* @param request servlet request

\* @param response servlet response

\* @throws ServletException if a servlet-specific error occurs

\* @throws IOException if an I/O error occurs

\*/

@Override

protected void doGet(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException {

response.setContentType("text/html;charset=UTF-8");

// Obtiene la cadena de peticion

String userPath = request.getServletPath();

// Verifica el userPath

if (userPath.equals("/editartabla")) {

mostrarEditarTabla(request, response);

}

if (userPath.equals("/eliminartabla")) {

HttpSession sesion = request.getSession();

TablaBO cl = new TablaBO();

List<Tabla> listatabla = (List<Tabla>) sesion.getAttribute("temporal");

//Recorre la lista de los cursos

for(Tabla T : listaTabla){

//Obtiene el id de la Tabla y luego lo elimina

int id = T.getId();

cl.eliminarTabla (id);

}

String usuario = (String) sesion.getAttribute("NombreCompleto"); -> Campos Tabla

Tabla p = new Tabla BO().obtenerUsuario Tabla (usuario);

int id = p.getId();

Si tiene más foraneas hace lo mismo …

Tabla BO uni =new Tabla BO();

List< Tabla > Lista2 = new ArrayList< Tabla >();

Lista2 = uni.getAllTodo(id);

request.setAttribute("listaUni", Lista2);

RequestDispatcher view = request.getRequestDispatcher("WEB-INF/views/formulario.jsp");

view.forward(request, response); }

}

private void mostrarEditarTabla(HttpServletRequest request, HttpServletResponse response) throws IOException, NumberFormatException, ServletException {

// Lee los parametros

String idParametro = request.getParameter("id");

int id = Integer.parseInt(idParametro);

// Obtiene la entidad de la capa de negocio

Tabla p = new TablaBO().getById(id);

// Escribe el objeto en memoria

request.setAttribute("tabla", p);

// Asigna a la vista el jsp

RequestDispatcher view = request.getRequestDispatcher("editartabla.jsp");

// Redirecciona a la vista

view.forward(request, response);

}

private void crearTabla(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

// Lee los parametros de la tablas

String campotabla = request.getParameter("campotabla "); -> Campos de la tabla

int u = Integer.parseInt(request.getParameter("TablaForanea")); -> Foraneas

// Crea una entidad con los parametros anteriores

tabla p = new tabla();

p.setNombre(nombre); -> Set de los campos de la tabla

// Crea un objeto de negocio

TablaBO bo = new TablaBO();

bo.create(p);

// Obtiene la lista de tablas

List<Tabla> lista = new ArrayList<Tabla>();

lista = bo.getAll();

// Escribe el objeto en memoria

request.setAttribute("listaTabla(plural)", lista);

mostrarUniversidades (request, response); -> Foraneas

// Asigna a la vista el jsp

RequestDispatcher view = request.getRequestDispatcher("WEB-INF/views/tabla(plural).jsp");

// Redirecciona a la vista

view.forward(request, response);

}

private void mostrarForaneas(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

TablaForaneaBO bo= new TablaForaneaBO();

List<TablaForanea> lista= new ArrayList< TablaForanea >();

lista=bo.getAll();

// Escribe el objeto en memoria

request.setAttribute("listaUniversidades", lista);

// Establece la vista

RequestDispatcher view = request.getRequestDispatcher("formulario.jsp");

// Redirecciona a la vista

view.forward(request, response);

}

/\*\*

\* Handles the HTTP

\* <code>POST</code> method.

\*

\* @param request servlet request

\* @param response servlet response

\* @throws ServletException if a servlet-specific error occurs

\* @throws IOException if an I/O error occurs

\*/

@Override

protected void doPost(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException {

// Obtiene la cadena de peticion

String userPath = request.getServletPath();

if (userPath.equals("/creartabla")) {

crearTabla(request, response);

}

if (userPath.equals("/editartabla")) {

// Obtiene los parametros

Trae los campos de la tabla y dependiendo del tipo de dato realiza lo siguiente

int id = Integer.parseInt( request.getParameter("idprofesor"));

String nombre = request.getParameter("nombre");

String usuario = request.getParameter("usuario");

String clave = request.getParameter("clave");

String email = request.getParameter("email");

// Crea la entidad

Tabla p = new Tabla();

Set de los campos de la tabla

p.setId(id);

p.setNombre(nombre);

p.setUsuario(usuario);

p.setClave(clave);

p.setEmail(email);

// Modifica la entidad

new TablaBO().update(p);

// Obtiene la lista de tablas

List<Tabla> lista = new ArrayList<Tabla>();

lista = new TablaBO().getAll();

// Escribe el objeto en memoria

request.setAttribute("listaTabla(plural)", lista);

// Asigna a la vista el jsp

RequestDispatcher view = request.getRequestDispatcher("WEB-INF/views/tabla(plural).jsp");

// Redirecciona a la vista

view.forward(request, response);

}

}

/\*\*

\* Returns a short description of the servlet.

\*

\* @return a String containing servlet description

\*/

@Override

public String getServletInfo() {

return "Short description";

}// </editor-fold>

}