# JSP - MVC - DAO - DTO - MYSQL

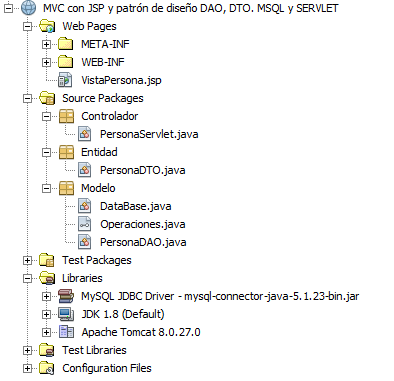
## ENTIDAD:

Representa una “cosa”, "objeto" o "concepto" del mundo real con existencia independiente. Por cada entidad (tabla) de la base de datos, debe de haber una clase en el sistema que haga referencia a esta. Para esto utilizamos el patrón de diseño DTO.

Ejemplo:

|  |  |
| --- | --- |
| **Tabla “Persona” de la BD** | **Clase “PersonaDTO” del sistema** |
|  |  |

### Repositorio Completo de la Aplicación:



Ejemplo de Código **“PersonaDTO”**:

|  |
| --- |
| **package** Entidad;  /\*\* **@author** Luis Angel\*/  **public** **class** PersonaDTO {  **int** id;  String nombres;  String apellidos;  **int** edad;  **public** PersonaDTO(**int** id, String nombres, String apellidos, **int** edad) {  **this**.id = id;  **this**.nombres = nombres;  **this**.apellidos = apellidos;  **this**.edad = edad;  }  **public** PersonaDTO() {  }  //Getter and Setters  } |

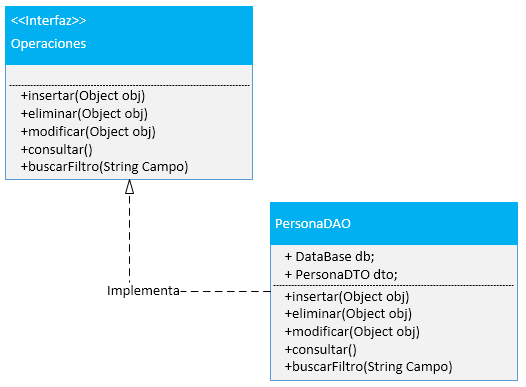
## MODELO:

Es quien se encarga de la comunicación con la base de datos, ejecuta consultas de tipo CRUD y maneja la conexión con la BD.

Para la comunicación de datos entre el sistema y la BD utilizamos el patrón de diseño DAO.

* Contiene una interfaz “Operaciones”, una clase “DataBase” y una clase “PersonaDAO”, donde la clase “PersonaDAO” implementa la interfaz “Operaciones”.

Ejemplo:



Ejemplo de Código **“DataBase”:**

|  |
| --- |
| **package** Modelo;  /\*\***@author** Luis Angel \*/  **public** **class** DataBase {  **private** String driver;  **private** String url;  **private** String usuario;  **private** String contraseña;  **public** DataBase() {  **this**.driver = "com.mysql.jdbc.Driver";  **this**.url = "jdbc:mysql://localhost/MVC-JSP-DAO-MYSQL";  **this**.usuario = "root";  **this**.contraseña = "";  }  //Getter and Setter  } |

Ejemplo de Código **“Operaciones”:**

|  |
| --- |
| **package** Modelo;  **import** java.util.List;  /\*\***@author** Luis Angel\*/  **public** **interface** Operaciones {    **public** String insertar(Object obj);  **public** String eliminar(Object obj);  **public** String modificar(Object obj);  **public** List<?> consultar();  **public** List<?> filtrar(String campo, String criterio);  } |

Ejemplo de Código **“PersonaDAO”:**

|  |
| --- |
| **package** Modelo;  **import** java.util.\*;  **import** java.sql.\*;  **import** Entidad.PersonaDTO;  /\*\***@author** Luis Angel\*/  **public** **class** PersonaDAO **implements** Operaciones {  DataBase db = **new** DataBase();  @Override  **public** String insertar(Object obj) {  //hacemos un casting para traducir el parametro de tipo objeto a tipo persona  PersonaDTO p = (PersonaDTO) obj;  //para conectarme a mysql necesito:  Connection conn;  PreparedStatement pst;  //creamos una sentencia preparada de tipo sql para la consulta  //la sentencia preparada evita inyección de codigo, los valores van en incognito  String sentenciaSql = "insert into persona values (?,?,?,?)";  String respuesta="";  //nos conectamos a la base de datos  **try** {  //cargamos el driver  Class.forName(db.getDriver());  conn = DriverManager.getConnection(db.getUrl(), db.getUsuario(), db.getContraseña());  //preparamos la consulta  pst = conn.prepareStatement(sentenciaSql);  //establecemos los valores de las incognitas de la consulta  pst.setInt(1, p.getId());//el 1 equivale al primer ?  pst.setString(2, p.getNombres());  pst.setString(3, p.getApellidos());  pst.setInt(4, p.getEdad());  //ejecutamos la consulta  //el update es para inserciones, modificaciones y eliminaciones  //para consultas select, se hace executeQuery  **int** filas = pst.executeUpdate();  respuesta="se registraron: "+filas+" elementos";  //cerramos la conexion  conn.close();  } **catch** (Exception e) {  }  **return** respuesta;  }  @Override  **public** String eliminar(Object obj) {  **return** **null**;  }  @Override  **public** String modificar(Object obj) {  **return** **null**;  }  @Override  **public** List<PersonaDTO> consultar() {  List<PersonaDTO> datos = **new** ArrayList<>();  //para conectarme a mysql necesito:  Connection conn;  PreparedStatement pst;  //variable que recupera los datos que retorna la consulta  ResultSet rs;  //creamos una sentencia preparada de tipo sql para la consulta  //la sentencia preparada evita inyección de codigo, los valores van en incognito  String sentenciaSql = "select \* from persona";  //nos conectamos a la base de datos  **try** {  //cargamos el driver  Class.forName(db.getDriver());  conn = DriverManager.getConnection(db.getUrl(), db.getUsuario(), db.getContraseña());  //preparamos la consulta  pst = conn.prepareStatement(sentenciaSql);  //ejecutamos la consulta y guardamos los resultados en el rs  rs = pst.executeQuery();  //hacemos un recorrido en el rs y llenamos la List datos  **while**(rs.next()){  //rs.getInt("nombre\_columna\_en\_bd")  datos.add(**new** PersonaDTO(rs.getInt("id"), rs.getString("nombres"), rs.getString("apellidos"), rs.getInt("edad")));  }  conn.close();  } **catch** (Exception e) {  }  **return** datos;  }    @Override  **public** List<?> filtrar(String campo, String criterio) {  **return** **null**;  }  } |

## CONTROLADOR:

Se encarga de recibir las peticiones hechas por el usuario mediante la vista y las comunica al modelo en caso de que sea una solicitud de manejo de información (por ejemplo, editar un documento o un registro en una base de datos).

* Para esto utilizamos los Servlet en proyectos Web que respectan a JSP.
* **SERVLET:** es una [clase](https://es.wikipedia.org/wiki/Clase_(inform%C3%A1tica)" \o "Clase (informática)) en el [lenguaje de programación Java](https://es.wikipedia.org/wiki/Lenguaje_de_programaci%C3%B3n_Java" \o "Lenguaje de programación Java), utilizada para ampliar las capacidades de un servidor. Aunque los servlets pueden responder a cualquier tipo de solicitudes, estos son utilizados comúnmente para extender las aplicaciones alojadas por servidores web, de tal manera que pueden ser vistos como **[applets de Java](https://es.wikipedia.org/wiki/Applet_Java" \o "Applet Java)** que se ejecutan en servidores en vez de navegadores web.

Ejemplo de Código **“PersonaServlet”:**

|  |
| --- |
| **package** Controlador;  **import** Entidad.PersonaDTO;  **import** Modelo.PersonaDAO;  **import** java.io.\*;  **import** java.util.\*;  **import** javax.servlet.RequestDispatcher;  **import** javax.servlet.ServletException;  **import** javax.servlet.annotation.WebServlet;  **import** javax.servlet.http.HttpServlet;  **import** javax.servlet.http.HttpServletRequest;  **import** javax.servlet.http.HttpServletResponse;  /\*\***@author** Luis Angel\*/  @WebServlet(name = "PersonaServlet", urlPatterns = {"/PersonaServlet"})  **public** **class** PersonaServlet **extends** HttpServlet {  **protected** **void** processRequest(HttpServletRequest request, HttpServletResponse response)  **throws** ServletException, IOException {  response.setContentType("text/html;charset=UTF-8");  **try** (PrintWriter out = response.getWriter()) {  //creamos instancias de las variables a utilizar  PersonaDAO dao = **new** PersonaDAO();  PersonaDTO dto = **new** PersonaDTO();  List<PersonaDTO> datos = **new** ArrayList<>();    String respuesta = "";  //se encarga de despachar las solicitudes que llegan al servlet  RequestDispatcher rd = **null**;    **try** {  //validamos si se le ha dado clic al boton insertar  **if**(request.getParameter("btninsertar")!= **null**){  //agregamos los valores de la vista al objeto dto  dto.setId(Integer.parseInt(request.getParameter("txtid")));  dto.setNombres(request.getParameter("txtnombres"));  dto.setApellidos(request.getParameter("txtapellidos"));  dto.setEdad(Integer.parseInt(request.getParameter("txtedad")));  //insertamos en la bd el obj dto a través del obj dao  respuesta= dao.insertar(dto);  //mandamos la respuesta a la vista  request.setAttribute("respuesta", respuesta);  }    //por ultimo vamos a despachar o mostrar la pagina deseada,  //en este caso la index.jsp para que se vean los datos previamente modificados  rd= request.getRequestDispatcher("index.jsp");  } **catch** (Exception e) {  }  //ejecutamos la respuesta  rd.forward(request, response);  }  }    @Override  **protected** **void** doGet(HttpServletRequest request, HttpServletResponse response)  **throws** ServletException, IOException {  processRequest(request, response);  }  @Override  **protected** **void** doPost(HttpServletRequest request, HttpServletResponse response)  **throws** ServletException, IOException {  processRequest(request, response);  }  } |

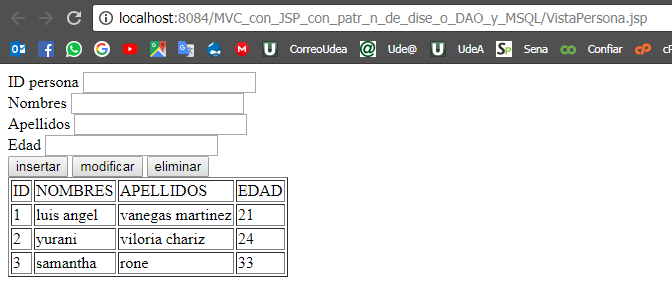
## VISTA:

Es la misma interfaz gráfica. Se encarga de interactuar con el usuario y emitir instrucciones al sistema.

Ejemplo de Código **“index.jsp”:**

|  |
| --- |
| <%--  Document : index  Created on : 24-jun-2018, 17:17:28  Author : Luis Angel  --%>  <%--  como directiva de pagina ponemos:  las importaciones de los paquetes y clases a utilizar.  --%>  <%@page **import** ="Modelo.PersonaDAO"%>  <%@page **import** ="java.util.\*"%>  <%@page **import** ="Entidad.PersonaDTO"%>  <%@page contentType="text/html" pageEncoding="UTF-8"%>  <!DOCTYPE html>  <html>  <head>  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">  <title>vista persona</title>  </head>  <%//lo que esta dentro de estas llaves son codigo JSP o JAVA  //declaramos el dao para realizar directamente algun tipo de operacion  PersonaDAO dao = **new** PersonaDAO();  List<PersonaDTO> datos = **new** ArrayList<>();  %>  <body>  <form name="formPersona" method="POST" action="PersonaServlet">  ID persona <input type="text" name="txtid" ><br>  Nombres <input type="text" name="txtnombres" ><br>  Apellidos <input type="text" name="txtapellidos" ><br>  Edad <input type="text" name="txtedad" ><br>  <input type="submit" name="btninsertar" value="insertar">  <input type="submit" name="btnmodificar" value="modificar">  <input type="submit" name="btneliminar" value="eliminar">  </form>    <table border="lpx">  <tr>  <td>ID</td> <td>NOMBRES</td> <td>APELLIDOS</td> <td>EDAD</td>  </tr>  <%  //hace un recorrido sobre el resultado de la consulta y muestra los datos  datos = dao.consultar();  **for** (PersonaDTO p : datos){  %>  <tr>  <td><%= p.getId()%> </td>  <td><%= p.getNombres()%> </td>  <td><%= p.getApellidos()%> </td>  <td><%= p.getEdad()%> </td>  </tr>  <%  }  %>  </table>  </body>  </html> |

### Vista Desde el Navegador:



### Recomendación al crear e Proyecto en Netbeans:

* Crear una aplicación de tipo Web Application
* Importar la librería o driver de conexión con mysql:

