

UNIVERSIDAD TECNOLÓGICA DEL PERÚ - FISE
Laboratorio de Desarrollo Web Integrado.
I50N-Lab. 04-Rev.1

CARRERA	CURSO	AMBIENTE
Ingeniería de Sistemas e Informática	Desarrollo web integrado	C206

PRACTICA No	NOMBRE DE PRÁCTICA	CODIGO DE LAB.	DURACION (HORAS)
04	MVC	C206	2
Elaborado por Docente	Revisado por Jefe de Lab.	Aprobado por Coordinador	Autorizado por Director
_____ Vicente Machaca Arceda	_____ David Galvez Galvez	_____ Maribel Urquizo Abril	_____ Luis Pinedo Delgado

REVISION	FECHA	DESCRIPCIÓN
0	19/03/2019	Revisión de Guías de Laboratorio

1. OBJETIVOS

- Comprender el patrón de diseño MVC
- Implementar el patrón de diseño MVC

2. TEMAS A TRATAR

- MVC

3. INTRODUCCIÓN (MARCO TEÓRICO)

Patrón MVC

(Fuente: [1,2])

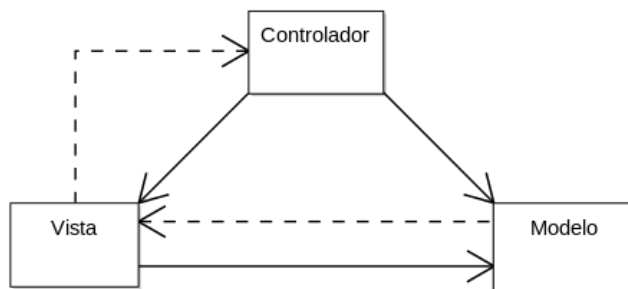


Figura 1. Patrón MVC

Modelo-vista-controlador (MVC) es un patrón de arquitectura de software, que separa los datos y la lógica de negocio de una aplicación de su representación y el módulo encargado de gestionar los eventos y las comunicaciones. Para ello MVC propone la construcción de tres componentes distintos que son el modelo, la vista y el controlador, es decir, por un lado define componentes para la representación de la información, y por otro lado para la interacción del usuario.

MVC con DTO y DAO

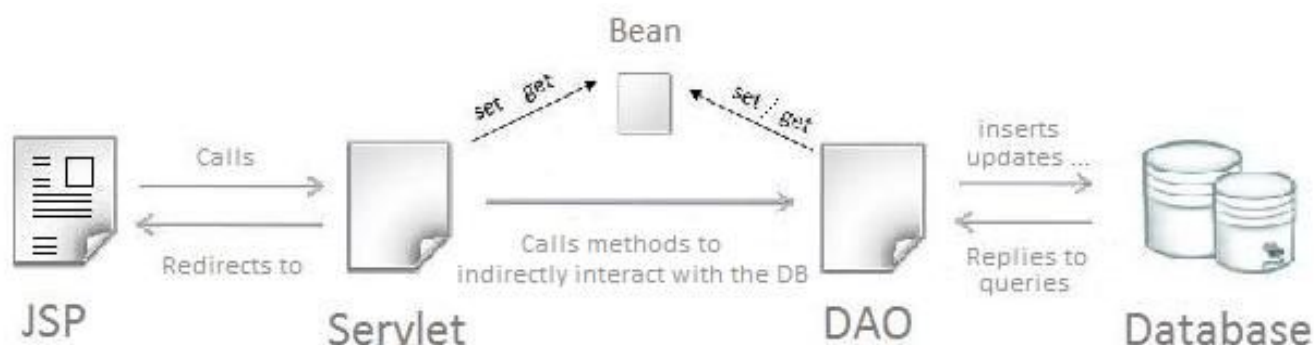


Figura 2. Patrones MVC, DAO y DTO

Como se ve en la imagen el patrón MVC se puede acoplar perfectamente con el patrón DAO y DTO. Como se ve en la Figura 2, los JSP representan la Vista, mientras que los servlets representan los Controladores, Lo modelos están conformados por los Bean (DTO) y DAO.

4. MATERIALES, EQUIPOS, SOFTWARE

- Ordenador
- Editor de texto Sublime, Notepad ++ o Netbeans.

5. FUNDAMENTO Y FÓRMULAS

El quinto taller de Desarrollo Web Integrado, muestra la implementación del patrón MVC.

6. PROCEDIMIENTO (DESCRIPCIÓN)

DESARROLLO DE LA PRÁCTICA

1. Cree una Tabla en Mysql

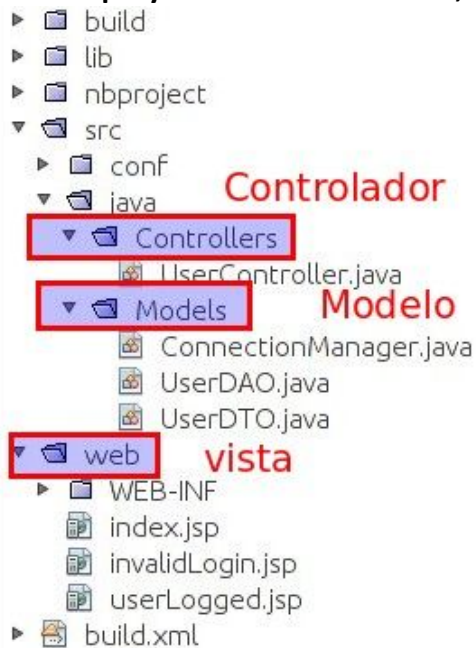
UNIVERSIDAD TECNOLÓGICA DEL PERÚ - FISE
Laboratorio de Desarrollo Web Integrado.
I50N-Lab. 04-Rev.1

```
28 -- Table structure for table `user`
29 --
30
31 CREATE TABLE `user` (
32   `id` bigint(20) UNSIGNED NOT NULL,
33   `alias` varchar(100) DEFAULT NULL,
34   `pass` varchar(100) DEFAULT NULL,
35   `firstname` varchar(100) DEFAULT NULL,
36   `lastname` varchar(100) DEFAULT NULL,
37   `valid` tinyint(1) DEFAULT NULL
38 ) ENGINE=InnoDB DEFAULT CHARSET=latin1;
39
```

2. Inserte dos registros en la Tabla user

```
44 INSERT INTO `user` (`id`, `alias`, `pass`, `firstname`, `lastname`, `valid`) VALUES
45 (1, 'vicente', '123', 'vicente', 'machaca', 1),
46 (2, 'enrique', '123', 'enrique', 'machaca', 1);
47
```

3. Cree un proyecto web en netbeans, consideremos los siguientes archivos:



4. Cree una clase ConnectionManager.java para conectarse a la base de datos

UNIVERSIDAD TECNOLÓGICA DEL PERÚ - FISE
Laboratorio de Desarrollo Web Integrado.
I50N-Lab. 04-Rev.1

```
13 import java.sql.Connection;
14 import java.sql.DriverManager;
15 import java.sql.ResultSet;
16 import java.sql.SQLException;
17 import java.sql.Statement;
18 import java.util.*;
19
20
21 public class ConnectionManager {
22
23     //private static String url = "jdbc:postgresql://localhost:5432/mvc";
24     private static String url = "jdbc:mysql://localhost/mvc";
25     private static String user = "root";
26     private static String password = "*****";
27     private static Connection conn;
28
29     public static Connection getConnection()
30     {
31
32         conn = null;
33         try {
34             Class.forName("com.mysql.jdbc.Driver");
35             conn = DriverManager.getConnection(url, user, password);
36             System.out.println("Connected to the Mysql server successfully.");
37         } catch (SQLException e) {
38             System.out.println(e.getMessage());
39         } catch (Exception e) {
40             System.out.println(e.getMessage());
41         }
42
43         return conn;
44     }
45
46     /*public static void main(String args[]){
47         getConnection();
48     }*/
49 }
```

5. Agregar una clase UserDTO



UNIVERSIDAD TECNOLÓGICA DEL PERÚ - FISE
Laboratorio de Desarrollo Web Integrado.
I50N-Lab. 04-Rev.1

```
public class UserDTO {  
    private String username;  
    private String password;  
    private String firstName;  
    private String lastName;  
    public boolean valid;  
  
    public String getFirstName() {  
        return firstName;  
    }  
  
    public void setFirstName(String newFirstName) {  
        firstName = newFirstName;  
    }  
  
    public String getLastName() {  
        return lastName;  
    }  
  
    public void setLastName(String newLastName) {  
        lastName = newLastName;  
    }  
  
    public String getPassword() {  
        return password;  
    }  
  
    public void setPassword(String newPassword) {  
        password = newPassword;  
    }  
  
    public String getUsername() {  
        return username;  
    }  
}
```

UNIVERSIDAD TECNOLÓGICA DEL PERÚ - FISE
Laboratorio de Desarrollo Web Integrado.
I50N-Lab. 04-Rev.1

```
public void setPassword(String newPassword) {
    password = newPassword;
}

public String getUsername() {
    return username;
}

public void setUsername(String newUsername) {
    username = newUsername;
}

public boolean isValid() {
    return valid;
}

public void setValid(boolean newValid) {
    valid = newValid;
}
}
```

6. Agregar una clase UserDao

```
import java.text.*;
import java.util.*;
import java.sql.*;

public class UserDao
{
    static Connection currentCon = null;
    static ResultSet rs = null;

    public static UserDTO login(UserDTO bean) {

        //preparing some objects for connection
        Statement stmt = null;

        String username = bean.getUsername();
        String password = bean.getPassword();

        String searchQuery =
            "select * from public.user where alias='"
            + username
            + "' AND pass='"
            + password
            + "'";

        // "System.out.println" prints in the console; Normally used to trace the process
        System.out.println("Your user name is " + username);
        System.out.println("Your password is " + password);
        System.out.println("Query: "+searchQuery);
    }
}
```


UNIVERSIDAD TECNOLÓGICA DEL PERÚ - FISE
Laboratorio de Desarrollo Web Integrado.
I50N-Lab. 04-Rev.1

```
try
{
    //connect to DB
    currentCon = DriverManager.getConnection();
    stmt=currentCon.createStatement();
    rs = stmt.executeQuery(searchQuery);
    boolean more = rs.next();

    // if user does not exist set the isValid variable to false
    if (!more)
    {
        System.out.println("Sorry, you are not a registered user! Please sign up first");
        bean.setValid(false);
    }

    //if user exists set the isValid variable to true
    else if (more)
    {
        String firstName = rs.getString("FirstName");
        String lastName = rs.getString("LastName");

        System.out.println("Welcome " + firstName);
        bean.setFirstName(firstName);
        bean.setLastName(lastName);
        bean.setValid(true);
    }
}

catch (Exception ex)
{
    System.out.println("Log In failed: An Exception has occurred! " + ex);
}

//some exception handling
finally
{
    if (rs != null) {
        try {
            rs.close();
        } catch (Exception e) {}
        rs = null;
    }

    if (stmt != null) {
        try {
            stmt.close();
        } catch (Exception e) {}
        stmt = null;
    }

    if (currentCon != null) {
        try {
            currentCon.close();
        } catch (Exception e) {}
    }

    currentCon = null;
}

return bean;
}
```


UNIVERSIDAD TECNOLÓGICA DEL PERÚ - FISE
Laboratorio de Desarrollo Web Integrado.
I50N-Lab. 04-Rev.1

7. Agregar el controlador UserController desde netbeans (add servlet) y agregar el siguiente código al método processRequest():

```
protected void processRequest(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
    response.setContentType("text/html;charset=UTF-8");

    try
    {
        UserDTO user = new UserDTO();
        user.setUsername(request.getParameter("un"));
        user.setPassword(request.getParameter("pw"));

        System.out.println("username:"+request.getParameter("un"));
        System.out.println("password:"+request.getParameter("pw"));

        user = UserDAO.login(user);

        if (user.isValid())
        {
            HttpSession session = request.getSession(true);
            session.setAttribute("currentSessionUser",user);
            response.sendRedirect("userLogged.jsp"); //logged-in page
        }

        else
            response.sendRedirect("invalidLogin.jsp"); //error page
    }

    catch (Throwable theException)
    {
        System.out.println(theException);
    }
}
```

considere que la url de acceso al servlet es:

```
import Models.UserDTO;
import Models.UserDAO;

/**
 * @author vicente
 */
@WebServlet(urlPatterns = {"/UserController"})
public class UserController extends HttpServlet {
```

UNIVERSIDAD TECNOLÓGICA DEL PERÚ - FISE
Laboratorio de Desarrollo Web Integrado.
I50N-Lab. 04-Rev.1

8. Agregamos 3 vistas:

index.jsp

```

1  <%@ page language="java"
2      contentType="text/html; charset=windows-1256"
3      pageEncoding="windows-1256"
4  %>
5
6  <!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
7      "http://www.w3.org/TR/html4/loose.dtd">
8
9  <html>
10     <head>
11         <meta http-equiv="Content-Type" content="text/html; charset=windows-1256">
12         <title>Login Page</title>
13     </head>
14
15     <body>
16         <form action="UserController" method="post">
17
18             Please enter your username
19             <input type="text" name="un"/><br>
20
21             Please enter your password
22             <input type="text" name="pw"/>
23
24             <input type="submit" value="submit">
25
26         </form>
27     </body>
28 </html>

```

invalidLogin.jsp

```

1  <%@ page language="java"
2      contentType="text/html; charset=windows-1256"
3      pageEncoding="windows-1256"
4  %>
5
6  <!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
7      "http://www.w3.org/TR/html4/loose.dtd">
8
9  <html>
10     <head>
11         <meta http-equiv="Content-Type"
12             content="text/html; charset=windows-1256">
13         <title>Invalid Login</title>
14     </head>
15
16     <body>
17         <center>
18             Sorry, you are not a registered user! Please sign up first
19         </center>
20     </body>
21 </html>

```

UNIVERSIDAD TECNOLÓGICA DEL PERÚ - FISE
Laboratorio de Desarrollo Web Integrado.
I50N-Lab. 04-Rev.1

userLogged

```

1  <%@ page language="java"
2      contentType="text/html; charset=windows-1256"
3      pageEncoding="windows-1256"
4      import="Models.UserDTO"
5  %>
6
7  <!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
8      "http://www.w3.org/TR/html4/loose.dtd">
9
10 <html>
11
12     <head>
13         <meta http-equiv="Content-Type"
14             content="text/html; charset=windows-1256">
15         <title>    User Logged Successfully    </title>
16     </head>
17
18     <body>
19
20         <center>
21             <% UserDTO currentUser = (UserDTO)session.getAttribute("currentSessionUser");%>
22
23             Welcome <%= currentUser.getFirstName() + " " + currentUser.getLastName() %>
24         </center>
25
26     </body>
27
28 </html>

```

9. No olvide agregar la librería correspondiente al conector de mysql.

7. PROBLEMAS PROPUESTOS

1. Agregue un formulario para el registro de nuevos usuarios, este debe de registrar los datos en la base de datos.

8. BIBLIOGRAFÍA Y FUENTES PARA CONSULTAS

[1] "More deeply, the framework exists to separate the representation of information from user interaction." The DCI Architecture: A New Vision of Object-Oriented Programming - Trygve Reenskaug and James Coplien - March 20, 2009.

[2] "... the user input, the modeling of the external world, and the visual feedback to the user are explicitly separated and handled by three types of object." Applications Programming in Smalltalk-80(TM):How to use Model-View-Controller (MVC).

9. PRECAUCIONES Y RECOMENDACIONES

1. Está prohibido que instale cualquier programa en la computadora y que cambie las configuraciones de la PC.
2. Está prohibido que utilice la computadora para otros fines que no sean los correspondientes a la sesión de aprendizaje.
3. Está prohibido que utilice su cuenta de correo institucional para otros fines que no sean los académicos.
4. Se recomienda realizar una copia de todos sus archivos desarrollados en las sesiones de aprendizaje.
5. Debe revisar con el programa antivirus todos los dispositivos externos que se vayan a conectar a la computadora (*como memorias pendrive USB, discos duros externos*) antes de realizar cualquier copia de archivos.
6. Antes de retirarse del laboratorio, debe cerrar sus sesiones abiertas en Canvas y las sesiones de su cuenta de correo electrónico.
7. Al finalizar la sesión debe cerrar correctamente el sistema operativo y apagar la PC.