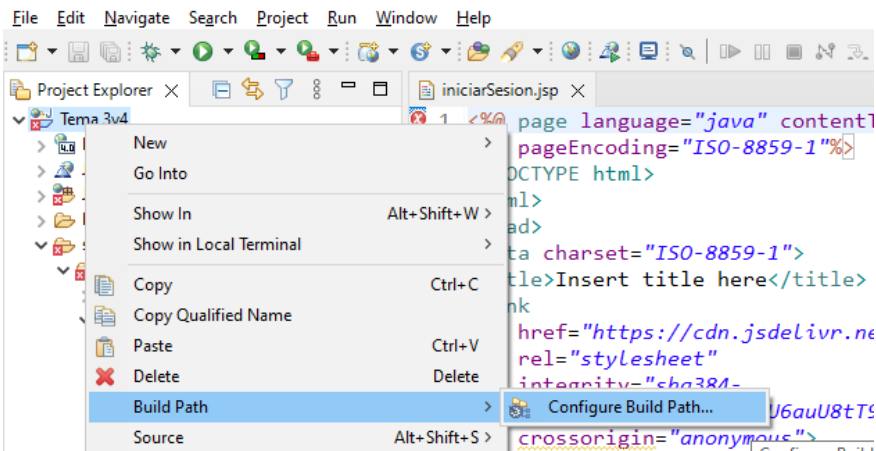
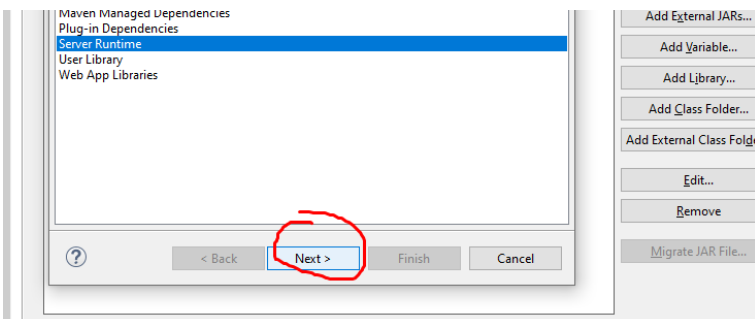
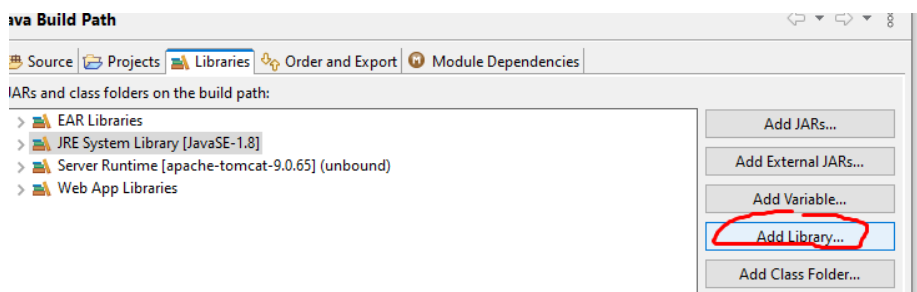


PRIMERO: CREA SERVER TOMCAT

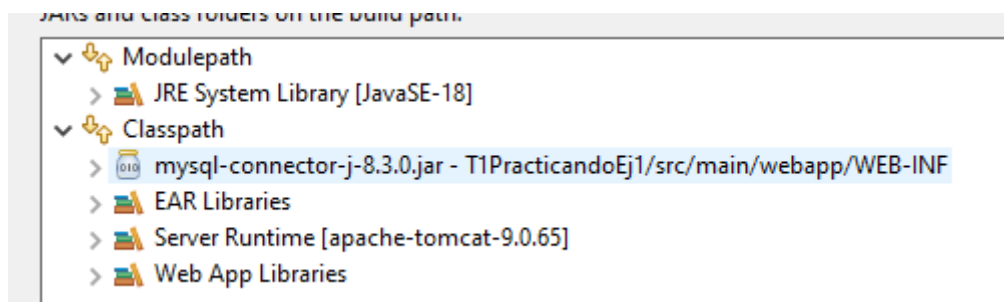


PARAR SOLUCIONAR ERROR
CON <% %>



+ apply and close

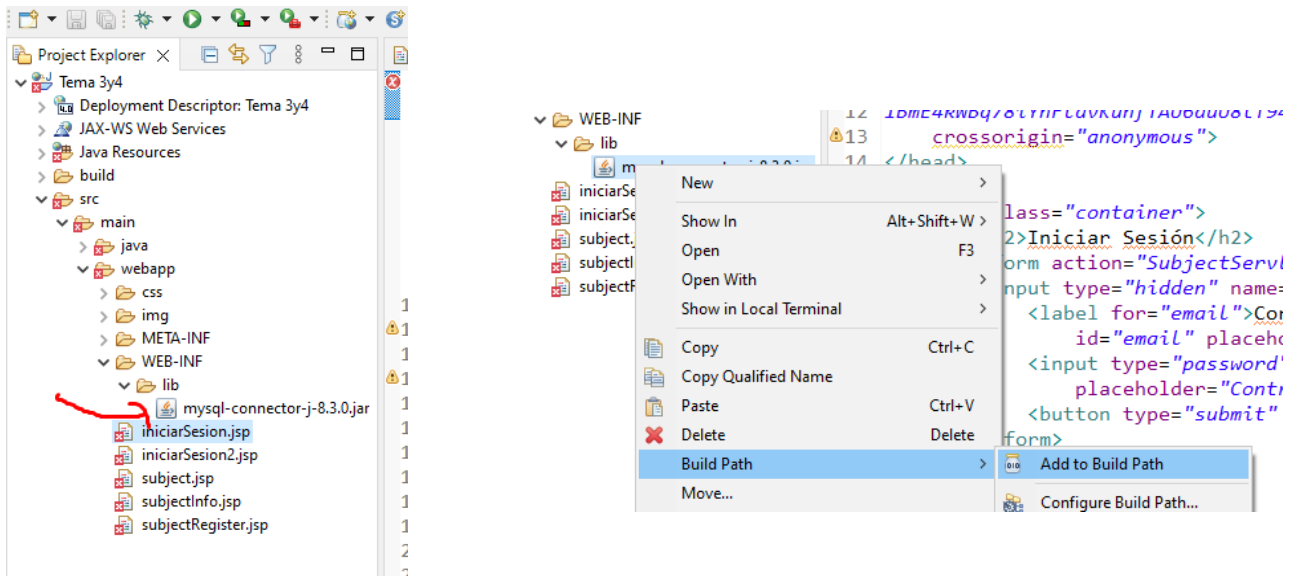
SI TE APARECEN 2 APARTADOS DE LIBRARIES



Lo colocas dentro del Classpath

PARA AÑADIR JAR DE SQL CONNECTION.JAR

Arrastramos el JAR a la carpeta LIB...

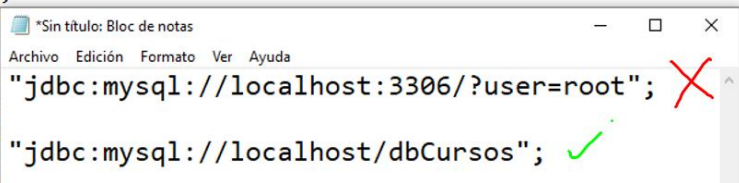


MySqlConexion.java

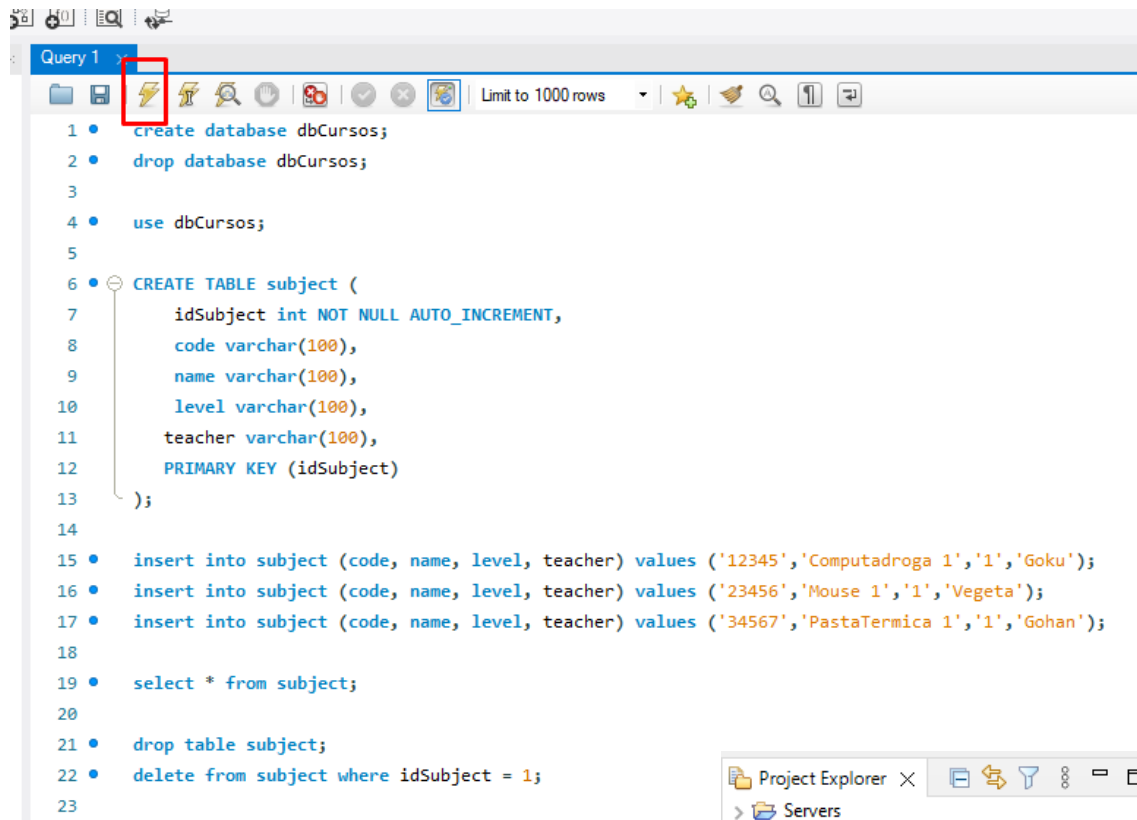
```
1 package util;
2
3 import java.sql.Connection;
4
5
6
7 public class MySqlConnection {
8     public static Connection getConexion() {
9         Connection con = null;
10         try {
11             Class.forName("com.mysql.cj.jdbc.Driver").newInstance();
12             String url = "jdbc:mysql://localhost/dbCursos?useSSL=false&useTimezone=true&serverTimezone=UTC";
13             String usr = "root";
14             String psw = "mysql";
15             con = DriverManager.getConnection(url,usr,psw);
16         } catch (ClassNotFoundException e) {
17             // TODO: handle exception
18             System.out.println("Error >> Driver no Instalado!!"+e.getMessage());
19         } catch (SQLException e) {
20             // TODO: handle exception
21             System.out.println("Error >> de conexión con la BD"+e.getMessage());
22         } catch (Exception e) {
23             // TODO: handle exception
24             System.out.println("Error >> general: "+ e.getMessage());
25         }
26         return con;
27     }
28 }
29
```

```
MySQLConexion.java X
1 package util;
2
3 import java.sql.Connection;
4
5
6
7 public class MySQLConexion {
8     public static Connection getConexion() {
9         Connection con = null;
10        try {
11            Class.forName("com.mysql.cj.jdbc.Driver").newInstance();
12            String url = "jdbc:mysql://localhost/dbCursos?useSSL=false&useTimezone=true&serverTimezone=UTC";
13            String usr = "root";
14            String psw = "mysql";
15            con = DriverManager.getConnection(url,usr,psw);
16        } catch (ClassNotFoundException e) {
17            // TODO: handle exception
18            System.out.println("Error >> Driver no Instalado!!"+e.getMessage());
19        } catch (SQLException e) {
20            // TODO: handle exception
21            System.out.println("Error >> de conexión con la BD"+e.getMessage());
22        } catch (Exception e) {
23            // TODO: handle exception
24            System.out.println("Error >> general: "+ e.getMessage());
25        }
26        return con;
27    }
28 }
29
```

```
try {
    Class.forName("com.mysql.cj.jdbc.Driver").newInstance();
    String url = "jdbc:mysql://localhost/dbCursos?useSSL=false&useTimezone=true&serverTimezone=UTC";
    String usr = "root";
    String psw = "mysql";
    con = DriverManager.getConnection(url,usr,psw);
} catch (ClassNotFoundException e) {
    // TODO: handle exception
    System.out.println("Error >> Driver no Instalado!!"+e.getMessage());
} catch (SQLException e) {
    // TODO: handle exception
    System.out.println("Error >> de conexión con la BD"+e.getMessage());
} catch (Exception e) {
    // TODO: handle exception
    System.out.println("Error >> general: "+ e.getMessage());
}
```

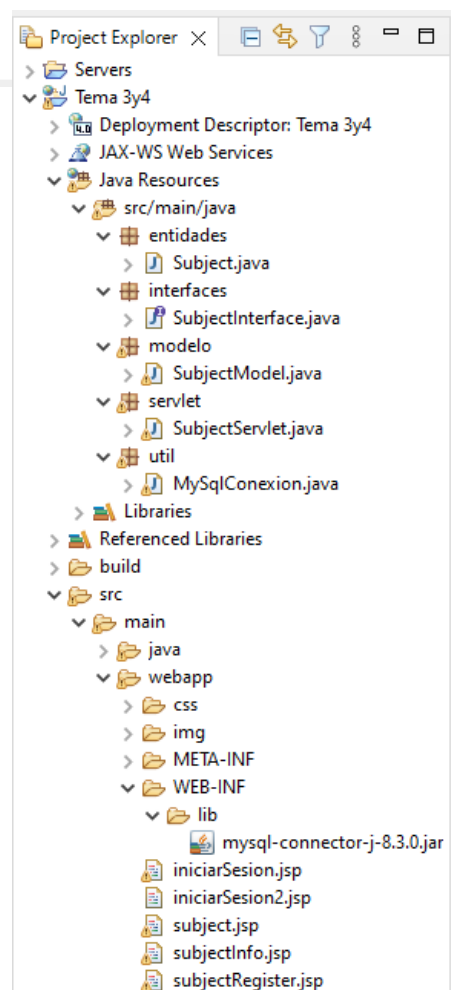


PARA CREAR LA BASE DE DATOS



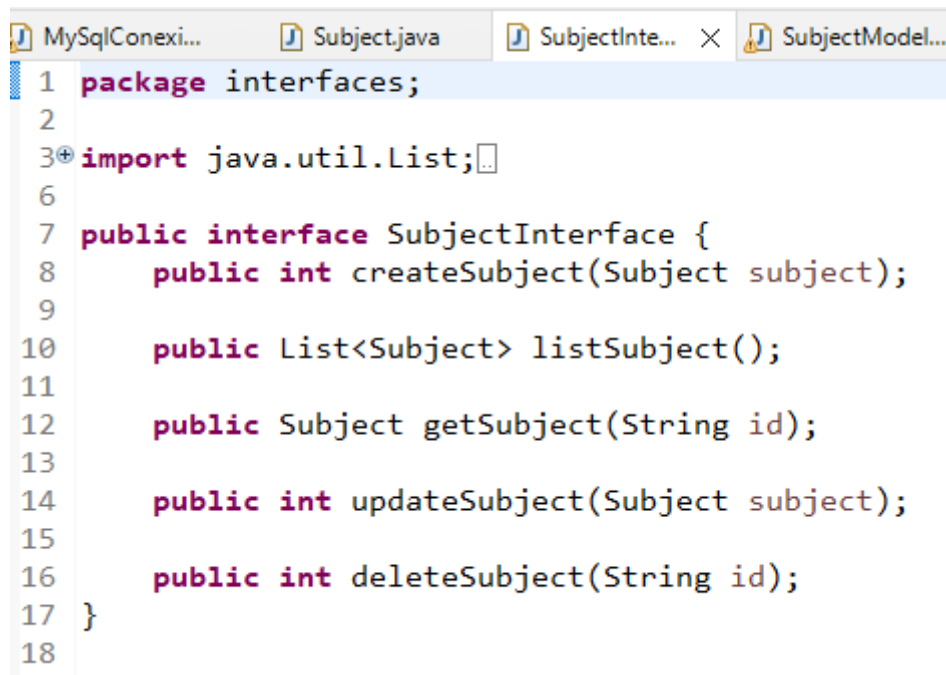
```
Query 1
1 • create database dbCursos;
2 • drop database dbCursos;
3
4 • use dbCursos;
5
6 • CREATE TABLE subject (
7     idSubject int NOT NULL AUTO_INCREMENT,
8     code varchar(100),
9     name varchar(100),
10    level varchar(100),
11    teacher varchar(100),
12    PRIMARY KEY (idSubject)
13 );
14
15 • insert into subject (code, name, level, teacher) values ('12345','Computadrogena 1','1','Goku');
16 • insert into subject (code, name, level, teacher) values ('23456','Mouse 1','1','Vegeta');
17 • insert into subject (code, name, level, teacher) values ('34567','PastaTermica 1','1','Gohan');
18
19 • select * from subject;
20
21 • drop table subject;
22 • delete from subject where idSubject = 1;
23
```

EXPLORADOR COMPLETO DEL PROYECTO



PASITOS

1. Estructurar del proyecto (carpetas, packages, implementar .JAR, solucionar error)
2. MySQL WORKBENCH
3. MySqlConnection.java
4. Entidades -> atributos + getters&Setters
5. Interfaces



```
1 package interfaces;
2
3 import java.util.List;
4
5
6
7 public interface SubjectInterface {
8     public int createSubject(Subject subject);
9
10    public List<Subject> listSubject();
11
12    public Subject getSubject(String id);
13
14    public int updateSubject(Subject subject);
15
16    public int deleteSubject(String id);
17 }
18
```

- a. método create retorna int y recibe Entidad
- b. método read(list) retorna List<Entidad>
- c. método update retorna int y recibe Entidad
- d. método delete retorna int y recibe String id
- e. método get retorna Entidad y recibe String id

6. Model

Implementa los métodos del interfaz.

PARA EJECUTAR UN

rs = psm.executeQuery();

value = psm.executeUpdate();

Primero tienes que llenar la variable PreparedStatement con

psm.setString(1,id);

psm.setString(1, subject.getCode());

a. Para método create

```
@Override
public int createSubject(Subject subject) {
    // TODO Auto-generated method stub
    int value = 0;
    Connection cn= null;
    PreparedStatement psm=null;
    try {
        cn= MySqlConnection.getConnection();
        String sql="INSERT INTO Subject VALUES (null, ?,?,?,?)";
        psm=cn.prepareStatement(sql);
        psm.setString(1, subject.getCode());
        psm.setString(2, subject.getName());
        psm.setString(3, subject.getLevel());
        psm.setString(4, subject.getTeacher());

        value=psm.executeUpdate();
    } catch (Exception e) {
        // TODO: handle exception
        e.printStackTrace();
    }finally {
        try {
            if(psm !=null) psm.close();
            if(cn != null) cn.close();
        } catch (Exception e) {
            // TODO: handle exception
            e.printStackTrace();
        }
    }
    return 0;
}
```

b. Para método list

```
@Override
public List<Subject> listSubject() {
    // TODO Auto-generated method stub
    List<Subject> listSubject = new ArrayList<Subject>();
    Connection cn=null;
    PreparedStatement psm=null;
    ResultSet rs=null;

    try {
        cn = MySqlConnection.getConnection();
        String sql="SELECT * FROM Subject";
        psm = cn.prepareStatement(sql);
        rs=psm.executeQuery();
        while (rs.next()) {
            Subject subj = new Subject();
            subj.setId(rs.getString("idSubject"));
            subj.setCode(rs.getString("code"));
            subj.setName(rs.getString("name"));
            subj.setLevel(rs.getString("level"));
            subj.setTeacher(rs.getString("teacher"));
            listSubject.add(subj);
        }
    } catch (Exception e) {
        // TODO: handle exception
        e.printStackTrace();
    } finally {
        try {
            if(rs != null) rs.close();
            if(psm != null) psm.close();
            if(cn != null) cn.close();
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
    return listSubject;
}
```

c. Para método update

```
@Override
public int updateSubject(Subject subject) {
    // TODO Auto-generated method stub
    int value = 0;
    Connection cn= null;
    PreparedStatement psm=null;
    try {
        cn= MySqlConnection.getConnection();
        String sql="UPDATE Subject SET code=?,
            + "name=?,level=?,teacher=? where idSubject=?";
        psm=cn.prepareStatement(sql);
        psm.setString(1, subject.getCode());
        psm.setString(2, subject.getName());
        psm.setString(3, subject.getLevel());
        psm.setString(4, subject.getTeacher());
        psm.setString(5, subject.getId());

        value=psm.executeUpdate();
    } catch (Exception e) {
        // TODO: handle exception
        e.printStackTrace();
    } finally {
        try {
            if(psm !=null) psm.close();
            if(cn != null) cn.close();
        } catch (Exception e) {
            // TODO: handle exception
            e.printStackTrace();
        }
    }
    return value;
}
```

d. Para el método delete

```
@Override
public int deleteSubject(String id) {
    // TODO Auto-generated method stub
    System.out.println("id dentro del model: "+id);
    int value = 0;
    Connection cn= null;
    PreparedStatement psm=null;
    try {
        cn= MySqlConnection.getConnection();
        String sql="DELETE FROM Subject where idSubject=?";
        psm=cn.prepareStatement(sql);
        psm.setString(1, id);

        value=psm.executeUpdate();
    } catch (Exception e) {
        // TODO: handle exception
        e.printStackTrace();
    }finally {
        try {
            if(psm !=null) psm.close();
            if(cn != null) cn.close();
        } catch (Exception e) {
            // TODO: handle exception
            e.printStackTrace();
        }
    }
    return 0;
}
```

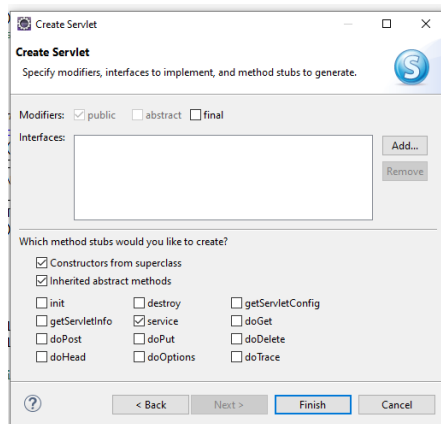
e. Para el método get

```
@Override
public Subject getSubject(String id) {
    // TODO Auto-generated method stub
    Subject subject=null;
    Connection cn=null;
    PreparedStatement psm=null;
    ResultSet rs=null;

    try {
        cn = MySqlConnection.getConnection();
        String sql="SELECT * FROM subject where idSubject=?";
        psm = cn.prepareStatement(sql);
        psm.setString(1, id);
        rs=psm.executeQuery();
        if (rs.next()) {
            subject = new Subject();
            subject.setId(rs.getString("idSubject"));
            subject.setCode(rs.getString("code"));
            subject.setName(rs.getString("name"));
            subject.setLevel(rs.getString("level"));
            subject.setTeacher(rs.getString("teacher"));
        }
    } catch (Exception e) {
        // TODO: handle exception
        e.printStackTrace();
    }finally {
        try {
            if(rs != null) rs.close();
            if(psm != null) psm.close();
            if(cn != null) cn.close();
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
    return subject;
}
```


7. Servlet

CUANDO CREES EL SERVLET tienes q configurar unas cosas:



Así tiene que estar

El servlet interactúa entre **la base de datos (MODEL)** y los **.jps** mediante el poderoso **String type** que recibirá las request y redireccionará el request hacia el método CRUD adecuado.

SERVICE

```
27  */
28  protected void service(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
29      // TODO Auto-generated method stub
30      String type = request.getParameter("type");
31      System.out.println("El parámetro es: " + type);
32      switch(type) {
33          case "register": registerCurso(request,response); break;
34          case "list": listCurso(request,response); break;
35          case "edit": editCurso(request,response); break;
36          case "delete": deleteCurso(request,response); break;
37          case "info": getCurso(request,response); break;
38          default:
39              request.setAttribute("mensaje", "Ocurrió un error");
40              request.getRequestDispatcher("curso.jsp").forward(request, response);
41      }
42  }
```

Los métodos CRUD aquí se identificarán con el type de la sgte forma:

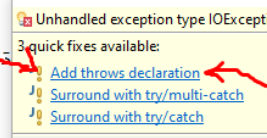
C	"register"	registerSubject
R	"list"	listSubject
U	"edit"	editSubject
D	"delete"	deleteSubject
get	"info"	getSubject

Los métodos los autogeneras con los errores que te saldrán dps de q hayas acabado con el método service.

Y luego de acabar de implementar cada método, te saldrá un error que autocompletará el nombre del método:

getCurso

```
private void getCurso(HttpServletRequest request, HttpServletResponse response) {  
    // TODO Auto-generated method stub  
    String idCurso = request.getParameter("id");  
    CursoModelo cursoModelo = new CursoModelo();  
  
    Curso curso = cursoModelo.getCurso(idCurso);  
    System.out.println("El id del curso: " + curso.getIdCurso());  
  
    request.setAttribute("data", curso);  
    request.getRequestDispatcher("cursoInfo.jsp").forward(request, response);  
}  
  
// TODO Auto-generated method stub  
}
```



deleteCurso

```
private void deleteSubject(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {  
    // TODO Auto-generated method stub  
    String id = request.getParameter("id");  
    System.out.println("Imprime subject : "+id);  
    SubjectModel subjectModel = new SubjectModel();  
    int value = subjectModel.deleteSubject(id);  
  
    if(value == 0) {  
        System.out.println("Entró al list subject");  
        listSubject(request, response);  
    } else {  
        request.setAttribute("mensaje", "Ocurrió un problema");  
        request.getRequestDispatcher("subject.jsp");  
    }  
}
```

editCurso

```
private void editSubject(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {  
    // TODO Auto-generated method stub  
    //Entradas  
    String code = request.getParameter("txtCode");  
    String name = request.getParameter("txtName");  
    String level = request.getParameter("txtLevel");  
    String teacher = request.getParameter("txtTeacher");  
    String id = request.getParameter("idSubject");  
  
    System.out.println("Imprime code : "+code);  
    System.out.println("Imprime name : "+name);  
    System.out.println("Imprime level : "+level);  
    System.out.println("Imprime teacher : "+teacher);  
    System.out.println("Imprime id : "+id);  
  
    //Creamos objeto  
    Subject subject = new Subject();  
    subject.setCode(code);  
    subject.setName(name);  
    subject.setLevel(level);  
    subject.setTeacher(teacher);  
    subject.setId(id);  
  
    //Procesos  
    SubjectModel model = new SubjectModel();  
    int value = model.updateSubject(subject);  
  
    if(value == 0) {  
        System.out.println("Entró al list subject");  
        listSubject(request, response);  
    } else {  
        request.setAttribute("mensaje", "Ocurrió un problema");  
        request.getRequestDispatcher("subject.jsp");  
    }  
}
```

listCurso

```
private void listSubject(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {  
    // TODO Auto-generated method stub  
    SubjectModel subjectModel = new SubjectModel();  
    List<Subject> data=subjectModel.listSubject();  
    request.setAttribute("data", data);  
    request.getRequestDispatcher("subject.jsp").forward(request, response);  
}
```

registerCurso

```
private void registerCurso(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {  
    // TODO Auto-generated method stub  
    String code = request.getParameter("txtCode");  
    String name = request.getParameter("txtName");  
    String level = request.getParameter("txtLevel");  
    String teacher = request.getParameter("txtTeacher");  
  
    Curso curso = new Curso();  
    curso.setCode(code);  
    curso.setName(name);  
    curso.setLevel(level);  
    curso.setTeacher(teacher);  
  
    CursoModelo cursoModelo = new CursoModelo();  
    int value = cursoModelo.createCurso(curso);  
  
    if(value==0) {  
        System.out.println("Ahora se ejecutará listCurso");  
        listCurso(request, response);  
    } else {  
        request.setAttribute("mensaje", "Ocurrió un error");  
        request.getRequestDispatcher("curso.jsp").forward(request, response);  
    }  
}
```

CDNs

JQUERY

Sus CDNs van justo debajo de la etiqueta de cierre del body

Debajo de los CDNs va la validación

```
<script  
type="text/javascript"src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1  
/jquery.min.js"></script>
```

```
<script src="https://cdnjs.cloudflare.com/ajax/libs/jquery-  
validate/1.19.1/jquery.validate.min.js"></script>
```

```
<script type="text/javascript"> //VALIDACI0000000000N </script>
```

J VALIDATE

```
30 //Mediante jquery y la librería validate se valida por medio de mensajes los campos requeridos
31 $(function() {
32     $("form[name='registration']").validate({
33         rules : {
34             email : {
35                 required : true,
36                 email : true
37             },
38             password : {
39                 required : true,
40                 minlength : 8
41             }
42         },
43         messages : {
44             password : {
45                 required : "Ingrese su contraseña",
46                 minlength : "Tu contraseña debe tener al menos 5 caracteres"
47             },
48             email : "Ingrese su correo"
49         },
50         submitHandler : function(form) {
51             form.submit();
52         }
53     });
54 });
55 </script>
56 </html>
```

Validación extra al querer REGISTRAR un CURSO

```
<body>
<div class="container">
    <form action="ServletObjeto" method="post" name="creacion">
        <input type="hidden" value="registrar" name="type">
        <div class="text-center">
            <h3>Registrar Curso</h3>
        </div>
        <div class="mb-3">
            <label>Codigo</label> <input name="txtCodigo" type="text"
            class="form-control">
        </div>
        <div class="mb-3">
            <label>Nombre</label> <input name="txtNombre" type="text"
            class="form-control">
        </div>
        <div class="mb-3">
            <label>Nivel</label> <input name="txtNivel" type="text"
            class="form-control">
        </div>
        <div class="mb-3">
            <label>Profesor</label> <input name="txtProfesor" type="text"
            class="form-control">
        </div>
        <button type="submit" class="btn btn-primary">Enviar Datos</button>
        <button type="button" class="btn btn-primary"
            onclick="location.href='ServletObjeto?type=listar'">Regresar</button>
    </form>
</div>
</body>
```

```
$(function() {
    $("form[name='creacion']").validate({
        rules: {
            txtCodigo: {
                required: true,
                minlength: 8
            },
            txtNombre: {
                required: true
            },
            txtNivel: {
                required: true
            },
            txtProfesor: {
                required: true
            }
        },
        messages: {
            txtCodigo: {
                required: "Ingrese su codigo",
                minlength: "Su codigo debe tener como minimo 8 caracteres"
            },
            txtNombre: "Ingrese su nombre",
            txtNivel: "Ingrese su nivel",
            txtProfesor: "Ingrese un docente"
        },
        submitHandler : function(form) {
            form.submit();
        }
    });
});
</script>
```

BOOTSTRAP

Esto va debajo del TITTLE

```
<link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css"
rel="stylesheet" integrity="sha384-
1BmE4kWbQ78iYhFLdvKuhfTAU6auU8tT94WrHftjDbrCEXSU1oBoqyl2QvZ6jIW3" crossorigin=
"anonymous">
```

8. .JSP

iniciarSesion.jsp

```
<form name="inicioSesion" action="SubjectServlet" method="post">

```

Y para la validación...

```
<!-- jQuery -->
<script type="text/javascript" src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>
<!-- jQuery Validation Plugin -->
<script src="https://cdnjs.cloudflare.com/ajax/libs/jquery-validate/1.19.1/jquery.validate.min.js"></script>
<!-- Validación de formulario -->
<script type="text/javascript">
$(function() {
    $("form[name='inicioSesion']").validate({
        rules : {
            email : {
                required : true,
                email : true
            },
            password : {
                required : true,
                minlength : 5
            }
        },
        messages : {
            password : {
                required : "Ingresa su contraseña",
                minlength : "Tu contraseña debe tener al menos 5 caracteres"
            },
            email : "Ingresa su correo"
        },
        submitHandler : function(form) {
            form.submit();
        }
    });
});
</script>
</html>
```

subject.jsp

```
MySqlConexion.java Subject.java SubjectInterface.java SubjectModel.java SubjectServlet.java iniciarSesion2.jsp subject.jsp x subjectInfo.jsp subjectRegister.jsp
1 <%@ page language="java" contentType="text/html; charset=ISO-8859-1"
2   pageEncoding="ISO-8859-1"%>
3 <%@ page import="java.util.ArrayList"%>
4 <%@ page import="java.util.List"%>
5 <%@ page import="entidades.Subject"%>
6 <!DOCTYPE html>
7 <html>
8 <head>
9   <meta charset="ISO-8859-1">
10  <title>Insert title here</title>
11  <!-- Procurar que en este apartado siempre vayan los css -->
12  <link
13    href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css"
14    rel="stylesheet"
15    integrity="sha384-18mE4kWBq78iYhFLdvhKuhfTAU6auU8tT94NrHftjDbrCEXSU1oBoqylL2QvZ6jIW3"
16    crossorigin="anonymous">
17 </head>
18 <body>
19   <div class="container">
20     <div class="row">
21       <div class="col-12 text-center">
22         <h3>Mantenimiento de Cursos</h3>
23       </div>
24       <div class="col-12">
25         <button type="button" class="btn btn-primary" onclick="window.location='subjectRegister.jsp'" >Registrar</button>
26       </div>
27     <br>
28     <div class="col-12">
29       <table class="table">
30         <thead>
31           <tr>
32             <th>Id</th>
33             <th>Código</th>
34             <th>Nombre</th>
35             <th>Nivel</th>
36             <th>Profesor</th>
37             <th>Acciones</th>
38           </tr>
39         </thead>
```

```

36<th>Acciones</th>
37</tr>
38</thead>
39<tbody>
40<tr>
41<td>
42List<Subject> listSubject = (List<Subject>) request.getAttribute("data");
43if (listSubject != null) {
44for (Subject item : listSubject) {
45
46<tr>
47<td><%=item.getId()%></td>
48<td><%=item.getCode()%></td>
49<td><%=item.getName()%></td>
50<td><%=item.getLevel()%></td>
51<td><%=item.getTeacher()%></td>
52<td>
53<button type="button" onclick="window.Location.href='SubjectServlet?type=info&id=<%=item.getId()%>' " class="btn btn-primary" title="Editar">Editar</button>
54<button type="button" onclick="window.Location.href='SubjectServlet?type=delete&id=<%=item.getId()%>' " class="btn btn-danger" title="Eliminar">Eliminar</button>
55</td>
56</tr>
57<tr>
58<td>
59}
60</td>
61</tr>
62</tbody>
63</table>
64</div>
65<div class="row"></div>
66</div>
67</body>
68</html>
69</html>

```

LAS LINEAS DE LAS ETIQUETAS BUTTON DICE:

```

<button type="button" onclick="window.Location.href='SubjectServlet?type=info&id=<%=item.getId()%>' " class="btn btn-primary">Editar</button>
<button type="button" onclick="window.Location.href='SubjectServlet?type=delete&id=<%=item.getId()%>' " class="btn btn-danger">Eliminar</button>
</td>

```

subjectInfo.jsp

```

16<body>
17<div class="container min-vh-100 d-flex justify-content-center align-items-center">
18
19
20<div class="col-6">
21<form action="SubjectServlet" method="post">
22<div class="form-group">
23<input type="hidden" name="type" value="edit">
24<input type="hidden" name="idSubject" value="<%=subject.getId()%>">
25
26<div class="form-group">
27<label class="text-secondary">Codigo</label>
28<input class="form-control" type="text" name="txtCode" value="<%= (subject!=null)? subject.getCode(): "" %>">
29</div>
30<div class="form-group">
31<label class="text-secondary">Nombre</label>
32<input class="form-control" type="text" name="txtName" value="<%= (subject!=null)? subject.getName(): "" %>">
33</div>
34<div class="form-group">
35<label class="text-secondary">Nivel</label>
36<input class="form-control" type="text" name="txtLevel" value="<%= (subject!=null)? subject.getLevel(): "" %>">
37</div>
38<div class="form-group">
39<label class="text-secondary">Profesor</label>
40<input class="form-control" type="text" name="txtTeacher" value="<%= (subject!=null)? subject.getTeacher(): "" %>">
41</div>
42<div>
43<button type="submit" class="btn btn-primary">Actualizar datos</button>
44<button type="button" class="btn btn-primary" onclick="location.href='SubjectServlet?type=list'">Regresar</button>
45</div>
46</form>
47</div>
48</div>
49</div>
50</body>
51</html>
52</html>

```

Esto es importante (lo de aquí abajo)

```

42<input class="form-control" type="text" name="txtTeacher" value="<%= (subject!=null)? subject.getTeacher(): "" %>">
43</div>
44<div>
45<button type="submit" class="btn btn-primary">Actualizar datos</button>
46<button type="button" class="btn btn-primary" onclick="location.href='SubjectServlet?type=list'">Regresar</button>
47</div>
48</form>
49</div>
50</div>
51</body>
52</html>

```

subjectRegister.jsp

```
MySqlConexion.java Subject.java SubjectInterface.java SubjectModel.java SubjectServlet.java iniciarSesion2.jsp subject.jsp subjectInfo.jsp subjectRegister.jsp X
6 <meta charset="ISO-8859-1">
7 <title>Insert title here</title>
8 <link
9 href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css"
10 rel="stylesheet"
11 integrity="sha384-1BmE4kWBq78iYhFLdvKuhfTAU6auU8tT94WrHftjDbrCEXSU1oBoqyl2QvZ6jIW3"
12 crossorigin="anonymous">
13 </head>
14 <body>
15 <div class="container min-vh-100 d-flex justify-content-center align-items-center">
16 <div class="col-6">
17 <h3>Registrar Curso</h3>
18 <form action="SubjectServlet" method="post">
19 <input type="hidden" name="type" value="register">
20 <div class="form-group">
21 <label>Codigo</label> <input class="form-control" type="text"
22 name="txtCode">
23 </div>
24 <div class="form-group">
25 <label>Nombre</label> <input class="form-control" type="text"
26 name="txtName">
27 </div>
28 <div class="form-group">
29 <label>Nivel</label> <input class="form-control" type="text"
30 name="txtLevel">
31 </div>
32 <div class="form-group">
33 <label>Profesor</label> <input class="form-control" type="text"
34 name="txtTeacher">
35 </div>
36 <br>
37 <button type="submit" class="btn btn-primary">Enviar datos</button>
38 <button type="button" class="btn btn-primary" onclick="location.href='SubjectServlet?type=list'">Regresar</button>
39 </form>
40 </div>
41 </div>
42
43 </body>
44 </html>
```