

ALUMNO : VENTOCILLA SANCHEZ ADONIS

PARTE I: Validando aspectos cognitivos de la competencia

1. Defina Programación Concurrente y Distribuida

La programación concurrente es cuando se realizan programas en los cuales pueda ser posible ejecutar varios procesos al mismo tiempo a través de hilos, lo cual hace que el programa sea más eficaz.

La programación distribuida es cuando se hacen aplicativos capaces de conectarse a través de una red, lo cual hace posible consumir recursos desde otras computadoras.

2. ¿Qué entiende por hilos y socket?

Los hilos son procesos ligeros los cuales ayudan a subdividir procesos de mayor tamaño y mejora los tiempos de respuesta del programa.

Los sockets son una puerta de enlace vía red entre máquinas, lo cual ayuda a la comunicación de información mediante el programa.

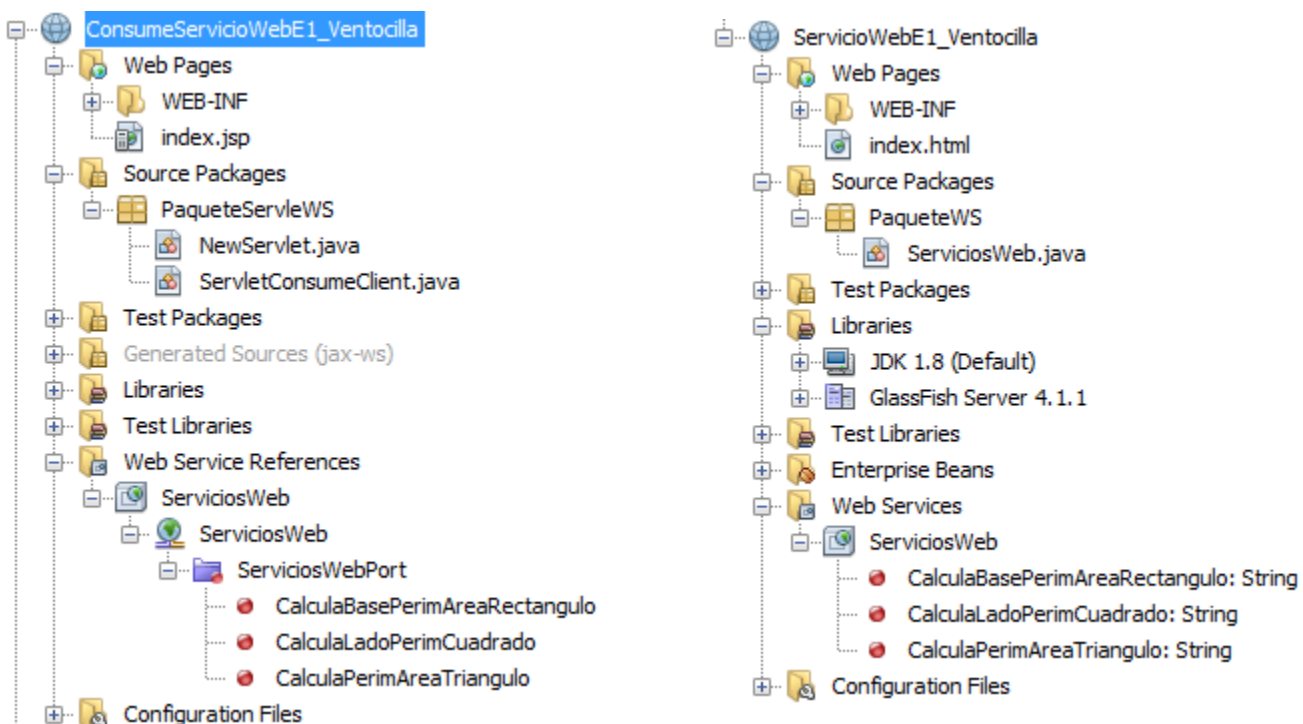
3. Defina servicios web

Son protocolos que ayudan al programa a intercambiar datos desde otro programa.

PARTE II: Validando aspectos procedimentales de la competencia

4. Desarrolle adecuadamente ServicioWebWeb

5. Desarrolle adecuadamente ConsumeServicioWeb



ALUMNO : VENTOCILLA SANCHEZ ADONIS

Código de ServicioWebE1_Ventocilla

```
public class ServiciosWeb {  
  
    /**  
     * Web service operation  
     */  
    @WebMethod(operationName = "CalculaLadoPerimCuadrado")  
    public String CalculaLadoPerimCuadrado(@WebParam(name = "area") double area, @WebParam(name = "medida") String medida) {  
        //TODO write your implementation code here:  
        String respuesta = "";  
        double lado = 0;  
        lado = Math.sqrt(area);  
        double perimetro = 0;  
        perimetro = lado*4;  
        respuesta = "El LADO del cuadrado es : " + lado + " " + medida  
            + "\n y su PERIMETRO es : " + perimetro + " " + medida;  
        return respuesta;  
    }  
  
    /**  
     * Web service operation  
     */  
    @WebMethod(operationName = "CalculaPerimAreaTriangulo")  
    public String CalculaPerimAreaTriangulo(@WebParam(name = "base") double base, @WebParam(name = "altura") double altura, @WebParam(name = "medida") String medida) {  
        //TODO write your implementation code here:  
        String respuesta = "";  
        double area = 0;  
        area = (base*altura)/2;  
        double lado = 0;  
        lado = Math.sqrt(Math.pow(base/2,2)+Math.pow(altura,2));  
        double perimetro = 0;  
        perimetro = (2*lado)+base;  
        respuesta = "El PERIMETRO del triangulo es : " + perimetro + " " + medida  
            + "\n y su ÁREA es : " + area + " " + medida;  
        return respuesta;  
    }  
  
    @WebMethod(operationName = "CalculaBasePerimAreaRectangulo")  
    public String CalculaBasePerimAreaRectangulo(@WebParam(name = "diagonal") double diagonal, @WebParam(name = "altura") double altura, @WebParam(name = "medida") String medida) {  
        //TODO write your implementation code here:  
        String respuesta = "";  
        double base = 0;  
        base = Math.sqrt(Math.pow(diagonal,2)-Math.pow(altura,2));  
        double perimetro = 0;  
        perimetro = 2*(altura+base);  
        double area = 0;  
        area = (base*altura);  
        respuesta = "La BASE del rectangulo es : " + base + " " + medida  
            + "\n su PERIMETRO es : " + perimetro + " " + medida  
            + "\n y su ÁREA es : " + area + " " + medida + " ";  
        return respuesta;  
    }  
}
```

Operations (3)

Add Operation... Remove Operation

CalculaLadoPerimCuadrado

Parameters		Output	Faults	Description
Parameter Name			Parameter Type	
area			double	
medida			java.lang.String	

CalculaPerimAreaTriangulo

Parameters		Output	Faults	Description
Parameter Name			Parameter Type	
base			double	
altura			double	
medida			java.lang.String	

CalculaBasePerimAreaRectangulo

Parameters		Output	Faults	Description
Parameter Name			Parameter Type	
diagonal			double	
altura			double	
medida			java.lang.String	

ServiciosWeb Web Service Tester

This form will allow you to test your web service implementation ([WSDL File](#))

To invoke an operation, fill the method parameter(s) input boxes and click on the button labeled with the method name.

Methods :

```
public abstract java.lang.String paquetews.ServiciosWeb.calculaBasePerimAreaRectangulo(double,double,java.lang.String)
calculaBasePerimAreaRectangulo (  ,  ,  )
```

```
public abstract java.lang.String paquetews.ServiciosWeb.calculaPerimAreaTriangulo(double,double,java.lang.String)
calculaPerimAreaTriangulo (  ,  ,  )
```

```
public abstract java.lang.String paquetews.ServiciosWeb.calculaLadoPerimCuadrado(double,java.lang.String)
calculaLadoPerimCuadrado (  ,  )
```

ServiciosWeb Web Service Tester

This form will allow you to test your web service implementation ([WSDL File](#))

To invoke an operation, fill the method parameter(s) input boxes and click on the button labeled with the method name.

Methods :

```
public abstract java.lang.String paquetews.ServiciosWeb.calculaBasePerimAreaRectangulo(double,double,java.lang.String)
calculaBasePerimAreaRectangulo (2  , 1  , m  )
```

```
public abstract java.lang.String paquetews.ServiciosWeb.calculaPerimAreaTriangulo(double,double,java.lang.String)
calculaPerimAreaTriangulo (2  , 1  , m  )
```

```
public abstract java.lang.String paquetews.ServiciosWeb.calculaLadoPerimCuadrado(double,java.lang.String)
calculaLadoPerimCuadrado (1  , m  )
```

ALUMNO : VENTOCILLA SANCHEZ ADONIS

calculaBasePerimAreaRectangulo Method invocation

Method parameter(s)

Type	Value
double	2
double	1
java.lang.String	m

Method returned

java.lang.String : "La BASE del rectangulo es : 1.7320508075688772 m su PERIMETRO es : 5.464101615137754 m y su ÁREA es : 1.7320508075688772 m²"

calculaPerimAreaTriangulo Method invocation

Method parameter(s)

Type	Value
double	2
double	1
java.lang.String	m

Method returned

java.lang.String : "El PERIMETRO del triangulo es : 4.82842712474619 m y su ÁREA es : 1.0 m"

calculaLadoPerimCuadrado Method invocation

Method parameter(s)

Type	Value
double	1
java.lang.String	m

Method returned

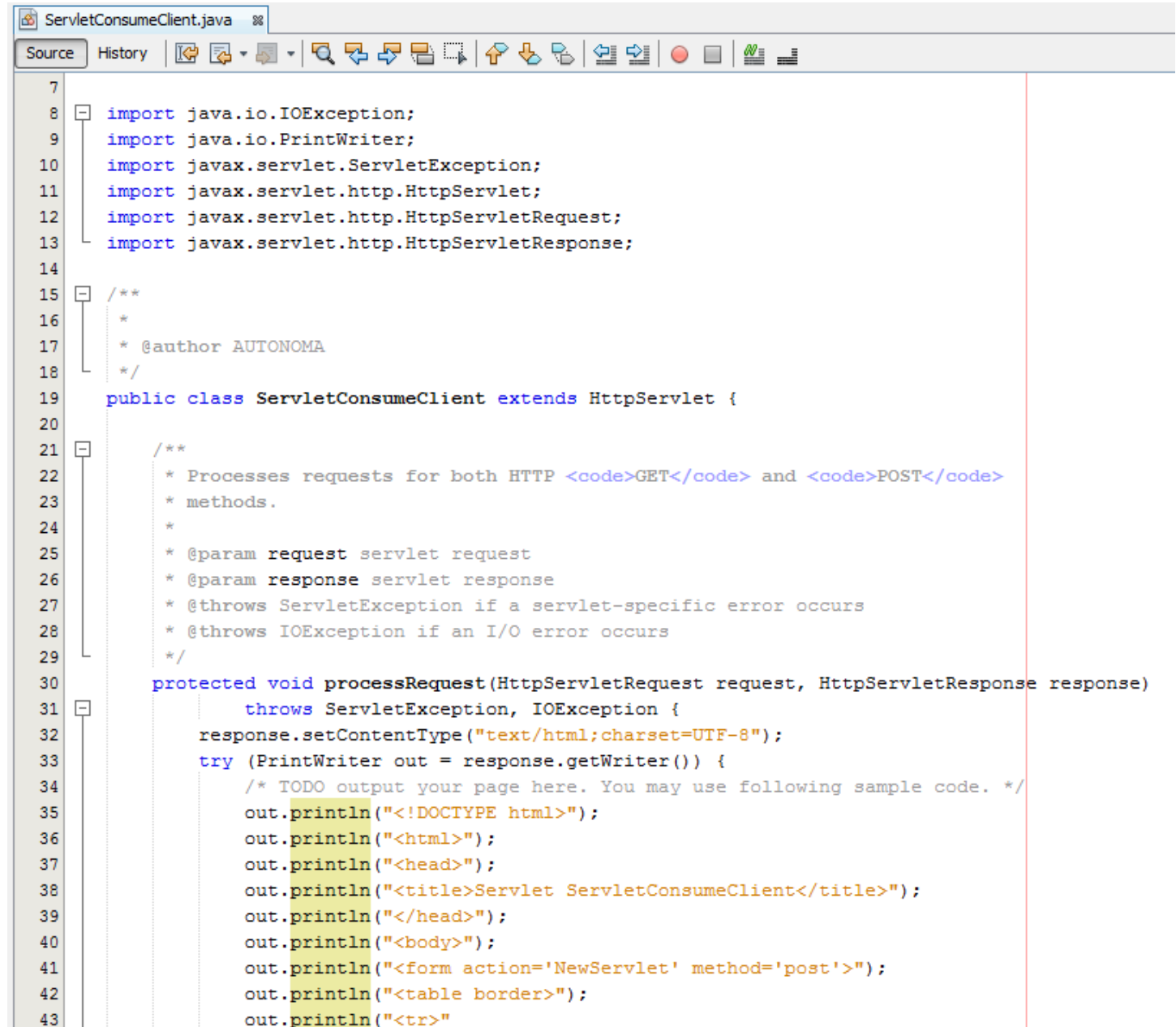
java.lang.String : "El LADO del cuadrado es : 1.0 m y su PERIMETRO es : 4.0 m"

ALUMNO : VENTOCILLA SANCHEZ ADONIS

Codigo de Consume ServicioWebE1_Ventocilla

Para esta parte se creo dos servlet: el primero para colocar los datos en una tabla y el otro para mostrar las respuestas en una tabla.

SERVLET ServletConsumeClient.java



```
ServletConsumeClient.java
Source History
7
8 import java.io.IOException;
9 import java.io.PrintWriter;
10 import javax.servlet.ServletException;
11 import javax.servlet.http.HttpServlet;
12 import javax.servlet.http.HttpServletRequest;
13 import javax.servlet.http.HttpServletResponse;
14
15 /**
16  *
17  * @author AUTONOMA
18  */
19 public class ServletConsumeClient extends HttpServlet {
20
21     /**
22      * Processes requests for both HTTP <code>GET</code> and <code>POST</code>
23      * methods.
24      *
25      * @param request servlet request
26      * @param response servlet response
27      * @throws ServletException if a servlet-specific error occurs
28      * @throws IOException if an I/O error occurs
29      */
30     protected void processRequest(HttpServletRequest request, HttpServletResponse response)
31         throws ServletException, IOException {
32         response.setContentType("text/html;charset=UTF-8");
33         try (PrintWriter out = response.getWriter()) {
34             /* TODO output your page here. You may use following sample code. */
35             out.println("<!DOCTYPE html>");
36             out.println("<html>");
37             out.println("<head>");
38             out.println("<title>Servlet ServletConsumeClient</title>");
39             out.println("</head>");
40             out.println("<body>");
41             out.println("<form action='NewServlet' method='post'>");
42             out.println("<table border>");
43             out.println("<tr>
```

ALUMNO : VENTOCILLA SANCHEZ ADONIS

```
ServletConsumeClient.java
Source History
40 out.println("<body>");
41 out.println("<form action='NewServlet' method='post'>");
42 out.println("<table border>");
43 out.println("<tr>"
44     + "<th colspan=6>Operaciones</th>"
45     + "</tr>"
46     + "<tr>"
47     + "<th colspan=2>CUADRADO</th>"
48     + "<th colspan=2>TRIANGULO</th>"
49     + "<th colspan=2>RECTANGULO</th>"
50     + "</tr>"
51     + "<tr>"
52     + "<td colspan=2>Calcular el Lado y el Perimetro <br> Ingresando:</td>"
53     + "<td colspan=2>Calcular el Perimetro y Area <br> Ingresando:</td>"
54     + "<td colspan=2>Calcular la Base y el Perimetro <br> Ingresando:</td>"
55     + "</tr>"
56     + "<td>Area: </td> <td><input type=text name=areaC ></td>"
57     + "<td>Base: </td> <td><input type=text name=baseT ></td>"
58     + "<td>Diagonal: </td> <td><input type=text name=diagonalR ></td>"
59     + "</tr>"
60     + "</tr>"
61     + "<td>Medida: </td> "
62     + "<td>"
63     + "<select name=medida>"
64     + "<option>Metros</option>"
65     + "</select>"
66     + "</td>"
67     + "<td>Altura: </td> <td><input type=text name=alturaI ></td>"
68     + "<td>Altura: </td> <td><input type=text name=alturaR ></td>"
69     + "</tr>"
70     + "<tr>"
71     + "<td></td> <td></td>"
72     + "<td>Medida: </td> "
73     + "<td>"
74     + "<select name=medida>"
75     + "<option>Metros</option>"
76     + "</select>"
```

ALUMNO : VENTOCILLA SANCHEZ ADONIS

```
ServletConsumeClient.java
Source History
72 + "<td>Medida: </td> "
73 + "<td>"
74 + "<select name=medida>"
75 + "<option>Metros</option>"
76 + "</select>"
77 + "</td>"
78 + "<td>Medida: </td> "
79 + "<td>"
80 + "<select name=medida>"
81 + "<option>Metros</option>"
82 + "</select>"
83 + "</td>"
84 + "</tr>"
85 + "<tr> "
86 + "<td colspan=6><center><input type=submit value=Calcular operaciones></center></td>"
87 + "</tr>";
88 out.println("</table>");
89 out.println("</form>");
90 out.println("</body>");
91 out.println("</html>");
92 }
93 }
94
95 + HttpServlet methods. Click on the + sign on the left to edit the code.
133
134 }
135
```

Como se puede observar toda la tabla dentro de un formulario que redirecciona al otro servlet con el método “post” cuando se hace click al botón “calcular operaciones” con tipo submit

←

→

↺

🏠

📄

localhost:8083/ConsumeServicioWebE1_Ventocilla/ServletConsumeClient

🎓 Universidad Autónoma

🎓 Login Autónoma Virt

📄 COMPARTIDA

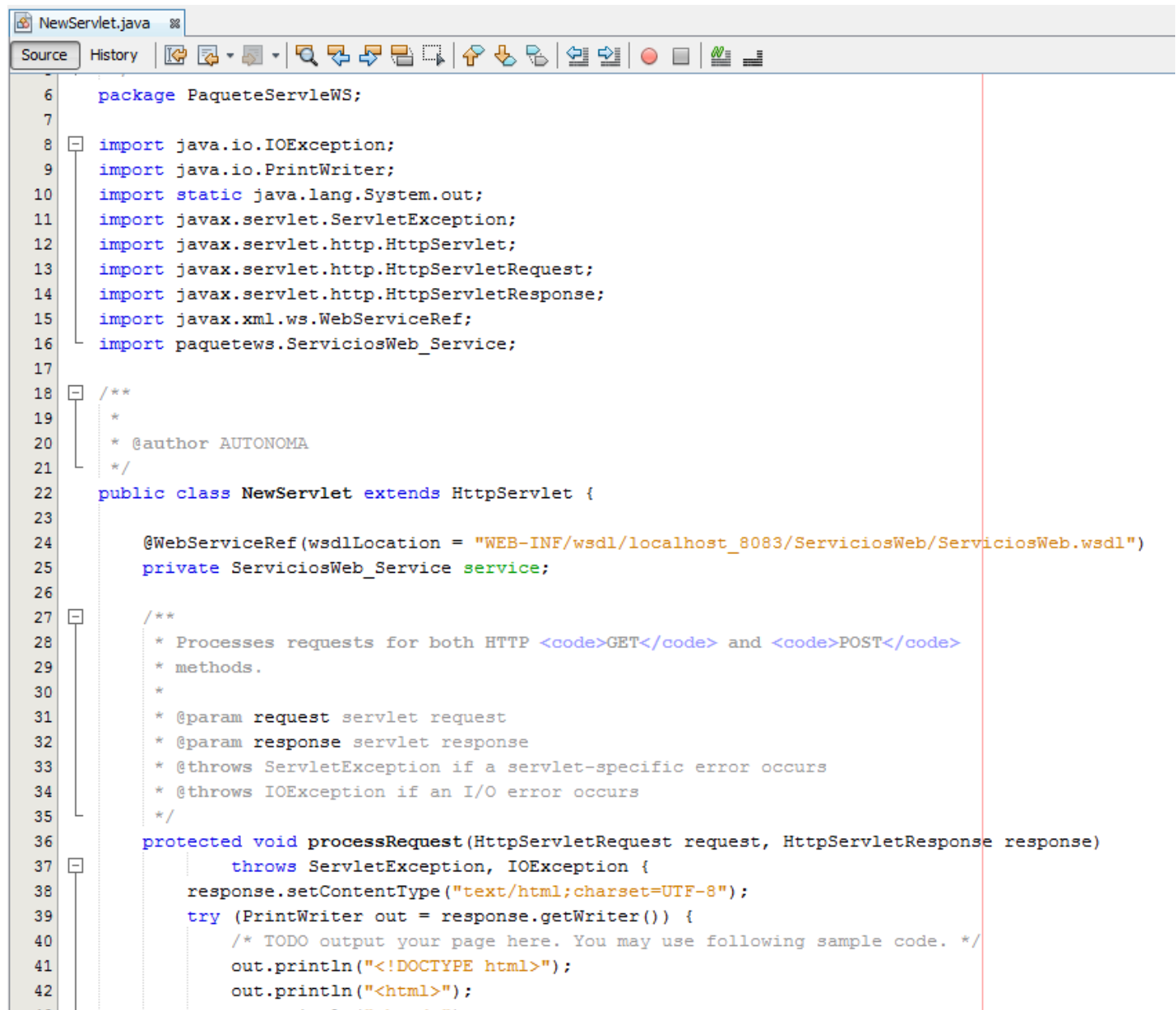
Operaciones					
CUADRADO		TRIANGULO		RECTANGULO	
Calcular el Lado y el Perimetro Ingresando:		Calcular el Perimetro y Area Ingresando:		Calcular la Base y el Perimetro Ingresando:	
Area:	<input type="text"/>	Base:	<input type="text"/>	Diagonal:	<input type="text"/>
Medida:	<input type="text" value="Metros"/>	Altura:	<input type="text"/>	Altura:	<input type="text"/>
	<input type="text"/>	Medida:	<input type="text" value="Metros"/>	Medida:	<input type="text" value="Metros"/>
<input type="button" value="Calcular operaciones"/>					

ALUMNO : VENTOCILLA SANCHEZ ADONIS

SERVLET NewServlet.java

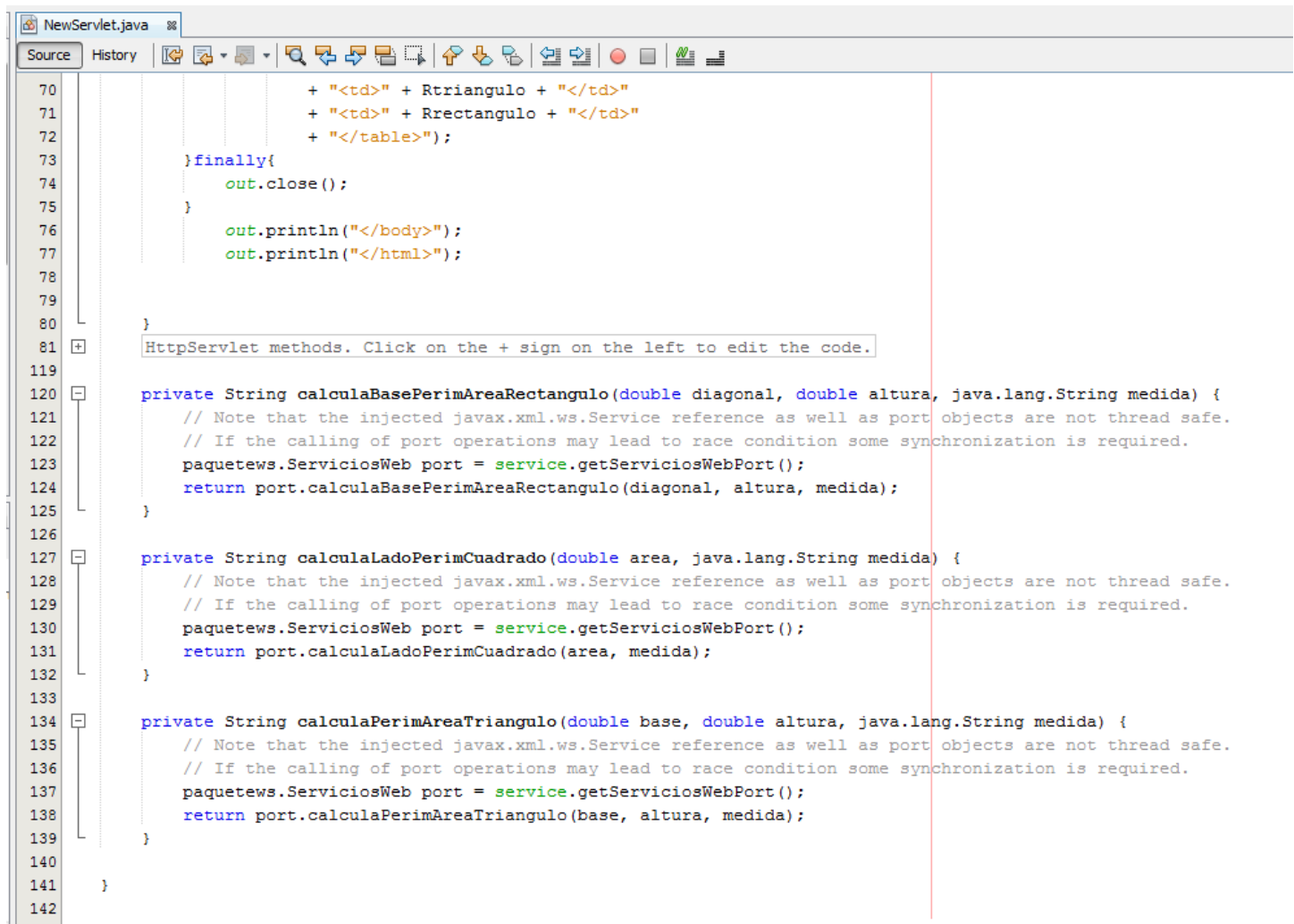
```
NewServlet.java
Source History
protected void processRequest(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
    response.setContentType("text/html;charset=UTF-8");
    try (PrintWriter out = response.getWriter()) {
        /* TODO output your page here. You may use following sample code. */
        out.println("<!DOCTYPE html>");
        out.println("<html>");
        out.println("<head>");
        out.println("<title>Servlet NewServlet</title>");
        out.println("</head>");
        out.println("<body>");
        //Recojo de los datos del formulario y conversion a double
        String areac = request.getParameter("areaC");
        double areaC = Double.parseDouble(areac);
        String baset = request.getParameter("baseT");
        double baseT = Double.parseDouble(baset);
        String diagonalr = request.getParameter("diagonalR");
        double diagonalR = Double.parseDouble(diagonalr);
        String alturat = request.getParameter("alturaT");
        double alturaT = Double.parseDouble(alturat);
        String alturar = request.getParameter("alturaR");
        double alturaR = Double.parseDouble(alturar);
        //
        //Obtencion de los resultados desde el Webservice
        String Rcuadrado = calculaLadoPerimCuadrado(areaC, "m");
        String Rtriangulo = calculaPerimAreaTriangulo(baseT, alturaT, "m");
        String Rrectangulo = calculaBasePerimAreaRectangulo(diagonalR, alturaR, "m");
        out.println("<table border>"
            + "<tr>"
            + "<th>CUADRADO</th>"
            + "<th>TRIANGULO</th>"
            + "<th>RECTANGULO</th>"
            + "</tr>"
            + "<td>" + Rcuadrado + "</td>"
            + "<td>" + Rtriangulo + "</td>"
            + "<td>" + Rrectangulo + "</td>"
            + "</table>");
    }
}
```

ALUMNO : VENTOCILLA SANCHEZ ADONIS



```
6 package PaqueteServleWS;
7
8 import java.io.IOException;
9 import java.io.PrintWriter;
10 import static java.lang.System.out;
11 import javax.servlet.ServletException;
12 import javax.servlet.http.HttpServlet;
13 import javax.servlet.http.HttpServletRequest;
14 import javax.servlet.http.HttpServletResponse;
15 import javax.xml.ws.WebServiceRef;
16 import paquetews.ServiciosWeb_Service;
17
18 /**
19  *
20  * @author AUTONOMA
21  */
22 public class NewServlet extends HttpServlet {
23
24     @WebServiceRef(wsdlLocation = "WEB-INF/wsdl/localhost_8083/ServiciosWeb/ServiciosWeb.wsdl")
25     private ServiciosWeb_Service service;
26
27     /**
28      * Processes requests for both HTTP <code>GET</code> and <code>POST</code>
29      * methods.
30      *
31      * @param request servlet request
32      * @param response servlet response
33      * @throws ServletException if a servlet-specific error occurs
34      * @throws IOException if an I/O error occurs
35      */
36     protected void processRequest(HttpServletRequest request, HttpServletResponse response)
37         throws ServletException, IOException {
38         response.setContentType("text/html;charset=UTF-8");
39         try (PrintWriter out = response.getWriter()) {
40             /* TODO output your page here. You may use following sample code. */
41             out.println("<!DOCTYPE html>");
42             out.println("<html>");
43             out.println("<head>");
```

ALUMNO : VENTOCILLA SANCHEZ ADONIS



```
70         + "<td>" + Rtriangulo + "</td>"
71         + "<td>" + Rrectangulo + "</td>"
72         + "</table>");
73     }finally{
74         out.close();
75     }
76     out.println("</body>");
77     out.println("</html>");
78
79
80 }
81
82 HttpServlet methods. Click on the + sign on the left to edit the code.
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120 private String calculaBasePerimAreaRectangulo(double diagonal, double altura, java.lang.String medida) {
121     // Note that the injected javax.xml.ws.Service reference as well as port objects are not thread safe.
122     // If the calling of port operations may lead to race condition some synchronization is required.
123     paquetews.ServiciosWeb port = service.getServiciosWebPort();
124     return port.calculaBasePerimAreaRectangulo(diagonal, altura, medida);
125 }
126
127 private String calculaLadoPerimCuadrado(double area, java.lang.String medida) {
128     // Note that the injected javax.xml.ws.Service reference as well as port objects are not thread safe.
129     // If the calling of port operations may lead to race condition some synchronization is required.
130     paquetews.ServiciosWeb port = service.getServiciosWebPort();
131     return port.calculaLadoPerimCuadrado(area, medida);
132 }
133
134 private String calculaPerimAreaTriangulo(double base, double altura, java.lang.String medida) {
135     // Note that the injected javax.xml.ws.Service reference as well as port objects are not thread safe.
136     // If the calling of port operations may lead to race condition some synchronization is required.
137     paquetews.ServiciosWeb port = service.getServiciosWebPort();
138     return port.calculaPerimAreaTriangulo(base, altura, medida);
139 }
140
141 }
142
```

Como se puede apreciar se ha capturado todas las variables de la primera pagina con el “request.getParameter()” y todas se han convertido en variable double para el uso de las funciones ya que por defecto se captura tipo String .

ALUMNO : VENTOCILLA SANCHEZ ADONIS

ENVIO DE DATOS

←

→

↺

🏠

📄 localhost:8083/ConsumeServicioWebE1_Ventocilla/ServletConsumeClient

👤 Universidad Autónoma

👤 Login Autónoma Virt

📄 COMPARTIDA

Operaciones		
CUADRADO	TRIANGULO	RECTANGULO
Calcular el Lado y el Perimetro Ingresando:	Calcular el Perimetro y Area Ingresando:	Calcular la Base y el Perimetro Ingresando:
Area: 1	Base: 2	Diagonal: 2
Medida: Metros ▾	Altura: 1	Altura: 1
	Medida: Metros ▾	Medida: Metros ▾
Calcular operaciones		

←

→

↺

🏠

📄 localhost:8083/ConsumeServicioWebE1_Ventocilla/NewServlet

👤 Universidad Autónoma

👤 Login Autónoma Virt

📄 COMPARTIDA

CUADRADO	TRIANGULO	RECTANGULO
El LADO del cuadrado es : 1.0 m y su PERIMETRO es : 4.0 m	El PERIMETRO del triangulo es : 4.82842712474619 m y su ÁREA es : 1.0 m	La BASE del rectangulo es : 1.7320508075688772 m su PERIMETRO es : 5.464101615137754 m y su ÁREA es : 1.7320508075688772 m²

*PEQUEÑA ACOTACION : EN EL ARCHIVO EN VE SDE DECIR “EP” DICE E1

CODIGO FUENTE: <https://github.com/AdonisVentocillaSanchez/EP-VENTOCILLA-WEBSERVICES.git>