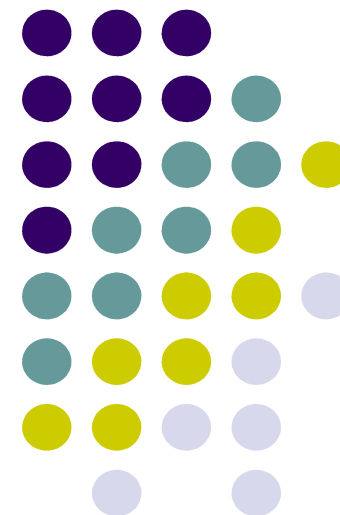
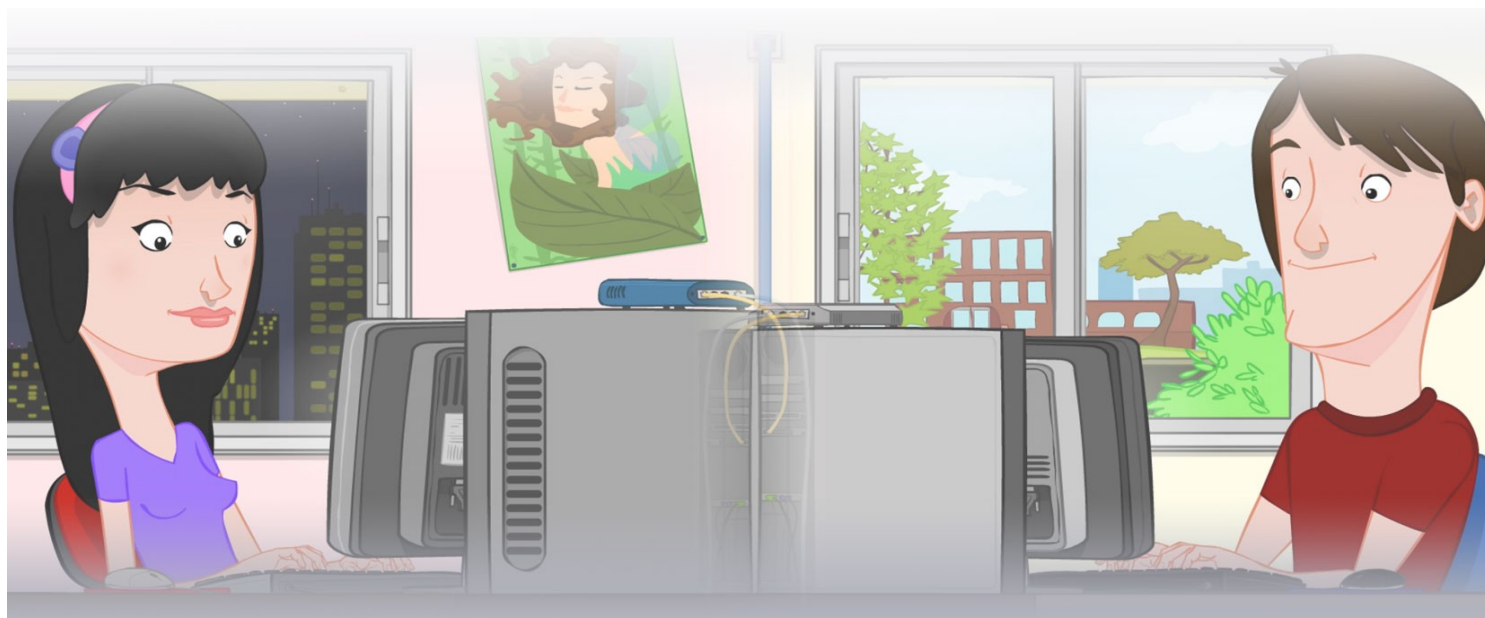


5.- AURKEZPENIK GABEKO WEB APLIKAZIOAK JAVAEE_n

Maidier Huarte Arrayago, Gorka Prieto Agujeta, Jasone Astorga Burgo, Nerea Toledo Gandarias
Komunikazioen Ingeniaritza Saila
Bilboko Ingeniaritza Goi Eskola Teknikoa
UPV/EHU



ZERBITZU TELEMATIKO AURRERATUAK:

5.- AURKEZPENIK GABEKO WEB APLIKAZIOAK JAVAEEEn



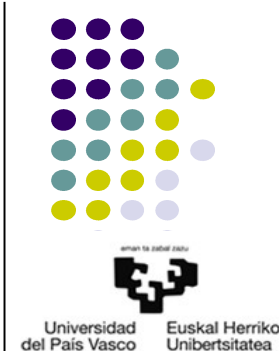
Copyright © 2015 Mainer Huarte Arrayago, Gorka Prieto Agujeta, Jasone Astorga Burgo, Nerea Toledo Gandarias

This is a derivative work from:

Mainer Huarte Arrayago, Gorka Prieto Agujeta

“Servicios Telemáticos Avanzados: 5.- Aplicaciones Web No Orientadas a Presentación con Java EE”

OCW UPV/EHU 2014 (ISSN 2255-2316), 2014



ZERBITZU TELEMATIKO AURRERATUAK: 5.- AURKEZPENIK GABEKO WEB APLIKAZIOAK JAVAEEEn lana, Mainer Huartek, Gorka Prietok, Jasone Astorga Burgok eta Nerea Toledo Gandariasek egina, Creative Commons-en Attribution-Share Alike 3.0 Unported License baimenaren menpe dago. Baimen horren kopia bat ikusteko, <http://creativecommons.org/licenses/by-sa/3.0/> webgunea bisitatu edo gutun bat bidali ondoko helbidera: Creative Commons, 171 2nd Street, Suite 300, San Francisco, California, 94105, USA.

Lan hau beste honen eratorria da: Mainer Huarte Arrayago, Gorka Prieto Agujeta, “Servicios Telemáticos Avanzados: 5.- Aplicaciones Web No Orientadas a Presentación con Java EE”, OCW UPV/EHU 2014 (ISSN 2255-2316), 2014

ZERBITZU TELEMATIKO AURRERATUAK: 5.- AURKEZPENIK GABEKO WEB APLIKAZIOAK JAVAEEEn by Mainer Huarte, Gorka Prieto, Jasone Astorga Burgo and Nerea Toledo Gandarias is licensed under a Creative Commons Attribution-Share Alike 3.0 Unported License. To view a copy of this license, visit <http://creativecommons.org/licenses/by-sa/3.0/> or, send a letter to Creative Commons, 171 2nd Street, Suite 300, San Francisco, California, 94105, USA.

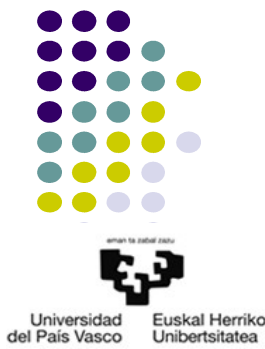
This is a derivative work from: Mainer Huarte Arrayago, Gorka Prieto Agujeta, “Servicios Telemáticos Avanzados: 5.- Aplicaciones Web No Orientadas a Presentación con Java EE” OCW UPV/EHU 2014 (ISSN 2255-2316), 2014

AURKEZPENIK GABEKO WEB APLIKAZIOAK

JAVAEEn

INDIZEA

- 1.- Sarrera
- 2.- REST motako Web Zerbitzuak
 - 2.1- Sarrera
 - 2.2- REST baliabideak JavaEErekin kodifikatzea
 - 2.3- JAX-RS bezeroa
- 3.- Aplikazioak konfiguratzea
 - 3.1- JAX-RS liburutegia
 - 3.2- Eclipse
- 4.- ERREFERENTZIAK



1.- Sarrera

- Zer da Web Zerbitzu bat?
 - JavaEEren definizioa
 - Aurkezpenerako web aplikazioekin ezberdintasunak
 - SOAren erlazioa
- Zer da SOA?
- Zer da SOA zerbitzu bat?

1.- Sarrera

- Web zerbitzu motak
 - Web zerbitzu “handiak”
 - SOAP
 - WSDL
 - Web zerbitzu “arinak”

2.- REST motako Web zerbitzuak

2.1.- Sarrera

- 2 mailatako sw arkitektura, egoerarik mantentzen ez duen protokoloarekin
- Baliabideak
- REST aplikazioetarako printzipioak
 - Baliabideak eta URLak
 - Baliabideekin lan egiteko eragiketak

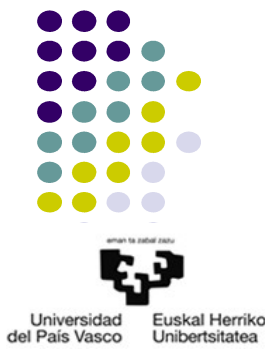
2.- REST motako Web zerbitzuak

2.2.- REST baliabideak JavaEerekin kodifikatzea

- REST baliabidea: POJO klasea+JAX-RS oharrak
- JAX-RS oharrak
 - @Path
 - @PathParam, @QueryParam
 - @GET, @POST, @PUT, @DELETE
 - @Consumes, @Produces
- Metodoak

2.- REST motako Web zerbitzuak

2.2.- REST baliabideak JavaEerekin kodifikatzea



- Oinarrizko adibidea

– URL: **`http://domeinualzerbitzua/rest/Ad1`**

```
//http://localhost:8080/proiektu_izena/rest/Ad1
```

Ad1.java

```
import javax.ws.rs.GET;
import javax.ws.rs.Path;
import javax.ws.rs.core.Response;

@Path("/Ej1")
public class Ej1 {

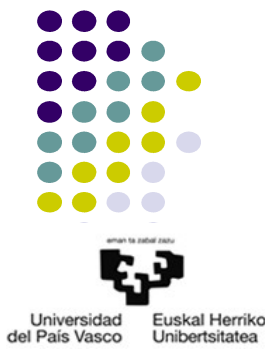
    @GET
    @Produces(MediaType.TEXT_PLAIN)
    public String ongietorriMezua() {

        String result = "KAIXO MUNDUA!";

        return result;
    }
}
```


2.- REST motako Web zerbitzuak

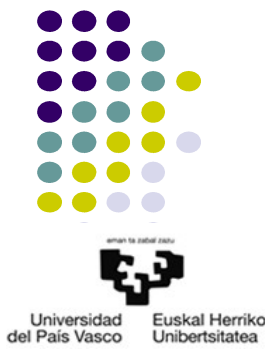
2.2.- REST baliabideak JavaEerekin kodifikatzea



- HTTP-request mezuetakoko URLa
 - **@Path**("URI_txantiloia")
 - Klase edo metodoetarako oharra
 - {aldagaia}
 - {aldagaia: *ohiko_espresioak*}
 - [a-zA-Z]*
 - [a-zA-Z][a-zA-Z_0-9]*
 - **@PathParam** *parametro_deklarazioa*
 - Metodoen parametroetarako oharra
 - **@QueryParam** *parametro_deklarazioa*
 - Metodoen parametroetarako oharra

2.- REST motako Web zerbitzuak

2.2.- REST baliabideak JavaEerekin kodifikatzea



- @PathParam adibidea

- URL: ***http://domeinualzerbitzualrest/Ad2/patroia/edozer1/edozer2***

```
import javax.ws.rs.GET;
import javax.ws.rs.Path;
import javax.ws.rs.PathParam;
import javax.ws.rs.core.Response;
```

```
@Path("/Ad2")
public class Ad2 {
```

```
    @GET
    @Path("/patroia/{p1}/{p2}")
    public Response erantzunMezua(@PathParam("p1") String parametro1,@PathParam("p2") String parametro2) {
```

```
        String mezua="URLan jasotako balioak hauek izan dira: p1="+parametro1+", p2="+parametro2;
```

```
        return Response.status(200).entity(mezua).build();
```

```
    }
}
```

Ej2

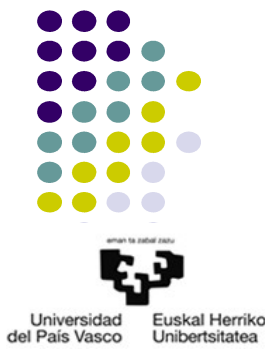
2.- REST motako Web zerbitzuak

2.2.- REST baliabideak JavaEerekin kodifikatzea

- HTTP-request mezueta metodoa
 - **@GET, @POST, @PUT, @DELETE**
 - Metodoetarako oharra
 - Atzigarritasuna
 - Metodoaren itzulera balioa: return mota
 - void
 - Response
 - JB
 - Parametroak
 - @PathParam
 - @QueryParam: @GET

2.- REST motako Web zerbitzuak

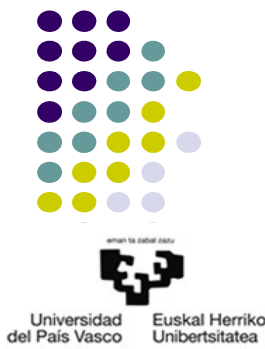
2.2.- REST baliabideak JavaEerekin kodifikatzea



- HTTP-request mezueta beste informazioa
 - **@RequestParam**("datuaren_izena")
 - @POST
 - @Consumes("application/x-www-form-urlencoded")
 - **@CookieParam**
 - **@HeaderParam**

2.- REST motako Web zerbitzuak

2.2.- REST baliabideak JavaEerekin kodifikatzea



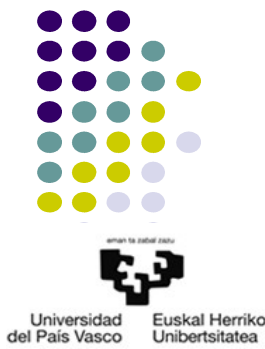
- Formularioen adibidea

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>Formularioak dituen orrialdea, frogak egiteko</title>
</head>
<body>
  <h1>Formularioak dituen orrialdea, frogak egiteko</h1>
  <form action="rest/Ad3/GETformularioa" method="get">
    <p><h2>GET FORMULARIOA</h2>
      <input type="text" name="er1" />
      <select name="er">
        <option value="+">+</option>
        <option value="-">-</option>
        <option value="*">*</option>
        <option value="/">/</option>
      </select>
      <input type="text" name="er2" />
    </p>
    <input type="submit" value="KALKULATU" />
  </form>
  <form action="rest/Ad3/POSTformularioa" method="post">
    <p><h2>POST FORMULARIOA</h2>
  <!-- Aurreko formularioan bezala -->
    </p>
    <input type="submit" value="KALKULATU" />
  </form>
</body>
</html>
```

Formularioak.html

2.- REST motako Web zerbitzuak

2.2.- REST baliabideak JavaEerekin kodifikatzea



- Formularioen adibidea

```
import javax.ws.rs.GET;
import javax.ws.rs.POST;
import javax.ws.rs.Path;
import javax.ws.rs.DefaultValue;
import javax.ws.rs.QueryParam;
import javax.ws.rs.FormParam;
import javax.ws.rs.core.Response;

@Path("/Ad3")
public class Ad3 {
    @GET
    @Path("/GETformularioa")
    public Response GETmetodoa(@QueryParam("er1") float er1,@QueryParam("er") String er,@QueryParam("er2") float er2) {

        String mezua="Formularioan adierazitako eragiketa hau izan da: "+er1+er+er2;

        return Response.status(200).entity(mezua).build();
    }

    @POST
    @Path("/POSTformularioa")
    public Response POSTmetodoa(@FormParam("er1") float er1,@FormParam("er") String er,@FormParam("er2") float er2) {

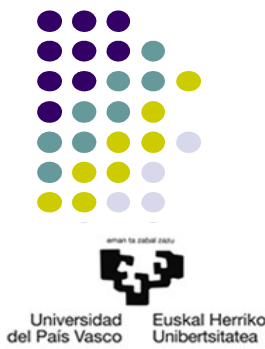
        String mezua="Formularioan adierazitako eragiketa hau izan da: "+er1+er+er2;

        return Response.status(200).entity(mezua).build();
    }
}
```

Ad3.java

2.- REST motako Web zerbitzuak

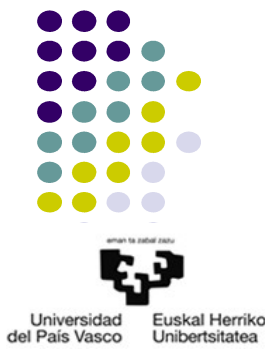
2.2.- REST baliabideak JavaEErekin kodifikatzea



- Ekoiztutako eta kontsumitutako MIME motak
 - **@Produces("MIME_mota")**
 - Klase edo metodoetarako oharra
 - Mota anitz
 - *Error HTTP 406 Not Acceptable*
 - **@Consumes("MIME_mota")**
 - Klase edo metodoetarako oharra
 - Mota anitz
 - *Error HTTP 415 Unsupported Media Type*

2.- REST motako Web zerbitzuak

2.2.- REST baliabideak JavaEerekin kodifikatzea



- MIME adibidea

```
import java.io.File;
import javax.ws.rs.GET;
import javax.ws.rs.Path;
import javax.ws.rs.Produces;
import javax.ws.rs.core.Response;
import javax.ws.rs.core.Response.ResponseBuilder;

@Path("/Ad4")
public class Ad4 {

    private static final String FILE = "/home/zta1314/Documents/frogetarakoFitxategia.pdf";

    @GET
    @Produces("application/pdf")
    public Response getFile() {

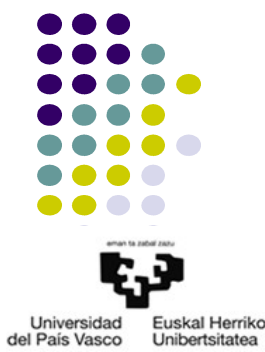
        File file = new File(FILE);

        ResponseBuilder response = Response.ok((Object) file);
        response.header("Content-Disposition", "attachment; filename="+FILE);
        return response.build();
    }
}
```

Ad4.java

2.- REST motako Web zerbitzuak

2.3.-JAX-RS bezeroa



- Pausoak

- Bezero objektua sortu

```
Client c= ClientBuilder.newClient();
```

- Target parametroa ezarri

```
WebTarget wt=c.target("REST_URL");
```

- Request sortu

```
Invocation.Builder b=wt.request(MediaType.TEXT_PLAIN);
```

- Request bidali:

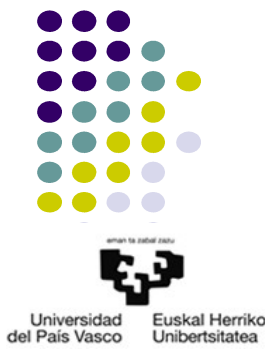
```
b.get(), b.post(), b.delete(), b.put(), ...
```

- Guztia pauso bakarrean

```
ClientBuilder.newClient().target("REST_URL").request(MediaType.TEXT_PLAIN).get()
```

2.- REST motako Web zerbitzuak

2.3.-JAX-RS bezeroa



- get() adibidea: @Produces(MediaType.APPLICATION_XML) zerbitzu baterako

```
import javax.ws.rs.client.ClientBuilder;
import javax.ws.rs.client.Entity;
import javax.ws.rs.core.MediaType;

...

JavaBean bean = ClientBuilder.newClient()
    .target("http://localhost:8080/P5/rest/denda/getProduktuZerrenda")
    .request(MediaType.APPLICATION_XML)
    .get(JavaBean.class);
```

- post() adibidea: @Consumes(MediaType.APPLICATION_XML) zerbitzu baterako

```
import javax.ws.rs.client.ClientBuilder;
import javax.ws.rs.client.Entity;
import javax.ws.rs.core.MediaType;

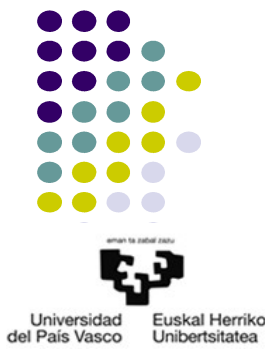
...

ClientBuilder.newClient()
    .target("http://localhost:8080/P5/rest/denda/gordeProduktua")
    .request(MediaType.APPLICATION_XML)
    .post(Entity.entity(bean, MediaType.APPLICATION_XML));

//bean: post mezuan bidali nahi diren datuak biltzen dituen JavaBean objektua
```

2.- REST motako Web zerbitzuak

2.3.-JAX-RS bezeroa



- JAX-RS bezeroaren adibidea

```
//AURREKO GAIKO AZKEN ADIBIDEKO BERDINA
```

EragiketakoDatuak.java

```
//AURREKO GAIKO AZKEN ADIBIDEKO BERDINA
```

EragiketenZerrenda.java

```
//AURREKO GAIKO AZKEN ADIBIDEKO BERDINA
```

DatuenBiltegitratzea.java

```
package logika;

import java.util.List;

import datuak.DatuenBiltegitratzea;
import datuak.EragiketakoDatuak;
import datuak.EragiketenZerrenda;

public class ZerrendakoLogika {
    public static EragiketenZerrenda zerrendaXMLIrakurri() {
        return DatuenBiltegitratzea.zerrendaXMLIrakurri();
    }

    public static void eragiketaBerriaXMLIdatzi(EragiketakoDatuak eD) {
        EragiketenZerrenda eZ=DatuenBiltegitratzea.zerrendaXMLIrakurri();

        List<EragiketakoDatuak> zerrenda=eZ.getErZerr();
        zerrenda.add(eD);
        eZ.setErZerr(zerrenda);

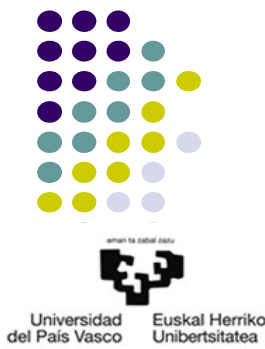
        DatuenBiltegitratzea.zerrendaXMLIdatzi(eZ);
    }
}
```

ZerrendakoLogika.java

2.- REST motako Web zerbitzuak

2.3.-JAX-RS bezeroa

- JAX-RS bezeroaren adibidea



```
package logika;

//BEHARREZKO import LERROAK

@Path("/AplikaziokoLogika")
public class AplikaziokoLogika
{
    @POST
    @Consumes("application/xml")
    @Produces("application/xml")
    public static EragiketakoDatuak kalkulatu(EragiketakoDatuak Ed) {

        float er1=eD.getEr1();
        char er=eD.getEr();
        float er2=eD.getEr2();
        float em;

        switch(er)
        {
            case '+':
                em=er1+er2;
                break;

//BESTE ERAGIKETETAKO case GUZTIAK

                default:
                    em=0.0f;
            }

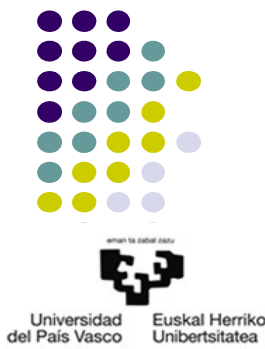
        eD.setEm(em);

        return eD;
    }
}
```

EragiketakoDatuak.java

2.- REST motako Web zerbitzuak

2.3.-JAX-RS bezeroa



- JAX-RS bezeroaren adibidea

```
//AURREKO GAIKO AZKEN ADIBIDEKO BERDINA
```

ServletErab.java

```
package aurkezpena;
```

FormularioaJasoEtaEmitzaSortzekoServlet.java

```
//BEHARREZKO import LERROAK
```

```
@WebServlet("/FormularioaJasoEtaEmitzaSortzekoServlet")
```

```
public class FormularioaJasoEtaEmitzaSortzekoServlet extends HttpServlet
```

```
{
```

```
    private static final long serialVersionUID = 1L;
```

```
    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
```

```
//AURREKO GAIKO AZKEN ADIBIDEKO BERDINA
```

```
}
```

```
    protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
```

```
        String sEr1=request.getParameter("eragigai1");
```

```
        String sEr=request.getParameter("eragiketa");
```

```
        String sEr2=request.getParameter("eragigai2");
```

```
        EragiketakoDatuk eD=new EragiketakoDatuk(sEr1,sEr,sEr2);
```

```
        eD=ClientBuilder.newClient()
```

```
            .target("http://localhost:8080/T5-4AD_RESTWS_4/rest/AplikaziokoLogika")
```

```
            .request("application/xml").post(Entity.entity(eD, MediaType.APPLICATION_XML),EragiketakoDatuk.class);
```

```
        ZerrendakoLogika.eragiketaBerriaXMLIdatzi(eD);
```

```
//AURREKO GAIKO AZKEN ADIBIDEKO LERRO BERDINAK
```

```
}
```

```
}
```

```
//AURREKO GAIKO AZKEN ADIBIDEKO BERDINA
```

Formularioa.html

```
//AURREKO GAIKO AZKEN ADIBIDEKO BERDINA
```

aplikAritEstiloak.css

3.- Aplikazioak konfiguratzea

3.1- JAX-RS liburutegia

- JAX-RS implementazioak
 - Jersey (Oracle), RESTeasy (JBoss), etab.
 - Java EE 6ik aurrera, integratua
- Java EE ez diren inguruneetarako
 - Liburutegia deskargatu
 - Javako WS kklaseak pakete ierarkia baten barruan egon behar dira
 - Aplikazioak web.xml fitxategia eduki behar du

```
<servlet>
  <servlet-name>Jersey REST Service</servlet-name>
  <servlet-class>org.glassfish.jersey.servlet.ServletContainer</servlet-class>
  <init-param>
    <param-name>jersey.config.server.provider.packages</param-name>
    <param-value>Garatutako_WS_klaseen_paketearen_izena</param-value>
  </init-param>
  <load-on-startup>1</load-on-startup>
</servlet>

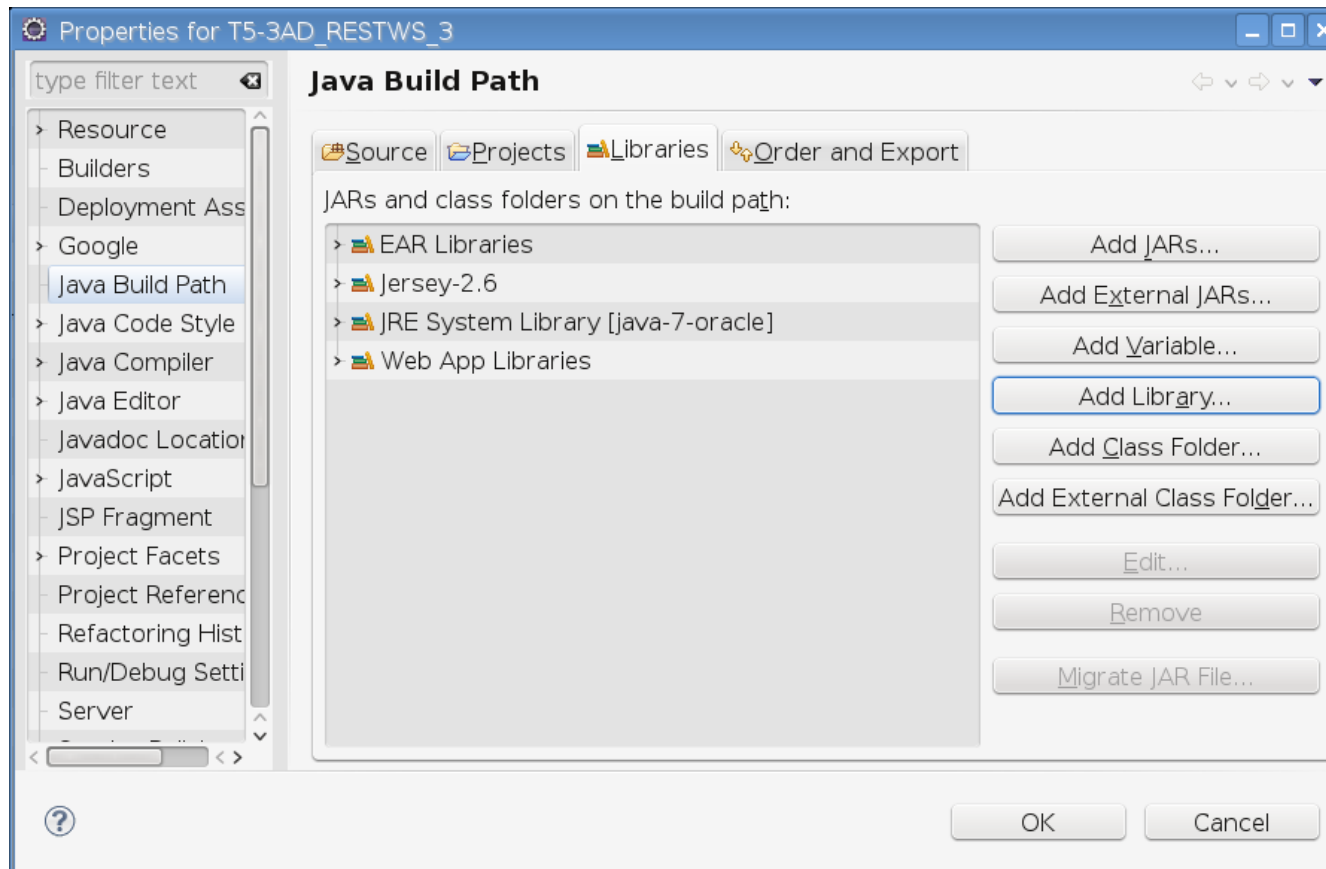
<servlet-mapping>
  <servlet-name>Jersey REST Service</servlet-name>
  <url-pattern>/rest/*</url-pattern>
</servlet-mapping>
```

web.xml

3.- Aplikazioak konfiguratzeko

3.2- Eclipse

- Liburutegia erreferentziatzea, beharrezkoa
- Build Path: User Library (berria)



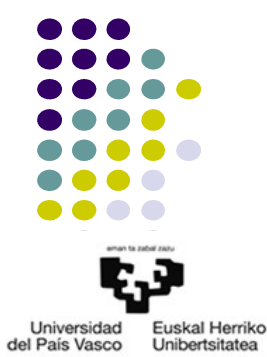
3.- Aplikazioak konfiguratzera

3.2- Eclipse

- Web Deployment Assembly



4.- ERREFERENTZIAK



- Java Platform, Enterprise Edition (Java EE) 7
 - Web Services: <http://docs.oracle.com/javaee/7/tutorial/partwebsvcs.htm#BNAYK>