

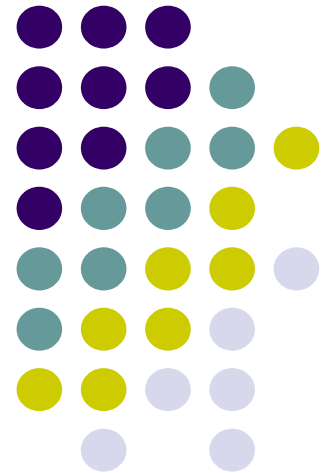
4.- DATUEN ADIERAZPENA

ESKEMAK

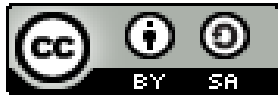
27380 ZERBITZU TELEMATIKO AURRERATUAK
Telekomunikazio Teknologiako Ingeniaritza Gradua, 3. maila

2023-2024

Egileak: Maider Huarte Arrayago, Gorka Prieto Agujeta



2023-2024 ZTA 4 DATUEN ADIERAZPENA - ESKEMAK.odp



Copyright © 2013-2024 Maider Huarte Arrayago, Gorka Prieto Agujeta



2023-2024 ZTA 4 DATUEN ADIERAZPENA - ESKEMAK.odp lana, Maider Huartek eta Gorka Prietok eginia, Creative Commons-en Attribution-Share Alike 3.0 Unported License baimenaren menpe dago. Baimen horren keria 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.

2023-2024 ZTA 4 DATUEN ADIERAZPENA - ESKEMAK.odp by Maider Huarte and Gorka Prieto 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.

WEB ORRIALDE DINAMIKOAK

EDUKIA

1.- SARRERA

2.- XML

3.- JSON

ERREFERENTZIAK

WEB ORRIALDE DINAMIKOAK

EDUKIA

1.- SARRERA

2.- XML

3.- JSON

ERREFERENTZIAK

1.- SARRERA

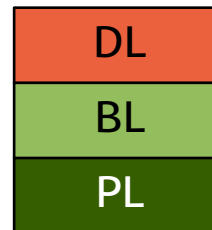
- Aplikazioen artean datuak mugitzea Internet bidez
 - HTTP
 - HTML
 - XML
 - JSON
 - 3 mailetako sw arkitektura jarraitzen duten aplikazioetan
 - DL: biltegitratutako datuen adierazpena
 - BL, PL: datuak ekipo ezberdinen artean garraiatzeko

Giza-erabiltzaileei datuak bistaratzeko

Aplikazioei datuak emateko

Biltegitik memoriara

Memoriatik biltegira



WEB ORRIALDE DINAMIKOAK

EDUKIA

1.- SARRERA

2.- XML

2.1.- Sarrera

2.2.- Oinarrizko XML

2.3.- JAXB

3.- JSON

ERREFERENTZIAK

2.- XML

2.1.- Sarrera

- Zer da XML?
 - eXtensible Markup Language
 - Oso kapaza
Self explained
 - CSV: Biltegitratzeko
 - Oso sinplea
 - Ez "self explained"
 - Egituratu
 - Garraiatu eta/edo biltegitratu
 - W3C
 - Erregulatzailerak erakundeak
 - Hw eta sw kontzeptuetatik independentea
 - SE guztiek XML ulertzen dute berdin
- Fitxategien atzizkia: **.xml**
- MIME mota: **application/xml**
- Erabilpen adibideak
 - Client-Side Scripting: AJAX
 - Server-Side Scripting
 - Datu zerbitzarian biltegitratzeko: DL
 - Datuak zerbitzarien artean garraiatzeko: PL-BL

2.- XML

2.1.- Sarrera

- XMLn oinarritutako teknologiak
 - xhtml
 - JSFn erabilia
 - HTML-ren bertsio zorrotzagoa, XML-n oinarritua
 - XSD: xml Schema
 - SOAP
 - WSDL
 - ...

WEB ORRIALDE DINAMIKOAK

EDUKIA

1.- SARRERA

2.- XML

2.1.- Sarrera

2.2.- Oinarrizko XML

2.3.- JAXB

3.- JSON

ERREFERENTZIAK

2.- XML

2.2.- Oinarrizko XML

- XML dokumentuak
 - Testu sinplea
 - XML deklarazioa, DOCTYPE deklarazioa
 - Elementu egituratuak: etiketak
 - Ez dago lengoaian berez definitutako etiketarik

- Etiketak definitzeko askatasuna- Egitura definitzeko askatasuna - Lehen

2.- XML

2.2.- Oinarrizko XML

- XML sintaxia
 - Iruzkinak: `<!-- lerroAnitzekoIruzkina -->`
 - Etiketak: Irekiera `<tag>`, itxiera `</tag>`
 - XML deklarazioa eta **DOCTYPE** Itxiera etiketa ez
 - Tag hitzak xeheak, LARRIAK -> irekieran == itxieran (CASE SENSITIVE DA!!)
 - Elementuak elkarren artean txertatzea
 - Erro-elementua Bakarra!
 - Elementuen atributuak: `izena="balioa"`

2.- XML

2.2.- Oinarrizko XML



- XML sintaxia
 - Karaktere bereziak Datuen balioetan suertatzen badira, nola ordezkatu?

KARAKTEREA	ORDEZKAPENA	ARRAZOIA
<	< less than	Etiketak mugatzeko erabiltzen delako
>	> greater than	Etiketak mugatzeko erabiltzen delako
“	"	Atributuen balioak mugatzeko erabiltzen delako
'	'	Atributuen balioak mugatzeko erabiltzen delako
&	& ampersand	Karaktereen ordezkapenetan erabiltzen delako

2.- XML

2.2.- Oinarritzko XML

- ADIBIDEA: Oinarritzko XML fitxategiak

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE sesioa SYSTEM "Sesioa.dtd">
<sesioa idSesioa="1" idErab="25">
  <eragiketa>
    <er1>12.0</er1>
    <er>-</er>
    <er2>2.0</er2>
    <em>10.0</em>
  </eragiketa>
  <eragiketa>
    <er1>34.0</er1>
    <er>+</er>
    <er2>5.0</er2>
    <em>39.0</em>
  </eragiketa>
</sesioa>
```

AritmetikakoSesioa1.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE sesioa SYSTEM "froga3.dtd">
<sesioa>
  <idSesioa>1</idSesioa>
  <idErab>25</idErab>
  <eragiketa>
    <er1>12.0</er1>
    <er>-</er>
    <er2>2.0</er2>
    <em>10.0</em>
  </eragiketa>
  <eragiketa>
    <er1>34.0</er1>
    <er>+</er>
    <er2>5.0</er2>
    <em>39.0</em>
  </eragiketa>
</sesioa>
```

AritmetikakoSesioa2.xml

Informazio berdina!
Egitura ezberdina!



2.- XML

2.2.- Oinarrizko XML

- Etiketetarako izenak DERRIGORREZKO ARAUAK
 - Letrak, zenbakiak, etab.
 - Zenbakiak, puntuazio karaktereak 1. karakterea LETRA IZAN BEHAR DA
 - Zuriuneak
 - "xml", "Xml", etab. } EZ
- Etiketen izenetarako gomendioak GOMENDIOAK
 - Izen esanguratsuak: '_' edo *xeheLARRI* camelCase, PascalCase
 - Laburrak eta sinpleak
 - Ez erabili q, q, q ...

2.- XML

2.2.- Oinarrizko XML

- Txertatzea vs atributuak

Txertatzea hobesten da

- Balio anitz
- Balio egituratuak
- Hedagarria

Atributuekin EZ

- Elementuen identifikazioa
- Konbinazioa

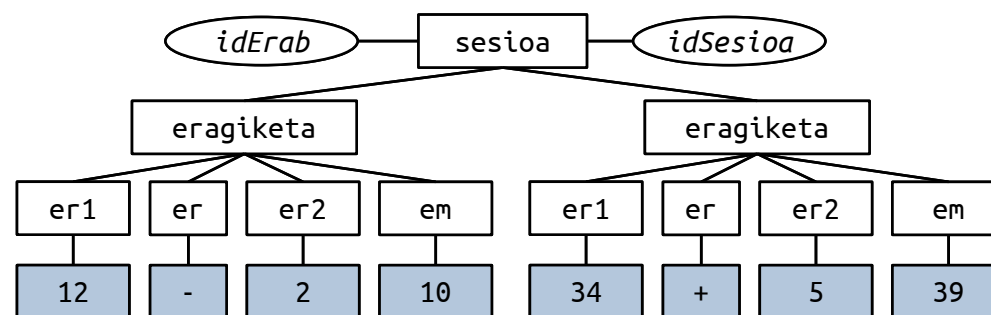
SALBUESPENA: id (eta antzekoak) Atributu moduan

- XML eta DOM

- Erro-elementua: zuhaitza
- Alabak, amak
- Ahizpak

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE sesioa SYSTEM "Sesioa.dtd">
<sesioa idSesioa="1" idErab="25">
  <eragiketa>
    <er1>12.0</er1>
    <er>-</er>
    <er2>2.0</er2>
    <em>10.0</em>
  </eragiketa>
  <eragiketa>
    <er1>34.0</er1>
    <er>+</er>
    <er2>5.0</er2>
    <em>39.0</em>
  </eragiketa>
</sesioa>
```

AritmetikakoSesioa1.xml



2.- XML

2.2.- Oinarrizko XML

- XML Namespaces Atributuak
 - Izenen errepikapen arazoak
 - Deklarazioa: `xmlns[:aurrizkia]="URL"`
 - Aurrizkiak Bakarrak XML fitxategian
 - URL Informazio gehigarria, EZ EGIAZTAPENETARAKO
 - xmlns deklarazioen kokapena
 - Erro-elementua Aurrizkia derrigorrezkoa, errepikapenak daudenean
 - Beste elementuak Aurrizkia aukerazkoa

2.- XML

2.2.- Oinarrizko XML

- ADIBIDEA: NameSpaces

```
<root xmlns:h="http://www.w3.org/TR/html4/" xmlns:f="http://www.w3schools.com/furniture">      Root elementuan ez

<h:table>
  <h:tr>
    <h:td>Sagararak</h:td>
    <h:td>Udareak</h:td>
  </h:tr>
</h:table>

<f:table>
  <f:name>Sukaldeko mahaia</f:name>
  <f:width>80</f:width>
  <f:length>120</f:length>
</f:table>

</root>
```

Aurrizkiak derrigorrez agertuko

TableAdibidea.xml

```
<table xmlns="http://www.w3.org/TR/html4/">
  <tr>
    <td>Sagararak</td>
    <td>Udareak</td>
  </tr>
</table>
```

Taula.xml

```
<table xmlns="http://www.w3schools.com/furniture">
  <name>Sukaldeko mahaia</name>
  <width>80</width>
  <length>120</length>
</table>
```

Mahaia.xml



2.- XML

2.2.- Oinarrizko XML

- Ondo osatua vs baliagarria

- Ondo osatua XML sintaxia

- Baliagarria Egitura ez-librea

Baliagarria ---> Ondo osatua
Ondo osatua -/-> Baliagarria

- Balidatze-patroia

- DTD

- XML Schema

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE sesioa SYSTEM "froga3.dtd">
<sesioa>
  <idSesioa>1</idSesioa>
  <idErab>25</idErab>
  <eragiketa>
    <er1>12.0</er1>
    <er>--</er>
    <er2>2.0</er2>
    <em>10.0</em>
  </eragiketa>
  <eragiketa>
    <er1>34.0</er1>
    <er>+</er>
    <er2>5.0</er2>
    <em>39.0</em>
  </eragiketa>
</sesioa>
```

AritmetikakoSesioa2.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<persistence version="2.1" xmlns="http://xmlns.jcp.org/xml/ns/persistence" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/persistence http://xmlns.jcp.org/xml/ns/persistence/persistence_2_1.xsd">
  <persistence-unit name="T08-1AD_JSF-EJB-JPA_1" transaction-type="JTA">
    <jta-data-source>java:jboss/datasources/ZTA_T08-1AD</jta-data-source>
    <class>dl.EragiketaE</class>
    <properties>
      <property name="hibernate.dialect" value="org.hibernate.dialect.MySQLDialect" />
      <property name="hibernate.hbm2ddl.auto" value="validate" />
    </properties>
  </persistence-unit>
</persistence>
```

persistence.xml

WEB ORRIALDE DINAMIKOAK

EDUKIA

1.- SARRERA

2.- XML

2.1.- Sarrera

2.2.- Oinarrizko XML

2.3.- JAXB

3.- JSON

ERREFERENTZIAK

2.- XML

2.3.- JAXB

- Java Architecture for XML Binding (JAXB)
 - Java klaseak XML formatura pasatzea eta alderantziz: mapeatzea
 - Oharrak + marshaller/unmarshaller objektuak
 - Oracle Java6tik aurrera
 - Aplikazio Zerbitzari arruntetan: EclipseLink liburutegia
- Prozedura
 - XML formaturako datuak: **JavaBeans + oharrak**
 - **@XmlElement, @XmlType** JB klaserako oharra
 - **@XmlElement, @XmlAttribute** Atributuetarako oharrak (getter-etan)
 - XMLren etiketarako izena
 - **@XmlElement(name="XML izena")**
 - **@XmlElement(name="XML izena")**

2.- XML

2.3.- JAXB

- Prozedura

- Mapeatzeko kodea idatzi

- JAXB testuingurua lortu

- testuingurua=JAXBContext.newInstance(XMLrootKlasea.class);*

- Testuingurutik marshaller-ak lortu

- marshaller=testuingurua.createMarshaller();* *JavaBean -> XML*

- unmarshaller=testuingurua.createUnmarshaller();* *XML -> JavaBean*

- Marshaller-arekin JavaBean objektutik XMLra pasa edo XMLtik JavaBean objektura

- marshaller.marshal(objektua, XMLfitxategia)*

- unmarshaller.unmarshal(XMLfitxategia)*

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE sesioa SYSTEM "Sesioa.dtd">
<sesioa idSesioa="1" idErab="25">
  <eragiketa>
    <er1>12.0</er1>
    <er>-</er>
    <er2>2.0</er2>
    <em>10.0</em>
  </eragiketa>
  <eragiketa>
    <er1>34.0</er1>
    <er>+</er>
    <er2>5.0</er2>
    <em>39.0</em>
  </eragiketa>
</sesioa>
```

AritmetikakoSesioa1.xml

XML-n txertatutako elementu bakoitza, JavaBean ezberdin bat.

XML root: JavaBean bakarra

2.- XML

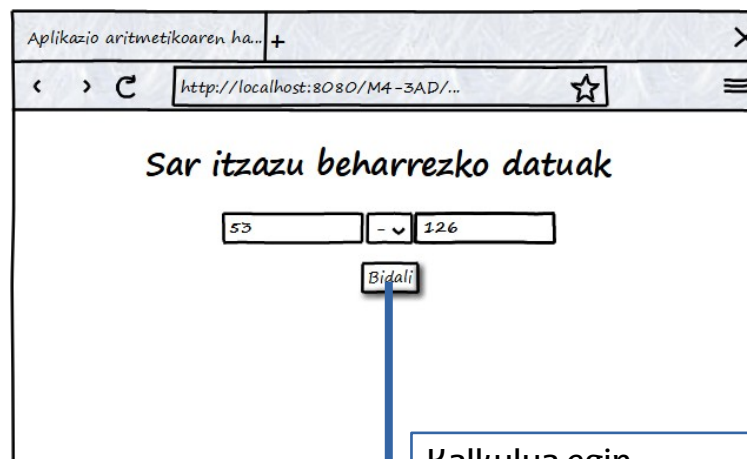
2.3.- JAXB

- ADIBIDEA: JAXB

1. pantaila: Hasierako egoera



1. pantaila: Erabiltzaileak formularioa beteta

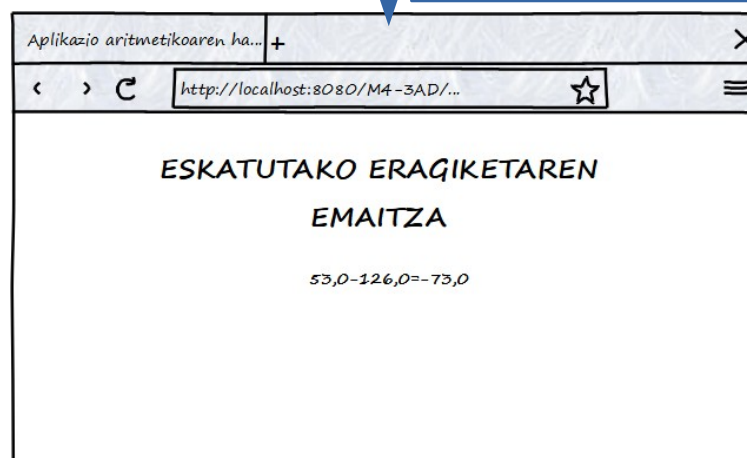


Zerbitzariko Web Aplikazioan estatikoki programatutako HTML

Kalkulua egin
Eragiketa XML fitxategi batean sartu
2. pantailara

Web Aplikazioa

2. pantaila: Eraitza















Zerbitzariko Web Aplikazioaren kodeak automatikoki sortutako HTML

2.- XML



2.3.- JAXB

- ADIBIDEA: JAXB

- JAVA osagaiak: **java**

- ▼  bl
 - ▶  EragiketaJB.java
 - ▶  KalkulagailuaPOJO.java
 - ▶  ZerbitzuakPOJO.java
- ▼  dl
 - ▶  EragiketaBXML.java
 - ▶  EragiketenTaulaXML.java
 - ▶  EragiketenWBXML.java
 - ▶  PaketekoKonstanteak.java
- ▼  pl
 - ▶  ErabiltzaileServlet.java
 - ▶  TxantiloiaPOJO.java

- HTML osagaiak: **webapp**

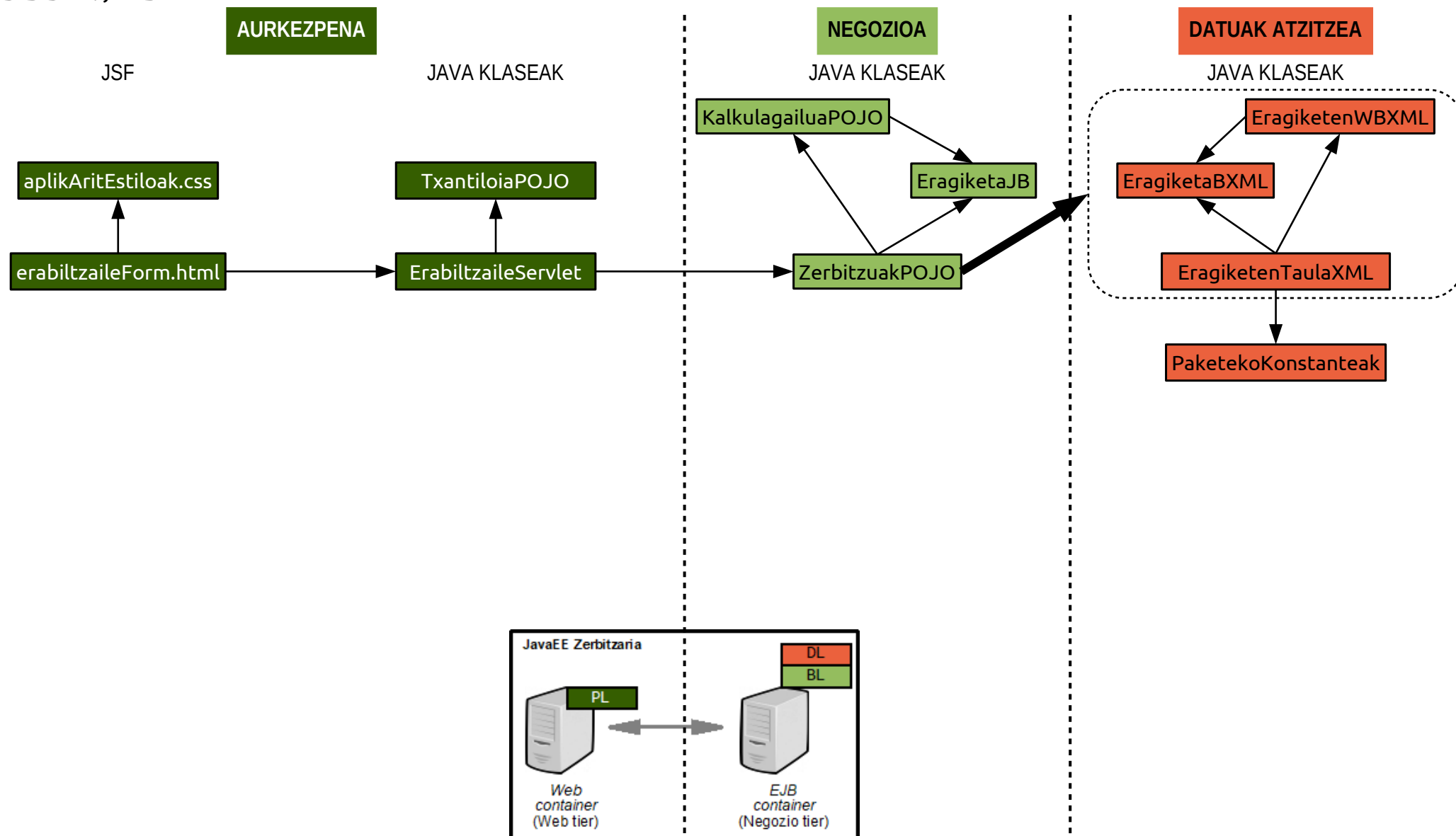
- ▼  EDUKIAK
 - ▶  aplikAritEstiloak.css
 - ▶  erabiltzaileForm.html



2.- XML

2.3.- JAXB

- ADIBIDEA: JAXB



2.- XML

2.3.- JAXB

- ADIBIDEA: JAXB

XML biltegrirako JavaBean: dl.EragiketaBXML.java

```
package dl;

//import lerroak

@XmlRootElement(name="eragiketa")
@XmlType(propOrder={"eragigai1", "eragilea", "eragigai2", "emaitza", "dataOrdua"})
public class EragiketaBXML {
    private float eragigai1;
    private String eragilea;
    private float eragigai2;
    private float emaitza;
    private String dataOrdua;

    public EragiketaBXML() {}

    public EragiketaBXML(float eragigai1, String eragilea, float eragigai2, float emaitza, String dataOrdua) {
        this.eragigai1=eragigai1;
        this.eragilea=eragilea;
        this.eragigai2=eragigai2;
        this.emaitza=emaitza;
        this.dataOrdua=dataOrdua;
    }

    @XmlElement
    public float getEragigai1() {
        return eragigai1;
    }

    public void setEragigai1(float eragigai1) {
        this.eragigai1 = eragigai1;
    }

    //Beste atributuen getter eta setter metodoak

    public String laburpena() {
        return ""+eragigai1+eragilea+eragigai2+"="+emaitza;
    }
}
```

2.- XML

2.3.- JAXB

- ADIBIDEA: JAXB

XML datuetarako JavaBean: dl.EragiketenWBXML.java

```
package dl;

import java.util.ArrayList;
import java.util.List;

import javax.xml.bind.annotation.XmlElement;
import javax.xml.bind.annotation.XmlRootElement;

@XmlRootElement(name="eragiketenTaula")
public class EragiketenWBXML {
    private List<EragiketaBXML> eragiketakBXML=new ArrayList<EragiketaBXML>();

    public EragiketenWBXML() {
    }

    @XmlElement(name="eragiketa")
    public List<EragiketaBXML> getEragiketakBXML() {
        return eragiketakBXML;
    }

    public void setEragiketakBXML(List<EragiketaBXML> eragiketakBXML) {
        this.eragiketakBXML=eragiketakBXML;
    }
}
```

2.- XML

2.3.- JAXB

- ADIBIDEA: JAXB

dl.EragiketenTaulaXML.java

```
package dl;

//import lerroak

public class EragiketenTaulaXML {
    private List<EragiketaBXML> guztiakIrakurri() {
        EragiketenWBXML eWBXML;
        File xmlFitxategia=new File(PaketekoKonstanteak.xmlFitxategiIzena);

        if(xmlFitxategia.exists()) {
            JAXBContext jaxbContext;

            try {
                jaxbContext = JAXBContext.newInstance(EragiketenWBXML.class);

                Unmarshaller jaxbMarshaller = jaxbContext.createUnmarshaller();
                eWBXML=(EragiketenWBXML)jaxbMarshaller.unmarshal(xmlFitxategia);
            } catch (JAXBException e) {
                e.printStackTrace();
                eWBXML=new EragiketenWBXML();
            }
        }
        else
            eWBXML=new EragiketenWBXML();

        return eWBXML.getEragiketakBXML();
    }

    //HEMENGOKO KODE GUZTIA HURRENGO ORRIALDEAN DAGO
}
```

Testuingurua lortu Testuingurutik marshaller-ak lortu Marshall

2.- XML

2.3.- JAXB

- ADIBIDEA: JAXB

//HEMENGOKO KODE GUZTIA AURREKO ORRIALDEAN DAGO

dl.EragiketenTaulaXML.java

```
private void guztiakIdatzi(List<EragiketaBXML> eragiketakBXML) {
    try {
        EragiketenWBXML eWBXML=new EragiketenWBXML();
        eWBXML.setEragiketakBXML(eragiketakBXML);

        JAXBContext jaxbContext = JAXBContext.newInstance(EragiketenWBXML.class);
        Marshaller jaxbMarshaller = jaxbContext.createMarshaller();
        jaxbMarshaller.setProperty(Marshaller.JAXB_FORMATTED_OUTPUT, true); CSV-tan lerroka idatzi ordez

        jaxbMarshaller.marshal(eWBXML, new File(PaketekoKonstanteak.xmlFitxategiIzena));

    } catch (JAXBException e) {
        e.printStackTrace();
    }
}

public void eragiketaSartuDB(EragiketaBXML eragiketaBXML) {
    List<EragiketaBXML> eragiketakBXML=guztiakIrakurri();

    eragiketakBXML.add(eragiketaBXML);

    guztiakIdatzi(eragiketakBXML);

    return;
}

public List<EragiketaBXML> eragiketaGuztiakLortuDB() {
    return guztiakIrakurri();
}
```

//AURREKO ADIBIDEKO KODE BALIOKIDEA

Konstanteen POJO: dl.PaketekoKonstanteak.java

2.- XML

2.3.- JAXB

- ADIBIDEA: JAXB

```
package bl;

public class EragiketaJB {
    private float eragigai1;
    private String eragilea;
    private float eragigai2;
    private float emaitza;

    public EragiketaJB() {
    }

    public EragiketaJB(float eragigai1, String eragilea, float eragigai2, float emaitza) {
        this.eragigai1 = eragigai1;
        this.eragilea = eragilea;
        this.eragigai2 = eragigai2;
        this.emaitza = emaitza;
    }

    public float getEragigai1() {
        return eragigai1;
    }

    public void setEragigai1(float eragigai1) {
        this.eragigai1 = eragigai1;
    }

    //Beste atributuen getter eta setter metodoak
}
```

bl.EragiketaJB.java

2.- XML

2.3.- JAXB

- ADIBIDEA: JAXB

```
package bl;

public class KalkulagailuaPOJO {
    public void kalkulatu(EragiketaJB eragiketaJB) {
        float er1=eragiketaJB.getEragigai1();
        char er=eragiketaJB.getEragilea().charAt(0);
        float er2=eragiketaJB.getEragigai2();

        float em;

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

            case '-':
                em=er1-er2;
                break;

            case '*':
                em=er1*er2;
                break;

            case '/':
                em=er1/er2;
                break;

            default:
                em=0.0f;
        }

        eragiketaJB.setEmitza(em);
    }
}
```

bl.KalkulagailuaPOJO.java

2.- XML

2.3.- JAXB

- ADIBIDEA: JAXB

```
package bl;

import dl.EragiketenTaulaXML;
import dl.EragiketaBXML;

public class ZerbitzuakPOJO {
    EragiketenTaulaXML etXML;
    KalkulagailuaPOJO kPOJO;

    public ZerbitzuakPOJO() {
        etXML=new EragiketenTaulaXML();
        kPOJO=new KalkulagailuaPOJO();
    }

    public void eragiketaEginBiltegitatu(EragiketaBXML eragiketaBXML) {
        EragiketaJB eragiketaJB=new EragiketaJB(eragiketaBXML.getEragigai1(), eragiketaBXML.getEragilea(),
                                                eragiketaBXML.getEragigai2(), 0.0f);
        kPOJO.kalkulatu(eragiketaJB);

        eragiketaBXML.setEmaitza(eragiketaJB.getEmaitza());
        etXML.eragiketaSartuDB(eragiketaBXML);
    }
}
```

bl.ZerbitzuakPOJO.java

2.- XML

2.3.- JAXB

- ADIBIDEA: JAXB

pl.ErabilitzaileServlet.java

```
package pl;

//import lerroak

@WebServlet("/EDUKIAK/ErabilitzaileServlet")
public class ErabilitzaileServlet extends HttpServlet {
    private static final long serialVersionUID = 1L;

    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
        //AURREKO BERTSIOKO KODE BERDINA
    }

    protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
        String sEr1=request.getParameter("er1");
        String sEr=request.getParameter("er");
        String sEr2=request.getParameter("er2");

        Date dataOrdua=new Date();
        SimpleDateFormat sdf=new SimpleDateFormat("yyyy/MM/dd HH:mm:ss");

        EragiketaBXML eragiketaBXML=new EragiketaBXML(Float.valueOf(sEr1), sEr, Float.valueOf(sEr2), 0.0f, sdf.format(dataOrdua));
        ZerbitzuakPOJO zPOJO=new ZerbitzuakPOJO();
        zPOJO.eragiketaEginBiltegiratu(eragiketaBXML);

        String mezua=eragiketaBXML.laburpena();

        response.setContentType("text/html");

        PrintWriter out = response.getWriter();
        out.println(TxantiloiaPOJO.HASIERA);
        out.println(TxantiloiaPOJO.headTitleStyle("Emaizta","aplikAritEstiloak.css"));
        out.println("<body>");
        out.println("<h1>ESKATUTAKO ERAGIKETAREN EMAITZA</h1>");
        out.println("<p>"+mezua+"</p>");
        out.println(TxantiloiaPOJO.AMAIERA);
        out.close();
    }
}
```


2.- XML

2.3.- JAXB



- ADIBIDEA: JAXB

//AURREKO ADIBIDEKO KODE BERDINA	pl.TxantiloiaPOJO.java
<!-- AURREKO ADIBIDEKO KODE BERDINA -->	EDUKIAK/erabiltzaileForm.html
/*AURREKO ADIBIDEKO KODE BERDINA*/	EDUKIAK/aplikAritEstiloak.css

2.- XML

2.3.- JAXB

- ADIBIDEA: JAXB

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<eragiketenTaula>
  <eragiketa>
    <eragigai1>1.0</eragigai1>
    <eragilea>+</eragilea>
    <eragigai2>2.0</eragigai2>
    <emaitza>3.0</emaitza>
    <dataOrdua>2022/12/20 14:44:33</dataOrdua>
  </eragiketa>
  <eragiketa>
    <eragigai1>2.0</eragigai1>
    <eragilea>+</eragilea>
    <eragigai2>3.0</eragigai2>
    <emaitza>5.0</emaitza>
    <dataOrdua>2022/12/20 14:50:49</dataOrdua>
  </eragiketa>
  <eragiketa>
    <eragigai1>1.0</eragigai1>
    <eragilea>+</eragilea>
    <eragigai2>3.0</eragigai2>
    <emaitza>4.0</emaitza>
    <dataOrdua>2022/12/20 15:36:51</dataOrdua>
  </eragiketa>
</eragiketenTaula>
```

Eragiketak.xml: 3 eragiketa exekutatu ondoren sortutako fitxategia

WEB ORRIALDE DINAMIKOAK

EDUKIA

1.- SARRERA

2.- XML

3.- JSON

3.1.- Sarrera

3.2.- Oinarrizko JSON

3.3.- JAXB

3.4.- JSON eta AJAX

ERREFERENTZIAK

3.- JSON

3.1.- Sarrera

- Zer da JSON?
 - JavaScript Object Notation
 - Egituratu Garraiatu eta/edo biltegiratu
- JSON vs XML
 - Antzekotasunak
 - Testu sinpleko fitxategiak
 - Egitura hierarkikoak
 - Ezberdintasunak
 - Sintaxi sinpleagoa: Informazio berdina byte gutxiagotan
 - Fitxategien atzizkiak: .json
 - MIME mota: application/json

WEB ORRIALDE DINAMIKOAK

EDUKIA

1.- SARRERA

2.- XML

3.- JSON

3.1.- Sarrera

3.2.- Oinarrizko JSON

3.3.- JAXB

3.4.- JSON eta AJAX

ERREFERENTZIAK

3.- JSON

3.2.- Oinarrizko JSON

- JSON dokumentuak: testu sinplea
- Sintaxia
 - Datuak: izena, balioa
 - Datuak banatzeko: ,
 - Objektuak: {}
 - Arrayak: []
- Datuen deklarazioa
 - “*izena*”: *balioa*
 - Balioak Zenbakiak, Strings(""), true/False, array, objektuak

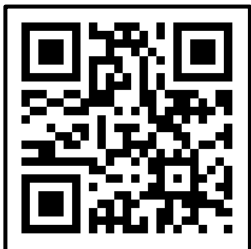
3.- JSON

3.2.- Oinarrizko JSON

- ADIBIDEA: Oinarrizko JSON fitxategiak

```
{
  "sesioa" : {
    "idSesioa" : 1,
    "idErab" : 25,
    "eragiketa" : [
      {"er1" : 12.0, "er" : "-", "er2" : 2.0, "em" : 10.0},
      {"er1" : 34.0, "er" : "+", "er2" : 5.0, "em" : 39.0}
    ]
  }
}
```

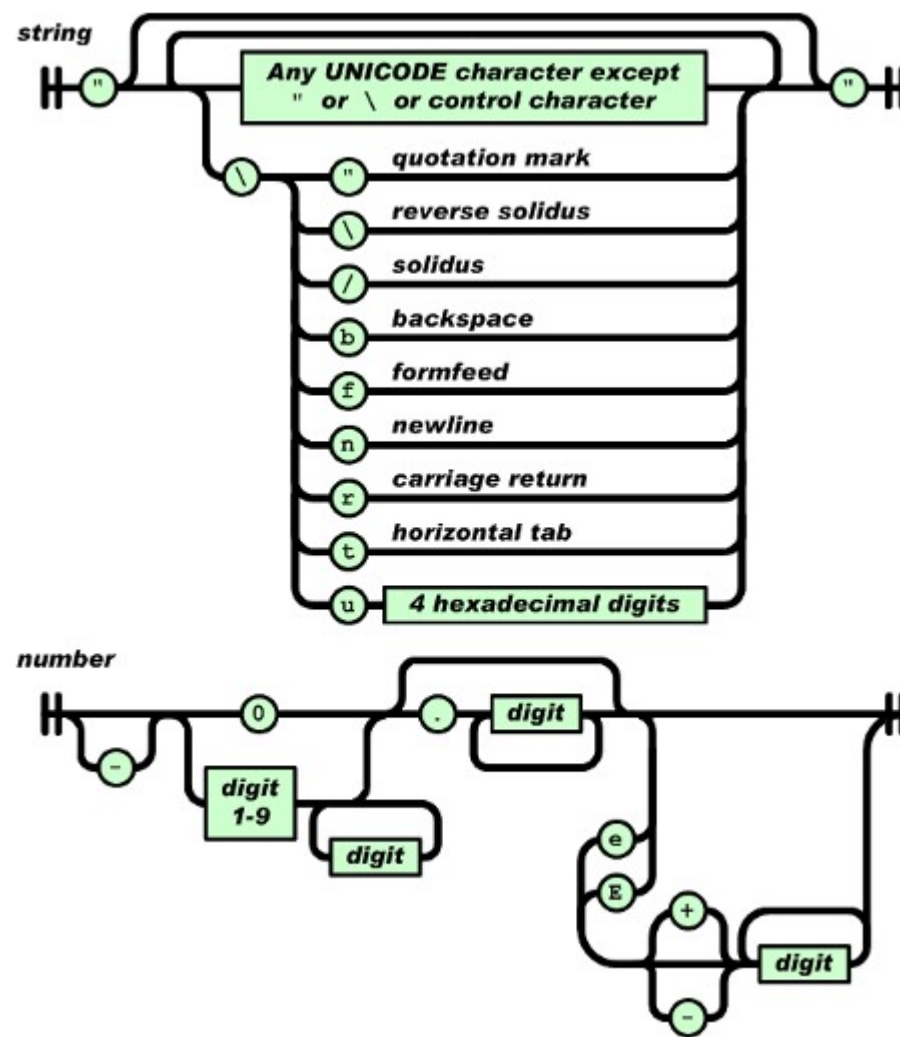
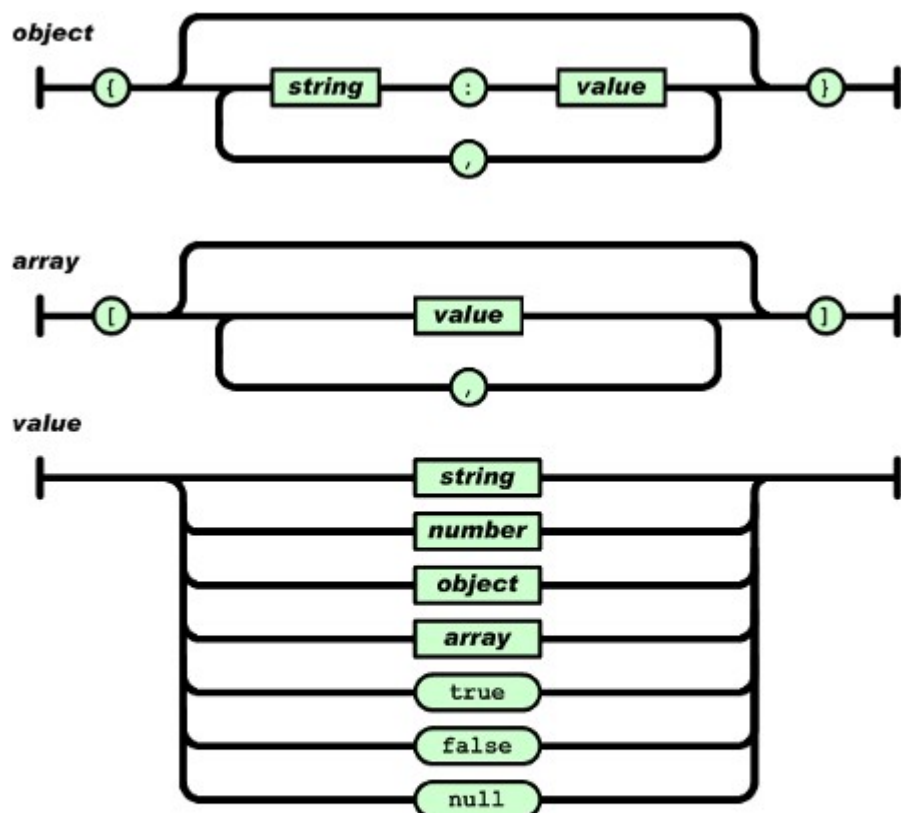
AritmetikakoSesioa2.json



3.- JSON

3.2.- Oinarrizko JSON

- Sintaxiaren laburpena



WEB ORRIALDE DINAMIKOAK

EDUKIA

1.- SARRERA

2.- XML

3.- JSON

3.1.- Sarrera

3.2.- Oinarrizko JSON

3.3.- JAXB

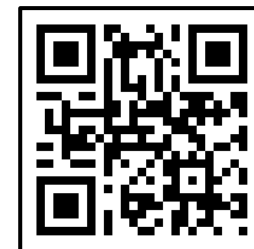
3.4.- JSON eta AJAX

ERREFERENTZIAK

3.- JSON

3.3.- JAXB

- Prozedura
 - JSON formaturako datuak: JavaBeans+oharrak
 - XML kasurako berdinak erabili daitezke
 - Aplikazio Zerbitzari arruntetan: **jaxb.properties**
 - Mapeatzeko kodea idatzi: XML kasuarekiko aldaketak
 - JSON formatua erabili behar dela adierazi behar da, marshal edo unmarshal egin baino lehen
 - Aplikazio Zerbitzari arruntetan: EclipseLink liburutegia



- Adibidea: JAXB

jaxb.properties: JAXB darabilten klaseak dauden paketearen karpetan

`javax.xml.bind.context.factory=org.eclipse.persistence.jaxb.JAXBContextFactory`

`//XML KASUKO KODE BERDINA`

JSON biltegirako JavaBean: `dl.EragiketaBXML.java`

`//XML KASUKO KODE BERDINA`

JSON biltegirako JavaBean: `dl.EragiketenWBXML.java`

`//XML KASUKO KODE BALIOKIDEA`

Konstanteen POJO: `dl.PaketekoKonstanteak.java`

3.- JSON

3.3.- JAXB

- ADIBIDEA: JAXB

dl.EragiketenTaulaJSON.java

```
package dl;

//import lerroak

public class EragiketenTaulaJSON {
    private List<EragiketaBXML> guztiakIrakurri() {
        EragiketenWBXML eWBXML;
        File jsonFitxategia=new File(PaketekoKonstanteak.jsonFitxategiIzena);

        if(jsonFitxategia.exists()) {
            JAXBContext jaxbContext;

            try {
                jaxbContext = JAXBContext.newInstance(EragiketenWBXML.class);

                Unmarshaller jaxbMarshaller = jaxbContext.createUnmarshaller();
                jaxbMarshaller.setProperty(UnmarshallerProperties.MEDIA_TYPE,"application/json");

                eWBXML=(EragiketenWBXML)jaxbMarshaller.unmarshal(jsonFitxategia);
            } catch (JAXBException e) {
                e.printStackTrace();
                eWBXML=new EragiketenWBXML();
            }
        }
        else
            eWBXML=new EragiketenWBXML();

        return eWBXML.getEragiketakBXML();
    }

    //HEMENGOKODEGUZTIAHURRENGORRRIALDEANDAGO

}
```

3.- JSON

3.3.- JAXB

- ADIBIDEA: JAXB

//HEMENGOKODEGUZTIAAURREKOORRIALDEANDAGO

dl.EragiketenTaulaJSON.java

```
private void guztiakIdatzi(List<EragiketaBXML> eragiketakBXML) {
    try {
        EragiketenWBXML eWBXML=new EragiketenWBXML();
        eWBXML.setEragiketakBXML(eragiketakBXML);

        JAXBContext jaxbContext = JAXBContext.newInstance(EragiketenWBXML.class);
        Marshaller jaxbMarshaller = jaxbContext.createMarshaller();
        jaxbMarshaller.setProperty(MarshallerProperties.MEDIA_TYPE,"application/json");
        jaxbMarshaller.setProperty(Marshaller.JAXB_FORMATTED_OUTPUT, true);

        jaxbMarshaller.marshal(eWBXML, new File(PaketekoKonstanteak.jsonFitxategiIzena));
    } catch (JAXBException e) {
        e.printStackTrace();
    }
}

public void eragiketaSartuDB(EragiketaBXML eragiketaBXML) {
    List<EragiketaBXML> eragiketakBXML=guztiakIrakurri();

    eragiketakBXML.add(eragiketaBXML);

    guztiakIdatzi(eragiketakBXML);

    return;
}
```

3.- JSON

3.3.- JAXB

- ADIBIDEA: JAXB

//AURREKO ADIBIDEKO KODE BERDINAK

bl.EragiketaJB.java
bl.KalkulagailuaPOJO.java

```
package bl;

import dl.EragiketenTaulaJSON;
import dl.EragiketaBXML;

public class ZerbitzuakPOJO {
    EragiketenTaulaJSON etJSON;
    KalkulagailuaPOJO kPOJO;

    public ZerbitzuakPOJO() {
        etJSON=new EragiketenTaulaJSON();
        kPOJO=new KalkulagailuaPOJO();
    }

    public void eragiketaEginBiltegitatu(EragiketaBXML eragiketaBXML) {
        EragiketaJB eragiketaJB=new EragiketaJB(eragiketaBXML.getEragigai1(), eragiketaBXML.getEragilea(),
                                                eragiketaBXML.getEragigai2(), 0.0f);
        kPOJO.kalkulatu(eragiketaJB);

        eragiketaBXML.setEmaitza(eragiketaJB.getEmaitza());
        etJSON.eragiketaSartuDB(eragiketaBXML);
    }
}
```

bl.ZerbitzuakPOJO.java

//AURREKO ADIBIDEKO KODE BERDINAK

pl.ErabilitzaileServlet.java
pl.TxantiloiaPOJO.java

<!-- AURREKO ADIBIDEKO KODE BERDINA -->

EDUKIAK/erabilitzaileForm.html

/*AURREKO ADIBIDEKO KODE BERDINA*/

EDUKIAK/aplikAritEstiloak.css

3.- JSON

3.3.- JAXB

- ADIBIDEA: JAXB

```
{
  "eragiketenTaula" : {
    "eragiketa" : [ {
      "eragigai1" : 1.0,
      "eragilea" : "+",
      "eragigai2" : 2.0,
      "emaitza" : 3.0,
      "dataOrdua" : "2018/12/20 15:07:14"
    }, {
      "eragigai1" : 2.0,
      "eragilea" : "-",
      "eragigai2" : 3.0,
      "emaitza" : -1.0,
      "dataOrdua" : "2018/12/20 15:09:01"
    } ]
  }
}
```

Eragiketak.json: 2 eragiketa exekutatu ondoren sortutako fitxategia

WEB ORRIALDE DINAMIKOAK

EDUKIA

1.- SARRERA

2.- XML

3.- JSON

3.1.- Sarrera

3.2.- Oinarrizko JSON

3.3.- JAXB

3.4.- JSON eta AJAX

ERREFERENTZIAK

3.- JSON

3.4.- JSON eta AJAX

- Zer da AJAX?
 - Orrialde baten bistaratzea aldatzea
 - JSko **XMLHttpRequest** motako objektua: zerbitzarirako HTTP-request
 - Erantzuneko JSON kodea prozesatu eta JS objektuan itzuli
 - `eval("(“+JSONKodea+”))”)`
 - `JSON.parse(JSONKodea)`

3.- JSON

3.4.- JSON eta AJAX

- ADIBIDEA: JSON+AJAX

```
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>Koordenatuak</title>
<script>
function koordinatuakLortu() {
    kokap=document.getElementById("kokapena").value;
    kokapZuriunerikGabe=zuriuneakOrdezkatu(kokap);

    if(window.XMLHttpRequest) { // code for IE7+, Firefox, Chrome, Opera, Safari
        xmlhttp=new XMLHttpRequest();
    }
    else { // code for IE6, IE5
        xmlhttp=new ActiveXObject("Microsoft.XMLHTTP");
    }
    xmlhttp.open("GET","https://maps.googleapis.com/maps/api/geocode/json?address="+kokapZuriunerikGabe+"&sensor=false&key=XXX",false);
    xmlhttp.send();
    jsonErantzuna=xmlhttp.responseText;

    var js0=JSON.parse(jsonErantzuna);

    document.getElementById("latitudea").innerHTML=""+js0.results[0].geometry.location.lat;
    document.getElementById("longitudea").innerHTML=""+js0.results[0].geometry.location.lng;
}

function zuriuneakOrdezkatu(esaldia) {
    hitzak=esaldia.split(" ");
    l=hitzak.length;
    zuriunerikGabe=hitzak[0];
    for(i=1;i<l;i++)
        zuriunerikGabe+=" "+hitzak[i];

    return zuriunerikGabe;
}
</script>
</head>
<!-- HEMENGO LERRO GUZTIAK HURRENGO ORRIALDEAN DAUDE -->
```

3.- JSON

3.4.- JSON eta AJAX

- ADIBIDEA: JSON+AJAX

```
<!-- HEMENGO LERRO GUZTIAK AURREKO ORRIALDEAN DAUDE -->
<body>
  <h1>TOKIKO KOORDENATUAK JAKITEKO APLIKAZIOA</h1>
  <b>Latitudea:</b><label id="latitudea"></label><br>
  <b>Longitudea:</b><label id="longitudea"></label><br>
  <input id="kokapena">
  <button onclick="koordinatuakLortu();">Koordenatuak lortu</button>
</body>
</html>
```

koordinatuak.html

DATUEN ADIERAZPENA

ERREFERENTZIAK

[1] XML: <http://www.w3schools.com/xml/default.asp>

[2] JSON: <http://www.w3schools.com/json/>