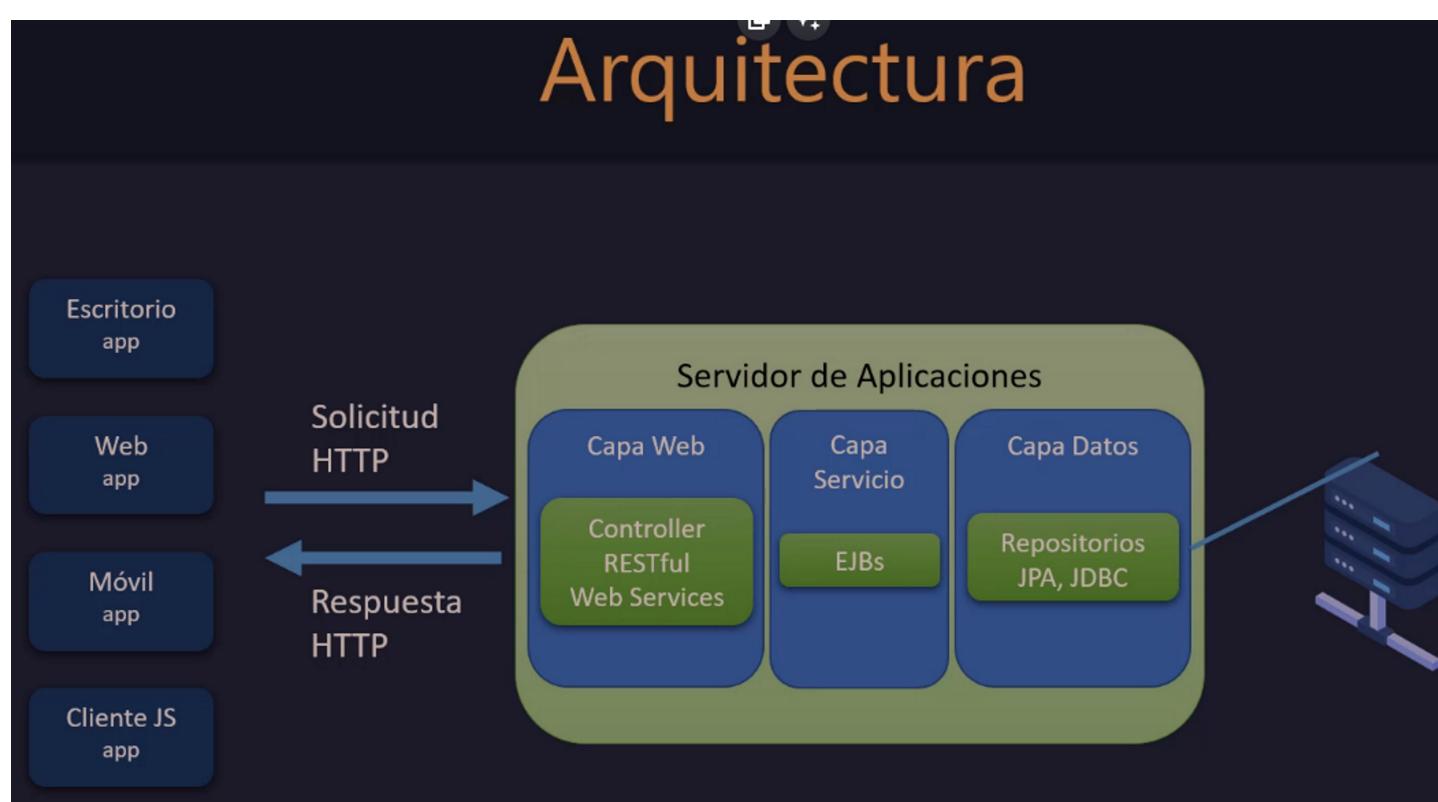


SECCION 73 :-----

----- CLASE 578 -----



Características

Interoperabilidad, permite compartir datos entre sistemas

Estándares basados en hipermedios (XML, JSON)

Protocolo cliente/servidor sin estado

Sintaxis universal para identificar los recursos, cada recurso es direccionable únicamente a través de su URI.

Utiliza diferentes métodos HTTP: GET, POST, PUT, DELETE, OPTIONS y HEAD

Utiliza diferentes códigos de estados HTTP: 200, 201, 204, 302, 400, 401, 403, 404, 500

----- CLASE 579 -----

PROYECTO JAX RS :
SE COPIA EL DE SOAP Y SE MODIFICA.

D4_73_579_webapp-jaxrs-rest - pom.xml (webapp-jaxrs-rest)

EL POM SE MANTIENE:

```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>

  <groupId>webapp.jaxrs.rest</groupId>
  <artifactId>webapp-jaxrs-rest</artifactId>
  <version>1.0-SNAPSHOT</version>

  <packaging>war</packaging>

  <properties>
    <maven.compiler.source>17</maven.compiler.source>
    <maven.compiler.target>17</maven.compiler.target>
    <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
    <project.reporting.outputEncoding>UTF-8</project.reporting.outputEncoding>
  </properties>

  <dependencies>
    <dependency>
      <groupId>jakarta.platform</groupId>
      <artifactId>jakarta.jakartaee-api</artifactId>
      <version>9.1.0</version>
      <scope>provided</scope>
    </dependency>
  </dependencies>

  <build>
    <finalName>${project.artifactId}</finalName>
    <plugins>
      <plugin>
        <artifactId>maven-compiler-plugin</artifactId>
        <version>3.8.1</version>
      </plugin>
      <plugin>
        <artifactId>maven-war-plugin</artifactId>
        <version>3.2.3</version>
        <configuration>
          <failOnMissingWebXml>false</failOnMissingWebXml>
        </configuration>
      </plugin>
      <plugin>
        <groupId>org.wildfly.plugins</groupId>
        <artifactId>wildfly-maven-plugin</artifactId>
        <version>2.1.0.Final</version>
      </plugin>
    </plugins>
  </build>
</project>
```

CONFIGURACIÓN DE WEB XML :

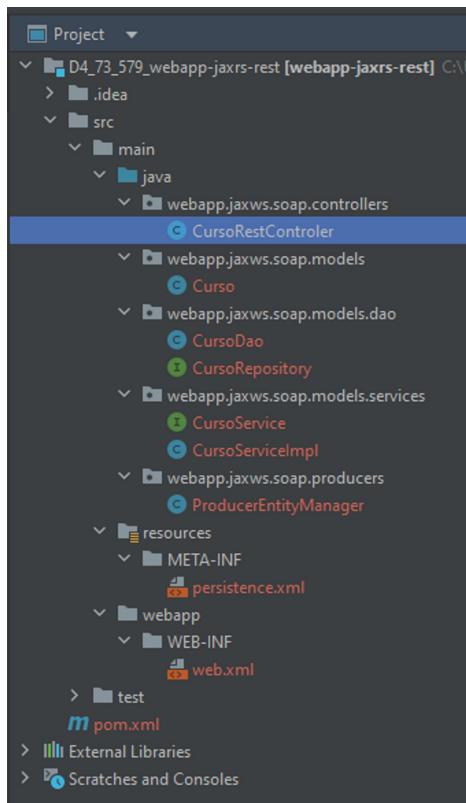
RUTA : src/main/webapp/WEB-INF

```
<?xml version="1.0" encoding="UTF-8" ?>
<!!--579-->
<!!--https://jakarta.ee/specifications/servlet/5.0/jakarta-servlet-spec-5.0.html#deployment-descriptor-2      sección 14.4.1-->

<web-app xmlns="https://jakarta.ee/xml/ns/jakartaee"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="https://jakarta.ee/xml/ns/jakartaee https://jakarta.ee/xml/ns/jakartaee/web-app_5_0.xsd "
  version="5.0">

  <servlet>
    <servlet-name>jakarta.ws.rs.core.Application</servlet-name>
  </servlet>
  <servlet-mapping>
    <servlet-name>jakarta.ws.rs.core.Application</servlet-name>
    <url-pattern>/api/*</url-pattern>
  </servlet-mapping>
</web-app>
```

EL PACKAGE DICE SOAP , PERO DEBERIA DECIR REST ...



The screenshot shows the IntelliJ IDEA code editor with the file 'CursoRestController.java' open. The code defines a REST controller for managing courses. It uses annotations like @RequestScoped, @Path, @Produces, @Inject, and @GET to define endpoints. The controller has methods for listing all courses, getting a course by ID, and creating a new course. It interacts with a 'service' which is injected via @Inject.

```
24     @RequestScoped
25     @Path("/cursos")      //1
26     @Produces(MediaType.APPLICATION_XML)      //2
27     public class CursoRestController {
28
29         @Inject
30         private CursoService service;
31
32         @GET
33         public List<Curso> listar(){
34             return service.listar();
35         }
36
37         /*
38         @GET
39         public Response listar(){
40             return Response.ok(service.listar()).build();
41         }
42         */
43
44         @GET
45         @Path("/{id}")
46         public Response porId(@PathParam("id") Long id){
47             Optional<Curso> optionalCurso = service.porId(id);
48             if(optionalCurso.isPresent()){
49                 //return Response.ok(optionalCurso.get(),MediaType.APPLICATION_XML).build();
50                 return Response.ok(optionalCurso.get()).build();
51             }
52             return Response.status(Response.Status.NOT_FOUND).build();
53         }
54
55         @POST
56         @Consumes(MediaType.APPLICATION_XML)      //3
57         public Response crear(Curso curso){
58             try{
59                 Curso nuevoCurso = service.guardar(curso);
60                 return Response.ok(nuevoCurso).build();
61             } catch (Exception e) {
62                 e.printStackTrace();
63                 return Response.serverError().build();
64             }
65         }
66     }
```

```

67
68     @PUT
69     @Path("/{id}")
70     @Consumes(MediaType.APPLICATION_XML)
71     public Response editar(@PathParam("id") Long id, Curso curso){
72         Optional<Curso> cursoOptional = service.porId(id);
73         if(cursoOptional.isPresent()){
74             Curso nuevoCurso = cursoOptional.get();
75             nuevoCurso.setNombre(curso.getNombre());
76             nuevoCurso.setDescripcion(curso.getDescripcion());
77             nuevoCurso.setDuracion(curso.getDuracion());
78             nuevoCurso.setInstructor(curso.getInstructor());
79             try{
80                 service.guardar(nuevoCurso);
81                 return Response.ok(nuevoCurso).build();
82             } catch (Exception e) {
83                 e.printStackTrace();
84                 return Response.serverError().build();
85             }
86         }
87         return Response.status(Response.Status.NOT_FOUND).build();
88     }
89
90     @DELETE
91     @Path("/{id}")
92     public Response eliminar(@PathParam("id") Long id){
93         Optional<Curso> cursoOptional = service.porId(id);
94         if(cursoOptional.isPresent()){
95             try{
96                 service.eliminar(cursoOptional.get().getId());
97                 return Response.noContent().build();
98             } catch (Exception e) {
99                 e.printStackTrace();
100                return Response.serverError().build();
101            }
102        }
103        return Response.status(Response.Status.NOT_FOUND).build();
104    }
105
106 }
107

```

AHORA

LEVANTA WILDFLY:

```

PS C:\developjava\wildfly-preview-26.1.3.Final\bin> cd C:\developjava\wildfly-preview-26.1.3.Final\bin
PS C:\developjava\wildfly-preview-26.1.3.Final\bin> .\standalone.bat

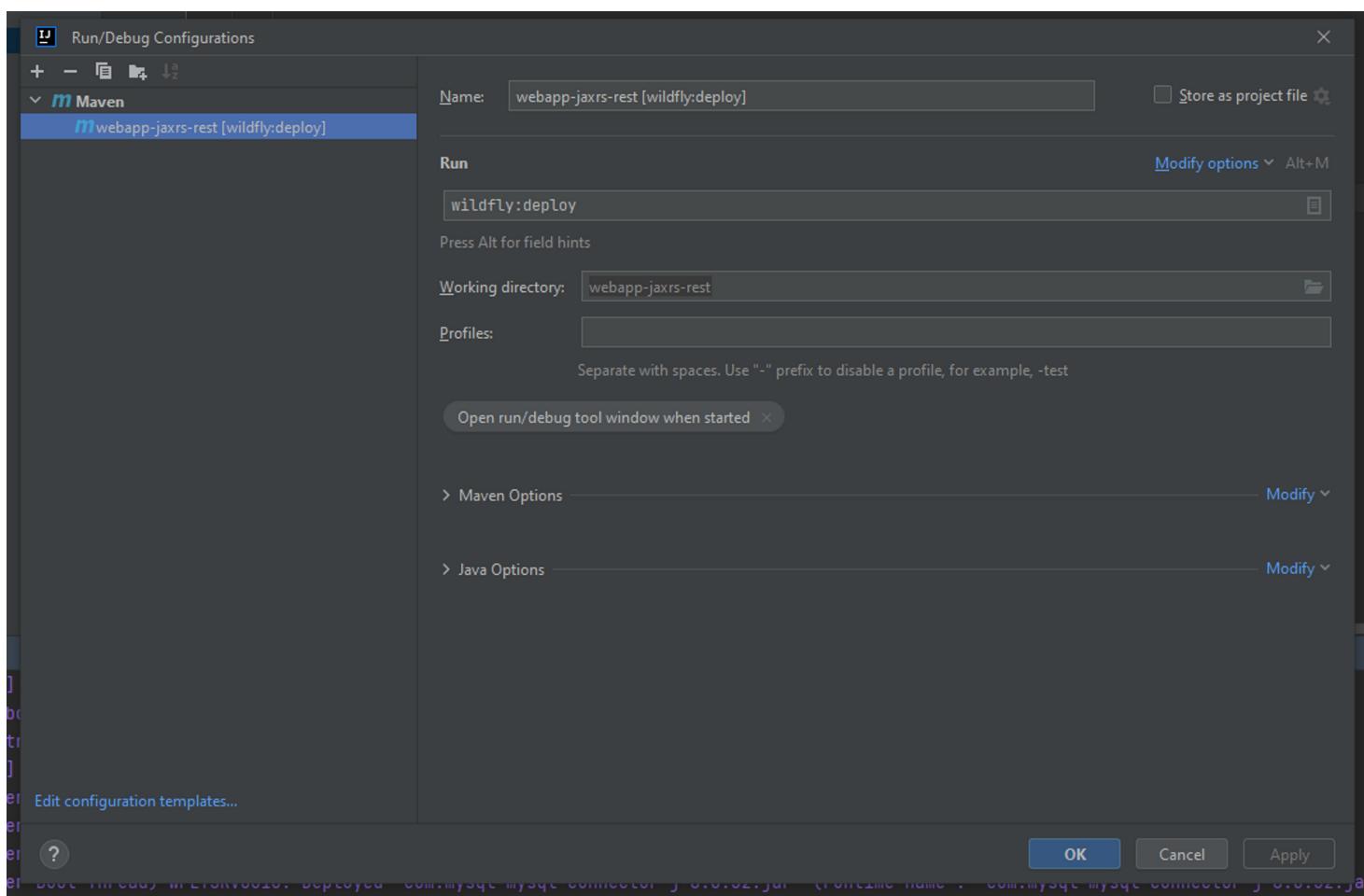
```

```

axel
12345

```

CONFIGURA EL RUN:



The deployment **webapp-jaxrs-rest.war** is enabled and active. [Disable](#)

Main Attributes	
Name:	webapp-jaxrs-rest.war
Runtime Name:	webapp-jaxrs-rest.war
Context Root:	/webapp-jaxrs-rest
Hash:	c5c097b7e5ae03b0384721a73c73d658a1c3ae00
Enabled, Managed, Exploded:	<input checked="" type="checkbox"/> <input checked="" type="checkbox"/> <input type="checkbox"/>
Status:	OK
Last enabled at:	5/14/23, 12:47 PM
Last disabled at:	n/a

AQUÍ ESTÁN TODOS LOS MÉTODOS REST DEL PROYECTO:

w.j.s.controllers.CursoRestController

REST Resource (1)

Filter by: Rest name, method or media type

w.j.s.controllers.CursoRestController
webapp-jaxrs-rest.war

Resource Paths

/cursos

- application/xml
- application/xml
- POST** /webapp-jaxrs-rest/api/cursos
- jakarta.ws.rs.core.Response webapp.jaxws.soap.controllers.CursoRestController.create(webapp.jaxws.soap.models.Curso arg0)

/cursos/{id}

- application/xml
- application/xml
- PUT** /webapp-jaxrs-rest/api/cursos/{id}
- jakarta.ws.rs.core.Response webapp.jaxws.soap.controllers.CursoRestController.editar(@PathParam java.lang.Long id, webapp.jaxws.soap.models.Curso arg1)

- application/xml
- DELETE** /webapp-jaxrs-rest/api/cursos/{id}
- jakarta.ws.rs.core.Response webapp.jaxws.soap.controllers.CursoRestController.eliminar(@PathParam java.lang.Long id)

- application/xml
- GET** /webapp-jaxrs-rest/api/cursos/{id}
- jakarta.ws.rs.core.Response webapp.jaxws.soap.controllers.CursoRestController.porId(@PathParam java.lang.Long id)

RECIBIENDO RESPUESTA XML AL HACER CLICK EN

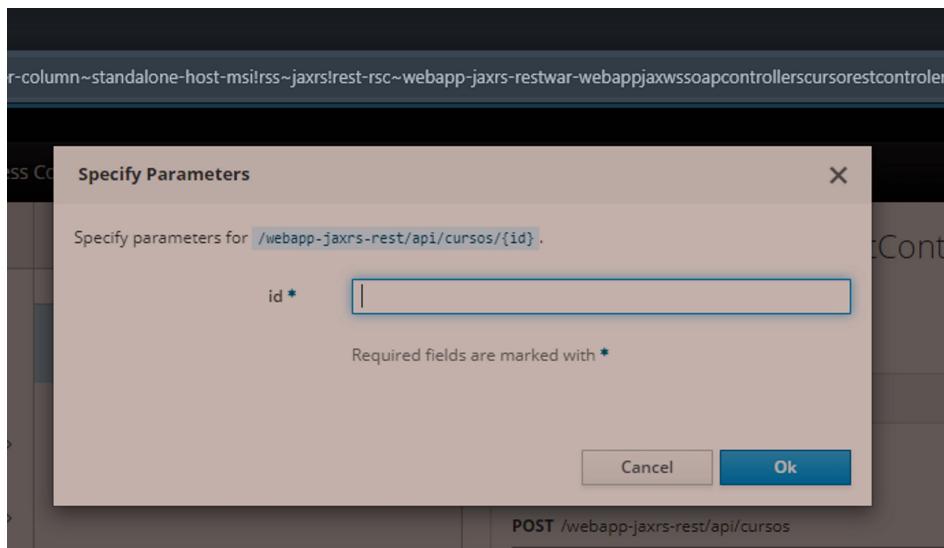
GET/webapp-jaxrs-rest/api/cursos

This XML file does not appear to have any style information associated with it. The document tree is shown below.

```
<collection>
  <link type="text/css" rel="stylesheet" id="dark-mode-custom-link"/>
  <link type="text/css" rel="stylesheet" id="dark-mode-general-link"/>
  <style lang="en" type="text/css" id="dark-mode-custom-style"/>
  <style lang="en" type="text/css" id="dark-mode-native-style"/>
  <curso>
    <descripcion>java</descripcion>
    <duracion>100.0</duracion>
    <id>1</id>
    <instructor>andres</instructor>
    <nombre>java</nombre>
  </curso>
  <curso>
    <descripcion>jakarta</descripcion>
    <duracion>50.0</duracion>
    <id>2</id>
    <instructor>andres</instructor>
    <nombre>javaee</nombre>
  </curso>
  <curso>
    <descripcion>angular</descripcion>
    <duracion>40.0</duracion>
    <id>3</id>
    <instructor>andres</instructor>
    <nombre>angular</nombre>
  </curso>
  <curso>
    <descripcion>spring boot</descripcion>
    <duracion>40.0</duracion>
    <id>4</id>
    <instructor>andres</instructor>
    <nombre>spring</nombre>
  </curso>
  <curso>
    <descripcion>react js</descripcion>
    <duracion>50.0</duracion>
    <id>5</id>
    <instructor>Andrés Guzmán</instructor>
    <nombre>React</nombre>
  </curso>
</collection>
```

AHORA USANDO EL ID POR ID :

GET/webapp-jaxrs-rest/api/cursos/{id}



This XML file does not appear to have any style information associated with it. The document tree is shown below.

```
<curso>
  <link type="text/css" rel="stylesheet" id="dark-mode-custom-link"/>
  <link type="text/css" rel="stylesheet" id="dark-mode-general-link"/>
  <style lang="en" type="text/css" id="dark-mode-custom-style"/>
  <style lang="en" type="text/css" id="dark-mode-native-style"/>
  <descripcion>java</descripcion>
  <duracion>100.0</duracion>
  <id>1</id>
  <instructor>andres</instructor>
  <nombre>java</nombre>
</curso>
```

AHORA CON EL POSTMAN

localhost:8080/webapp-jaxrs-rest/api/cursos

GET localhost:8080/webapp-jaxrs-rest/api/cursos

Params Authorization Headers (6) Body Pre-request Script Tests Settings

Query Params

Key	Value
Key	Value

Body Cookies Headers (4) Test Results

Pretty Raw Preview Visualize XML

```

3   <curso>
4     <descripcion>java</descripcion>
5     <duracion>100.0</duracion>
6     <id>1</id>
7     <instructor>andres</instructor>
8     <nombre>java</nombre>
9   </curso>
10  <curso>
11    <descripcion>jakarta</descripcion>
12    <duracion>50.0</duracion>
13    <id>2</id>
14    <instructor>andres</instructor>
15    <nombre>javaee</nombre>
16  </curso>
17  <curso>
18    <descripcion>angular</descripcion>
19    <duracion>40.0</duracion>
20    <id>3</id>
21    <instructor>andres</instructor>
22    <nombre>angular</nombre>
23  </curso>
24  <curso>
25    <descripcion>spring boot</descripcion>
26    <duracion>40.0</duracion>
27    <id>4</id>
28    <instructor>andres</instructor>
29    <nombre>spring</nombre>
30  </curso>
31  <curso>
32    <descripcion>react js</descripcion>
33    <duracion>50.0</duracion>
34    <id>5</id>
35    <instructor>Andrés Guzmán</instructor>
36    <nombre>React</nombre>
37  </curso>

```

localhost:8080/webapp-jaxrs-rest/api/cursos/4

GET localhost:8080/webapp-jaxrs-rest/api/cursos/4

Params Authorization Headers (6) Body Pre-request Script Tests Settings

Query Params

Key	Value
Key	Value

Body Cookies Headers (4) Test Results

Pretty Raw Preview Visualize XML

```

1 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
2 <curso>
3   <descripcion>spring boot</descripcion>
4   <duracion>40.0</duracion>
5   <id>4</id>
6   <instructor>andres</instructor>
7   <nombre>spring</nombre>
8 </curso>

```

localhost:8080/webapp-jaxrs-rest/api/cursos

POST localhost:8080/webapp-jaxrs-rest/api/cursos

Params Authorization Headers (8) Body Pre-request Script Tests Settings

none form-data x-www-form-urlencoded raw binary GraphQL XML

```

1 <curso>
2   <descripcion>node js</descripcion>
3   <duracion>40.0</duracion>
4   <instructor>andres</instructor>
5   <nombre>node js y express</nombre>
6 </curso>

```

Body Cookies Headers (4) Test Results

Pretty Raw Preview Visualize XML

⊕ Status: 200 OK Time: 43 ms Size: 350 B

Body Cookies Headers (4) Test Results

Status: 200 OK Time: 43 ms Size: 350 B

Pretty Raw Preview Visualize XML

```

1 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
2 <curso>
3   <descripcion>node js</descripcion>
4   <duracion>40.0</duracion>
5   <id>6</id>
6   <instructor>andres</instructor>
7   <nombre>node js y express</nombre>
8 </curso>

```

localhost:8080/webapp-jaxrs-rest/api/cursos/6

PUT localhost:8080/webapp-jaxrs-rest/api/cursos/6

Params Authorization Headers (8) Body Pre-request Script Tests Settings

Body

```

1 <curso>
2   <descripcion>node js</descripcion>
3   <duracion>400.0</duracion>
4   <instructor>andres</instructor>
5   <nombre>node js y express</nombre>
6 </curso>

```

Body Cookies Headers (4) Test Results

Status: 200 OK Time: 26 ms Size: 351 B

Pretty Raw Preview Visualize XML

```

1 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
2 <curso>
3   <descripcion>node js</descripcion>
4   <duracion>400.0</duracion>
5   <id>6</id>
6   <instructor>andres</instructor>
7   <nombre>node js y express</nombre>
8 </curso>

```

CREO UN CONTROLADOR REST PARA JSON AHORA

WildFly

Homepage Deployments Configuration Runtime Patching Access Control

Server	Monitor	REST Resource (2)
msi	Status	w.j.s.controllers.CursoRestController webapp-jaxrs-rest.war
	Management Operations	w.j.s.controllers.CursoRestControllerJson webapp-jaxrs-rest.war
	Batch jBeret	
	Datasources	
	EJB	
	IO	
	JAX-RS	
	JNDI	
	JPA	

w.j.s.controllers.CursoRest
webapp-jaxrs-rest.war

Resource Paths

- /cursos-json
 - application/json
 - ← application/json
 - POST /webapp-jaxrs-rest/api/cursos-json**
jakarta.ws.rs.core.Response webapp.jaxws.so
- ← application/json
- GET /webapp-jaxrs-rest/api/cursos-json**
java.util.List webapp.jaxws.soap.controllers
- /cursos-json/{id}

Java master jakarta EE9 / localhost:8080/webapp-jaxrs-rest/api/cursos-json

GET localhost:8080/webapp-jaxrs-rest/api/cursos-json

Params Authorization Headers (6) Body Pre-request Script Tests Settings

Query Params

Key	Value
-----	-------

Body Cookies Headers (4) Test Results

Pretty Raw Preview Visualize JSON

```

1 [
2   {
3     "descripcion": "java",
4     "duracion": 100.0,
5     "id": 1,
6     "instructor": "andres",
7     "nombre": "java"
8   },
9   {
10    "descripcion": "jakarta",
11    "duracion": 50.0,
12    "id": 2,
13    "instructor": "andres",
14    "nombre": "javaee"
15  },
16  {
17    "descripcion": "angular",
18    "duracion": 40.0,
19    "id": 3,
20    "instructor": "andres",
21  }
]

```

```

17     "descripcion": "angular",
18     "duracion": 40.0,
19     "id": 3,
20     "instructor": "andres",
21     "nombre": "angular"
22   },
23 }

```

localhost:8080/webapp-jaxrs-rest/api/cursos-json/2

GET localhost:8080/webapp-jaxrs-rest/api/cursos-json/2

Params Authorization Headers (6) Body Pre-request Script Tests Settings

Query Params

Key	Value

Body Cookies Headers (4) Test Results

Pretty Raw Preview Visualize JSON ↗

```

1
2   "descripcion": "jakarta",
3   "duracion": 50.0,
4   "id": 2,
5   "instructor": "andres",
6   "nombre": "javaee"
7

```

localhost:8080/webapp-jaxrs-rest/api/cursos-json

POST localhost:8080/webapp-jaxrs-rest/api/cursos-json

Params Authorization Headers (8) Body Pre-request Script Tests Settings

none form-data x-www-form-urlencoded raw binary GraphQL JSON ↗

```

1
2   "descripcion": "jakarta EE9",
3   "duracion": 150.0,
4   "instructor": "andres",
5   "nombre": "javaee"
6
7

```

Body Cookies Headers (4) Test Results

Pretty Raw Preview Visualize JSON ↗

```

1
2   "descripcion": "jakarta EE9",
3   "duracion": 150.0,
4   "id": 7,
5   "instructor": "andres",
6   "nombre": "javaee"
7

```

PUT localhost:8080/webapp-jaxrs-rest/api/cursos-json/7

Params Authorization Headers (8) Body Pre-request Script Tests Settings

none form-data x-www-form-urlencoded raw binary GraphQL JSON ↗

```

1
2   "descripcion": "Python",
3   "duracion": 70.0,
4   "instructor": "xxxxxxxx",
5   "nombre": "zzzzzz"
6
7

```

Body Cookies Headers (4) Test Results

Pretty Raw Preview Visualize JSON ↗

```

1
2   "descripcion": "Python",
3   "duracion": 70.0,
4   "id": 7,
5   "instructor": "xxxxxxxx",
6   "nombre": "zzzzzz"
7

```

localhost:8080/webapp-jaxrs-rest/api/cursos-json/6

DELETE localhost:8080/webapp-jaxrs-rest/api/cursos-json/6

Params Authorization Headers (6) Body Pre-request Script Tests Settings

Query Params

Key	Value	Description
Key	Value	Description

Body Cookies Headers (1) Test Results

Pretty Raw Preview Visualize Text ↗

Status: 204 No Content Time: 21 ms Size: 64

localhost:8080/webapp-jaxrs-rest/api/cursos-json/6

DELETE localhost:8080/webapp-jaxrs-rest/api/cursos-json/6

Params Authorization Headers (6) Body Pre-request Script Tests Settings

Query Params

Key	Value	Description
Key	Value	Description

Body Cookies Headers (1) Test Results

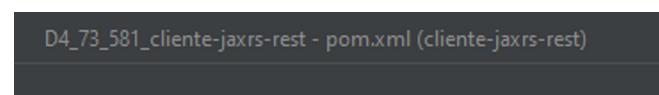
Status: 204 No Content Time: 21 ms Size: 64

Pretty Raw Preview Visualize Text

1

----- CLASE 581 -----

Ejemplo Cliente HTTP con RESTEasy JAX-RS parte 1



CONSUME EL API REST CON UNA APP CLIENTE

POM

```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">
<modelVersion>4.0.0</modelVersion>

<groupId>cliente.jaxrs.rest</groupId>
<artifactId>cliente-jaxrs-rest</artifactId>
<version>1.0-SNAPSHOT</version>

<properties>
    <maven.compiler.source>17</maven.compiler.source>
    <maven.compiler.target>17</maven.compiler.target>
    <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
</properties>

<dependencies>
    <!-- resteasy-client INCLUYE A resteasy-core, PERO MEJOR PONER resteasy-core DE MANERA EXPLICITA.-->
    <dependency>
        <groupId>org.jboss.resteasy</groupId>
        <artifactId>resteasy-core</artifactId>
        <version>6.2.3.Final</version>
    </dependency>
    <dependency>
        <groupId>org.jboss.resteasy</groupId>
        <artifactId>resteasy-client</artifactId>
        <version>6.2.3.Final</version>
    </dependency>
    <!--PARA SOPORTE XML-->
    <dependency>
        <groupId>org.jboss.resteasy</groupId>
        <artifactId>resteasy-jaxb-provider</artifactId>
        <version>6.2.3.Final</version>
    </dependency>
    <!--PARA SOPORTE JSON-->
    <dependency>
        <groupId>org.jboss.resteasy</groupId>
        <artifactId>resteasy-jackson2-provider</artifactId>
        <version>6.2.3.Final</version>
    </dependency>
</dependencies>
</project>
```

LEVANTA WILDFLY:

```
PS C:\developjava\wildfly-preview-26.1.3.Final\bin> cd C:\developjava\wildfly-preview-26.1.3.Final\bin
PS C:\developjava\wildfly-preview-26.1.3.Final\bin> .\standalone.bat
```

```
axel
12345
```

DESPLIEGA LA APP webapp-jaxrs-rest SI ES Q NO ESTÁ DESPLEGADA EN WILDFLY

- 1 CÓDIGO DE CONEXIÓN
- 2 EJECUTA CON EL RUN MAIN
- 3 CONSUMIENDO EL API REST

D4_73_581_cliente-jaxrs-rest - src - main - java - cliente - jaxrs - rest - Main

Main.java

```
package cliente.jaxrs.rest;
//581 : SE CONSUME EL API REST CON JAX RS Y LA IMPLEMENTACION REST EASY. WILDFLY, POR DEBAJO USA REST EASY PARA EJECUTAR LAS APLICACIONES REST.
//1 - RUTA BASE PARA TODOS LOS MÉTODOS DEL API REST.

import cliente.jaxrs.rest.models.Curso;
import jakarta.ws.rs.client.Client;
import jakarta.ws.rs.client.ClientBuilder;
import jakarta.ws.rs.client.WebTarget;
import jakarta.ws.rs.core.MediaType;

public class Main {
```

```

1 package cliente.jaxrs.rest;
2
3 //581 : SE CONSUME EL API REST CON JAX RS Y LA IMPLEMENTACION REST EASY. WILDFLY, POR DEBAJO USA REST EASY PARA EJECUTAR LAS APLICACIONES REST.
4 //1 - RUTA BASE PARA TODOS LOS MÉTODOS DEL API REST.
5
6 import cliente.jaxrs.rest.models.Curso;
7 import jakarta.ws.rs.client.Client;
8 import jakarta.ws.rs.client.ClientBuilder;
9 import jakarta.ws.rs.client.WebTarget;
10 import jakarta.ws.rs.core.MediaType;
11
12 public class Main {
13     public static void main(String[] args) {
14
15         Client client = ClientBuilder.newClient();
16         WebTarget rootUri = client.target("http://localhost:8080/webapp-jaxrs-rest/api").path("/cursos-json"); //1
17
18         System.out.println("===== POR ID =====");
19         Curso curso = rootUri.path("/2").request(MediaType.APPLICATION_JSON).get(Curso.class);
20         System.out.println(curso);
21
22     }
23 }

```

Run: Main x

C:\developjava\jdk-17.0.2\bin\java.exe ...
===== POR ID =====
Curso{id=2, nombre='javabee', descripcion='jakarta EE9', instructor='andres', duracion=100.0} ✓ 3
Process finished with exit code 0

AHORA USANDO RESPONSE

```

14 public class Main {
15     public static void main(String[] args) {
16
17         Client client = ClientBuilder.newClient();
18         WebTarget rootUri = client.target("http://localhost:8080/webapp-jaxrs-rest/api").path("/cursos-json"); //1
19
20         /*
21         System.out.println("===== POR ID =====");
22         Curso curso = rootUri.path("/2").request(MediaType.APPLICATION_JSON).get(Curso.class);
23         System.out.println(curso);
24         */
25
26         System.out.println("===== POR ID =====");
27         Response response = rootUri.path("/2").request(MediaType.APPLICATION_JSON).get();
28         Curso curso = response.readEntity(Curso.class);
29         System.out.println(curso);
30         System.out.println(response.getStatus() + " / " + response.getMediaType() + " / " + response.getLocation() + " / " + response.getDate() + " / " + response.getLength() + " / " + response.getHeaders());
31     }

```

Run: Main x

C:\developjava\jdk-17.0.2\bin\java.exe ...
===== POR ID =====
Curso{id=2, nombre='javabee', descripcion='jakarta EE9', instructor='andres', duracion=100.0}
200 / application/json / null / Sun May 14 20:50:06 CDT 2023 / 93 / [Connection=keep-alive,Content-Length=93,Content-Type=application/json,Date=Mon, 15 May 2023 01:50:06 GMT]
Process finished with exit code 0

CONSUMIENDO MÉTODO LISTAR:

```

35
36     System.out.println("===== LISTAR =====");
37     List<Curso> cursos = rootUri.request(MediaType.APPLICATION_JSON).get(Response.class).readEntity(new GenericType<List<Curso>>(){}); //2
38     cursos.forEach(System.out::println);
39
40 }

```

Run: Main x

C:\developjava\jdk-17.0.2\bin\java.exe ...
===== POR ID =====
Curso{id=2, nombre='javabee', descripcion='jakarta EE9', instructor='andres', duracion=100.0}
===== LISTAR =====
Curso{id=1, nombre='java', descripcion='java', instructor='andres', duracion=100.0}
Curso{id=2, nombre='javabee', descripcion='jakarta EE9', instructor='andres', duracion=100.0}
Curso{id=3, nombre='angular', descripcion='angular', instructor='andres', duracion=40.0}
Curso{id=4, nombre='spring', descripcion='spring boot', instructor='andres', duracion=40.0}
Curso{id=5, nombre='React', descripcion='react js', instructor='Andrés Guzmán', duracion=50.0}
Curso{id=7, nombre='zzzzzz', descripcion='Python', instructor='xxxxxx', duracion=70.0}

Process finished with exit code 0

----- CLASE 582 -----

Ejemplo Cliente HTTP con RESTEasy JAX-RS parte 2

POST - CREANDO NUEVO CURSO

```

//582
//3 - entityHeader : ES UNA ENTIDAD. UN OBJ QUE SE VA A GUARDAR EN EL REQUEST. (ESTO NO TIENE NADA Q VER CON EL ENTITY Q SE USA EN JPA). ..
//.. ESTE ENTITY RECIBE UN GENERICO, ACEPTE EL TIPO DE CONTENIDO, JSON O XML, QUE SE VA A MANDAR EN EL REQUEST, EN TE CASO "CURSO". ..
//.. SE LE PONE ENTITYHEADER PQ VA GUARDADO EN EL CUERPO DEL REQUEST, PERO VA ASOCIADO A LA CABECERA Q CONTIENE EL TIPO DE CONTENIDO.

```

```

46     System.out.println("===== CREANDO =====");
47     Curso cursoNuevo = new Curso();
48     cursoNuevo.setNombre("Spring cloud");
49     cursoNuevo.setDescripcion("Spring cloud - microservicios");
50     cursoNuevo.setDuracion(250);
51     cursoNuevo.setInstructor("Andrés Guzmán");
52     Entity<Curso> entityHeader = Entity.entity(cursoNuevo, MediaType.APPLICATION_JSON); //3
53     curso = rootUri.request(MediaType.APPLICATION_JSON).post(entityHeader, Curso.class); //CREA EL CURSO
54     System.out.println(curso + "\n");
55
56     cursos = rootUri.request(MediaType.APPLICATION_JSON).get(Response.class).readEntity(new GenericType<List<Curso>>(){}); //LISTA ACTUALIZADA
57     cursos.forEach(System.out::println);
58 }
59 }

Run: Main ×
===== CREANDO =====
Curso{id=8, nombre='Spring cloud', descripcion='Spring cloud - microservicios', instructor='Andrés Guzmán', duracion=25.0} ✓
Curso{id=1, nombre='java', descripcion='java', instructor='andres', duracion=100.0}
Curso{id=2, nombre='javaee', descripcion='jakarta EE9', instructor='andres', duracion=100.0}
Curso{id=3, nombre='angular', descripcion='angular', instructor='andres', duracion=40.0}
Curso{id=4, nombre='spring', descripcion='spring boot', instructor='andres', duracion=40.0}
Curso{id=5, nombre='React', descripcion='react js', instructor='Andrés Guzmán', duracion=50.0}
Curso{id=7, nombre='zzzzzz', descripcion='Python', instructor='xxxxxx', duracion=70.0}
Curso{id=8, nombre='Spring cloud', descripcion='Spring cloud - microservicios', instructor='Andrés Guzmán', duracion=25.0} ✓

46     System.out.println("===== CREANDO =====");
47     Curso cursoNuevo = new Curso();
48     cursoNuevo.setNombre("Spring cloud");
49     cursoNuevo.setDescripcion("Spring cloud - microservicios");
50     cursoNuevo.setDuracion(250);
51     cursoNuevo.setInstructor("Andrés Guzmán");
52     Entity<Curso> entityHeader = Entity.entity(cursoNuevo, MediaType.APPLICATION_JSON); //3
53     curso = rootUri.request(MediaType.APPLICATION_JSON).post(entityHeader, Curso.class); //CREA EL CURSO
54     System.out.println(curso + "\n");
55
56     cursos = rootUri.request(MediaType.APPLICATION_JSON).get(Response.class).readEntity(new GenericType<List<Curso>>(){}); //LISTA ACTUALIZADA
57     cursos.forEach(System.out::println);
58 }
59 }

Run: Main ×
===== CREANDO =====
Curso{id=8, nombre='Spring cloud', descripcion='Spring cloud - microservicios', instructor='Andrés Guzmán', duracion=25.0} ✓
Curso{id=1, nombre='java', descripcion='java', instructor='andres', duracion=100.0}
Curso{id=2, nombre='javaee', descripcion='jakarta EE9', instructor='andres', duracion=100.0}
Curso{id=3, nombre='angular', descripcion='angular', instructor='andres', duracion=40.0}
Curso{id=4, nombre='spring', descripcion='spring boot', instructor='andres', duracion=40.0}
Curso{id=5, nombre='React', descripcion='react js', instructor='Andrés Guzmán', duracion=50.0}
Curso{id=7, nombre='zzzzzz', descripcion='Python', instructor='xxxxxx', duracion=70.0}
Curso{id=8, nombre='Spring cloud', descripcion='Spring cloud - microservicios', instructor='Andrés Guzmán', duracion=25.0} ✓

```

PUT - EDITANDO CURSO

```

System.out.println("===== EDITANDO =====");
Curso cursoEditado = curso;
cursoEditado.setNombre("Microservicios con SpringCloud Eureka");
entityHeader = Entity.entity(cursoEditado, MediaType.APPLICATION_JSON);
curso = rootUri.path("/") + curso.getId().request(MediaType.APPLICATION_JSON).put(entityHeader, Curso.class);
System.out.println(curso);

cursos = rootUri.request(MediaType.APPLICATION_JSON).get(Response.class).readEntity(new GenericType<List<Curso>>(){}); //LISTA ACTUALIZADA
cursos.forEach(System.out::println);

```

```

=====
===== CREANDO =====
Curso{id=9, nombre='Spring cloud', descripcion='Spring cloud - microservicios', instructor='Andrés Guzmán', duracion=25.0} ← (⊕)
Curso{id=1, nombre='java', descripcion='java', instructor='andres', duracion=100.0}
Curso{id=2, nombre='javaee', descripcion='jakarta EE9', instructor='andres', duracion=100.0}
Curso{id=3, nombre='angular', descripcion='angular', instructor='andres', duracion=40.0}
Curso{id=4, nombre='spring', descripcion='spring boot', instructor='andres', duracion=40.0}
Curso{id=5, nombre='React', descripcion='react js', instructor='Andrés Guzmán', duracion=50.0}
Curso{id=7, nombre='zzzzzz', descripcion='Python', instructor='xxxxxx', duracion=70.0}
Curso{id=8, nombre='Spring cloud', descripcion='Spring cloud - microservicios', instructor='Andrés Guzmán', duracion=25.0}
Curso{id=9, nombre='Spring cloud', descripcion='Spring cloud - microservicios', instructor='Andrés Guzmán', duracion=25.0} ✓ ✓
=====
===== EDITANDO =====
Curso{id=9, nombre='Microservicios con SpringCloud Eureka', descripcion='Spring cloud - microservicios', instructor='Andrés Guzmán', duracion=25.0}
Curso{id=1, nombre='java', descripcion='java', instructor='andres', duracion=100.0}
Curso{id=2, nombre='javaee', descripcion='jakarta EE9', instructor='andres', duracion=100.0}
Curso{id=3, nombre='angular', descripcion='angular', instructor='andres', duracion=40.0}
Curso{id=4, nombre='spring', descripcion='spring boot', instructor='andres', duracion=40.0}
Curso{id=5, nombre='React', descripcion='react js', instructor='Andrés Guzmán', duracion=50.0}
Curso{id=7, nombre='zzzzzz', descripcion='Python', instructor='xxxxxx', duracion=70.0}
Curso{id=8, nombre='Spring cloud', descripcion='Spring cloud - microservicios', instructor='Andrés Guzmán', duracion=25.0}
Curso{id=9, nombre='Microservicios con SpringCloud Eureka', descripcion='Spring cloud - microservicios', instructor='Andrés Guzmán', duracion=25.0} ✓ ✓

```

Process finished with exit code 0

DELETE - ELIMINANDO CURSO

```

69     System.out.println("===== ELIMINANDO =====");
70     rootUri.path="/" + curso.getId()).request().delete();
71
72     cursos = rootUri.request(MediaType.APPLICATION_JSON).get(Response.class).readEntity(new GenericType<List<Curso>>(){}); //LISTA ACTUALIZADA
73     cursos.forEach(System.out::println);
74 }
75

```

Run: Main x

```

CURSO[id=0, nombre='Spring cloud', descripcion='Spring cloud - microservicios', instructor='Andrés Guzmán', duracion=25.0]
=====
EDITANDO =====
Curso{id=10, nombre='Microservicios con SpringCloud Eureka', descripcion='Spring cloud - microservicios', instructor='Andrés Guzmán', duracion=25.0}
Curso{id=1, nombre='java', descripcion='java', instructor='andres', duracion=100.0}
Curso{id=2, nombre='javaee', descripcion='jakarta EE9', instructor='andres', duracion=100.0}
Curso{id=3, nombre='angular', descripcion='angular', instructor='andres', duracion=40.0}
Curso{id=4, nombre='spring', descripcion='spring boot', instructor='andres', duracion=40.0}
Curso{id=5, nombre='React', descripcion='react js', instructor='Andrés Guzmán', duracion=50.0}
Curso{id=7, nombre='zzzzzz', descripcion='Python', instructor='xxxxxx', duracion=70.0}
Curso{id=8, nombre='Spring cloud', descripcion='Spring cloud - microservicios', instructor='Andrés Guzmán', duracion=25.0}
Curso{id=9, nombre='Microservicios con SpringCloud Eureka', descripcion='Spring cloud - microservicios', instructor='Andrés Guzmán', duracion=25.0}
Curso{id=10, nombre='Microservicios con SpringCloud Eureka', descripcion='Spring cloud - microservicios', instructor='Andrés Guzmán', duracion=25.0}
=====
ELIMINANDO =====
Curso{id=1, nombre='java', descripcion='java', instructor='andres', duracion=100.0}
Curso{id=2, nombre='javaee', descripcion='jakarta EE9', instructor='andres', duracion=100.0}
Curso{id=3, nombre='angular', descripcion='angular', instructor='andres', duracion=40.0}
Curso{id=4, nombre='spring', descripcion='spring boot', instructor='andres', duracion=40.0}
Curso{id=5, nombre='React', descripcion='react js', instructor='Andrés Guzmán', duracion=50.0}
Curso{id=7, nombre='zzzzzz', descripcion='Python', instructor='xxxxxx', duracion=70.0}
Curso{id=8, nombre='Spring cloud', descripcion='Spring cloud - microservicios', instructor='Andrés Guzmán', duracion=25.0}
Curso{id=9, nombre='Microservicios con SpringCloud Eureka', descripcion='Spring cloud - microservicios', instructor='Andrés Guzmán', duracion=25.0}

Process finished with exit code 0

```

----- CLASE 583 -----

SE AGREGA RELACION ENTRE TABLAS

SE AGREGA TABLA INSTRUCTORES

The screenshot shows the MySQL Workbench interface. On the left, the database schema is visible with tables like 'direccion', 'direcciones', 'facturas', 'instructores', 'productos', 'tbl_alumnos_cursos', 'tbl_clientes_direcciones', and 'usuarios'. The 'instructores' table is selected, and its data grid shows one row: id=1, nombre='Andrés', apellido='Guzman'. A red checkmark is present in the bottom right corner.

SE MODIFICA TABLA CURSOSS

The screenshot shows the MySQL Workbench interface. On the left, the database schema includes 'cursos', 'cursos', 'direccion', 'direcciones', 'facturas', 'instructores', 'productos', 'tbl_alumnos_cursos', 'tbl_clientes_direcciones', and 'usuarios'. The 'cursos' table is selected, and its data grid shows nine rows of course data. A red arrow points from the 'instructores' table in the previous screenshot to the 'instructores' column in this table's data grid.

SE AGREGA LA RELACION

The screenshot shows the MySQL Workbench interface for creating a foreign key constraint. The 'Foreign Keys' tab is selected for the 'cursos' table. A new foreign key 'fk_cursos_instructores' is being defined, referencing the 'instructores' table. The 'Column' dropdown lists 'id', 'nombre', 'descripcion', 'instructor_id', and 'duracion', with 'instructor_id' checked. The 'Referenced Column' dropdown shows 'id' from the 'instructores' table. Under 'Foreign Key Options', 'On Update' and 'On Delete' are set to 'NO ACTION'. A red box highlights the 'Foreign Keys' tab at the bottom of the window.

D5_73_583_webapp-jaxrs-rest-relacionesEntreTablas

webapp-jaxrs-rest-2

LEVANTA WILDFLY:

```
PS C:\developjava\wildfly-preview-26.1.3.Final\bin> cd C:\developjava\wildfly-preview-26.1.3.Final\bin  
PS C:\developjava\wildfly-preview-26.1.3.Final\bin> .\standalone.bat
```

```
axel  
12345
```

DIFERENCIA ENTRE LAZY Y EAGER :

SI NO SE PONE EL ATRIBUTO FETCH EN LA ANOTACION, POR DEFECTO SERÁ EAGER, YA Q MANYTOONE TERMINA EN "ONE".
AL DEJARLO EN LAZY, MANUALMENTE, HAY Q TRAER AL CURSO Y AL INSTRUCTOR EN LA MISMA CONSULTA:

```
return em.createQuery("select c from Curso c left outer join fetch c.instructor", Curso.class).getResultList();  
return em.createQuery("select c from Curso c left outer join fetch c.instructor where c.id=:id", Curso.class).setParameter("id", id).getSingleResult();
```

```
//@XmlTransient //3  
//@JsonbTransient //4  
@ManyToOne(fetch = FetchType.LAZY) //2  
private Instructor instructor;
```

ANOTACIONES PARA QUITAR CAMPOS EN EL JSON O EL XML :

```
@XmlElementRootElement //1  
@Entity  
@Table(name="cursos")// si , con doble ss  
public class Curso {  
  
    @Id  
    @GeneratedValue(strategy = GenerationType.IDENTITY)  
    private Long id;  
  
    private String nombre;  
  
    private String descripcion;  
  
    //@XmlTransient //3  
    //@JsonbTransient //4  
    @ManyToOne(fetch = FetchType.LAZY) //2  
    private Instructor instructor;  
  
    private Double duracion;  
  
    public Curso() {  
    }
```

HAY UNA ANOTACION MEJOR PARA QUITAR CAMPOS DEL JSON:

AGREGA ESTA DEPENDENCIA:

```
<dependency>  
    <groupId>org.jboss.resteasy</groupId>  
    <artifactId>resteasy-jackson2-provider</artifactId>  
    <version>6.2.3.Final</version>  
    <scope>provided</scope>  
</dependency>
```

Le pones provided ya que el servidor de aplicaciones ya la incluye.

```
29  
30    private Double duracion;  
31  
32    //@XmlTransient //3  
33    //@JsonbTransient //4  
34    @JsonIgnore //5  
35    @ManyToOne(fetch = FetchType.LAZY) //2  
36    private Instructor instructor;  
37  
38    private Double duracion;  
39  
40    public Curso() {  
41    }
```

DESARROLLANDO AL PROYECTO CLIENTE QUE CONSUME EL APIREST CON LA RELACION ENTRE TABLAS:

D5_73_583_cliente-jaxrs-rest-relacionesEntreTablas

CAMBIOS EN ESTA VERSION DEL PROYECTO CLIENTE:

```
@XmlRootElement  
public class Curso {  
  
    private Long id;  
    private String nombre;  
    private String descripcion;  
    private Instructor instructor; ←  
    private Double duracion;  
  
    public Long getId() { return id; }
```

```
1 package cliente.jaxrs.rest.models;
2
3 public class Instructor {
4     private Long id;
5     private String nombre;
6     private String apellido;
7
8     public Long getId() {
9         return id;
10    }
11
12    public void setId(Long id) {
13        this.id = id;
14    }
15
16    public String getNombre() {
17        return nombre;
18    }
19}
```

< | Result Grid | | Filter Rows: | Edit

	id	nombre	apellido
▶	1	Andrés	Guzman
2	Pepe	Doe	
*	NULL	NULL	NULL

RECIBIENDO RESPUESTA DESDE EL WEB SERVICE

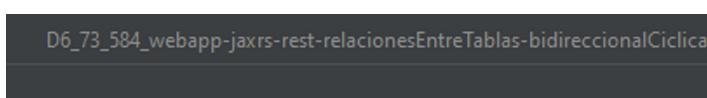
DS_73_583_cliente-jaxrs-rest-relacionesEntreTablas - Main.java

```
public static void main(String[] args) {  
    Client client = ClientBuilder.newClient();  
    WebTarget rootUri = client.target("http://localhost:8080/webapp-jaxrs-rest-2/api").path("/cursos-json"); //1  
  
    /*  
    System.out.println("===== POR ID =====");  
    Curso curso = rootUri.path("/2").request(MediaType.APPLICATION_JSON).get(Curso.class);  
    System.out.println(curso);  
    */  
  
    System.out.println("===== POR ID =====");  
    Response response = rootUri.path("/2").request(MediaType.APPLICATION_JSON).get();  
    Curso curso = response.readEntity(Curso.class);  
    System.out.println(curso);  
    System.out.println(response.getStatus() + " / " + response.getMediaType() + " / " + response.getLocation() + " / " + response.getDate() + " / " + response.  
  
    System.out.println("===== LISTAR =====");  
    List<Curso> cursos = rootUri.request(MediaType.APPLICATION_JSON).get(Response.class).readEntity(new GenericType<List<Curso>>(){}); //2  
    cursos.forEach(System.out::println);  
  
    System.out.println("===== CREANDO =====");  
    Curso cursoNuevo = new Curso();  
    cursoNuevo.setNombre("Spring cloud");  
    cursoNuevo.setDescripcion("Spring cloud - microservicios");  
    cursoNuevo.setDuracion(250);  
    Instructor instructor = new Instructor();  
    instructor.setId(2L); //ESTA LÍNEA ES LA NECESARIA  
    instructor.setNombre("Pepe"); //ESTA LÍNEA NO ES NECESARIA. SOLO PARA Q QUEDA MÁS COMPLETO.  
    instructor.setApellido("Doe"); //ESTA LÍNEA NO ES NECESARIA. SOLO PARA Q QUEDA MÁS COMPLETO.  
    cursoNuevo.setInstructor(instructor);  
}  
  
Run: Main x  
Curso{id=8, nombre='Spring cloud', descripcion='Spring cloud - microservicios', instructor='Instructor{id=1, nombre='Andrés', apellido='Guzman'}', duracion=25.0}  
Curso{id=9, nombre='Microservicios con SpringCloud Eureka', descripcion='Spring cloud - microservicios', instructor='Instructor{id=1, nombre='Andrés', apellido='Guzman'}', duracion=25.0}  
Curso{id=11, nombre='Spring cloud', descripcion='Spring cloud - microservicios', instructor='Instructor{id=2, nombre='Pepe', apellido='Doe'}', duracion=25.0}  
===== EDITANDO =====  
Curso{id=11, nombre='Microservicios con SpringCloud Eureka', descripcion='Spring cloud - microservicios', instructor='Instructor{id=2, nombre='Pepe', apellido='Doe'}', duracion=25.0}  
Curso{id=1, nombre='java', descripcion='java', instructor='Instructor{id=1, nombre='Andrés', apellido='Guzman'}', duracion=100.0}  
Curso{id=2, nombre='javabee', descripcion='jakarta EE9', instructor='Instructor{id=1, nombre='Andrés', apellido='Guzman'}', duracion=100.0}  
Curso{id=3, nombre='angular', descripcion='angular', instructor='Instructor{id=2, nombre='Pepe', apellido='Doe'}', duracion=40.0}  
Curso{id=4, nombre='spring', descripcion='spring boot', instructor='Instructor{id=1, nombre='Andrés', apellido='Guzman'}', duracion=40.0}  
Curso{id=5, nombre='React', descripcion='react js', instructor='Instructor{id=2, nombre='Pepe', apellido='Doe'}', duracion=50.0}  
Curso{id=7, nombre='zzzzzz', descripcion='Python', instructor='Instructor{id=1, nombre='Andrés', apellido='Guzman'}', duracion=70.0}
```

- CLASE 584 -

584. Ejemplo JAX-RS con relaciones cíclica de objetos y problema de recursividad

SE COPTA PROYECTO ANTERIOR



wehann-jaxrs-rest-3

EXPLICACIONES EN EL PROYECTO

File Edit View Navigate Code Refactor Build Run Tools Git Window Help D6_73_584_webapp-jaxrs-rest-relacionesEntreTablas-bidireccionalCiclica - Instructor.java

```

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

//584
//1 - CURSO ES LA CLASE DUEÑA DE LA REALACION.
//2 - EAGER : SE PONE EAGER PARA NO QUEDARNOS SIN CONEXION. .... LUEGO SE QUITA , YA Q CON LA ANOTACION @JsonIgnoreProperties({"cursos"}) EN LA CLASE CURSO, YA NO YA A TRAER EN ESTA CLASE LOS CURSOS DEL INSTRUCTOR.
//3 - CascadeType.ALL : SI ELIMINO UN INSTRUCTOR, Q ESTE SE ELIMINE DEL CURSO.
//4 - "instructor" : TIENIENDO ESTA ANOTACION EN LA CLASE CURSO, ES SUFFICIENTE, PERO SE RECOMIENDA EN AMBOS LADOS.
//5 - "handler", "hibernateLazyInitializer" : COMO EL LAZY TRABAJA CON PROXY, A VECES SE ALMACENA BASURA EN EL CACHE Y ESTA BASURA LA TENEMOS QUE IGNORAR, PQ A VECES ESTO GERA UN BUG EN LAS VERSIONES DE JPA.

@Entity
@Table(name="instructores")
public class Instructor {

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;
    private String nombre;
    private String apellido;

    @JsonIgnoreProperties({"instructor", "handler", "hibernateLazyInitializer"}) //4 //5
    @OneToMany(mappedBy = "instructor", cascade = CascadeType.ALL /*,fetch = FetchType.EAGER*/) //1 //2 //3
    private List<Curso> cursos;

    public Instructor() {
        this.cursos = new ArrayList<>();
    }

    public Long getId() {
        return id;
    }
}

```

File Edit View Navigate Code Refactor Build Run Tools Git Window Help D6_73_584_webapp-jaxrs-rest-relacionesEntreTablas-bidireccionalCiclica - Curso.java

```

// 579
//1 - ESTA ES PARA EL CONTROLADOR REST, CUANDO EMITE RESPUESTAS DEL TIPO XML, PARA Q GENERE CORRECTAMENTE EL XML.
//583
//2 - UN CURSO PUEDE TENER UN INSTRUCTOR, PERO UN INSTRUCTOR PUEDE TENER VARIOS CURSOS.
//3 - PARA NO INCLUIR ESTE CAMPO EN EL XML.
//4 - PARA NO INCLUIR ESTE CAMPO EN EL JSON.
//5 - ESTA ESTA MEJOR PARA IGNORAR UN CAMPO EN EL JSON.
//584
//6 - CURSO ES EL DUEÑO DE LA RELACION
//7 - "cursos" : PARA ELIMINAR EL LOOP INFINTO ENTRE CURSO E INSTRUCTOR SE USA ESTA ANOTACION DE LA DEPENDENCIA resteasy-jackson2-provider : @JsonIgnoreProperties . con {"cursos"}, SE LE DICE Q OMITA EL ATRIBUTO CURSOS Q ESTA EN EL JSON.
//8 - "handler", "hibernateLazyInitializer" : COMO EL LAZY TRABAJA CON PROXY, A VECES SE ALMACENA BASURA EN EL CACHE Y ESTA BASURA LA TENEMOS QUE IGNORAR, PQ A VECES ESTO GERA UN BUG EN LAS VERSIONES DE JPA.

@XmlRootElement //1
@Entity
@Table(name="cursos")// si , con doble ss
public class Curso {

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;

    private String nombre;

    private String descripcion;

    //@XmlTransient //3
    //@JsonTransient //4
    //@JsonIgnore //5
    @JsonIgnoreProperties({"cursos", "handler", "hibernateLazyInitializer"}) //7 //8
    @ManyToOne(fetch = FetchType.LAZY) //2 //6
    private Instructor instructor;

    private Double duracion;

    public Curso() {
    }

    public Curso(String nombre) { this.nombre = nombre; }

    public String getNombre() { return nombre; }
}

```

File Edit View Navigate Code Refactor Build Run Tools Git Window Help D6_73_584_webapp-jaxrs-rest-relacionesEntreTablas-bidireccionalCiclica - Instructor.java

My Workspace

- Java master jakarta EE9
- mine :8
- mine : microservicios+angular
- mine : ms-microservices
- mine : registro-usuarios-back
- mine : webflux
- ms-docker-kubernetes
- Spring_Con_Angular_webApp_Fullstack

localhost:8080/webapp-jaxrs-rest-3/api/cursos-json

GET localhost:8080/webapp-jaxrs-rest-3/api/cursos-json

Key	Value	Description
Key	Value	Description

Body

```

60     "descripcion": "Python",
61     "instructor": {
62         "id": 1,
63         "nombre": "Andrés",
64         "apellido": "Guzman"
65     },
66     "duracion": 70.0
67 },
68 [
69     {
70         "id": 8,
71         "nombre": "Spring cloud",
72         "descripcion": "Spring cloud - microservicios",
73         "instructor": {
74             "id": 1,
75             "nombre": "Andrés",
76             "apellido": "Guzman"
77         },
78         "duracion": 25.0
79     },
80     {
81         "id": 9,
82         "nombre": "Microservicios con SpringCloud Eureka",
83         "descripcion": "Spring cloud - microservicios",
84         "instructor": {
85             "id": 1,
86             "nombre": "Andrés",
87             "apellido": "Guzman"
88         },
89         "duracion": 25.0
90     },
91     {
92         "id": 11,
93         "nombre": "Microservicios con SpringCloud Eureka",
94         "descripcion": "Spring cloud - microservicios",
95     }
96 ]

```

NO HAY PROYECTO CLIENTE PARA ESTA VERSION DEL PROYECTO.

SE PUEDE USAR ESTE,

D5_73_583_cliente-jaxrs-rest-relacionesEntreTablas

... . PERO CAMBIANDO EN LA URL DE CONEXIÓN EN EL MAIN webapp-jaxrs-rest-2 por webapp-jaxrs-rest-3.

FIN...