

## MANUAL PARA DESARROLLAR ENDPOINTS EN JAX RS / JERSEY

Escrito por José Morejón

Descargar eclipse

<https://www.eclipse.org/downloads/packages/release/photon/r/eclipse-ide-java-ee-developers>

The easiest way to install and update your Eclipse Development Environment.

# Eclipse IDE for Java EE Developers

## Package Description

Tools for Java developers creating Java EE and Web applications, including a Java IDE, tools for Java EE, JPA, JSF, Mylyn, EGit and others.

This package includes:

- Data Tools Platform
- Git integration for Eclipse
- Eclipse Java Development Tools
- Eclipse Java EE Developer Tools
- JavaScript Development Tools
- Maven Integration for Eclipse
- Mylyn Task List
- Eclipse Plug-in Development Environment
- Code Recommenders Tools for Java Developers
- Eclipse XML Editors and Tools

► Detailed features list

## Download Links

Windows 32-bit  
Windows 64-bit  
Mac OS X (Cocoa) 64-bit  
Linux 32-bit  
Linux 64-bit

Downloaded 708,077 Times

► Checksums...

## Bugzilla

► Open Bugs: 72



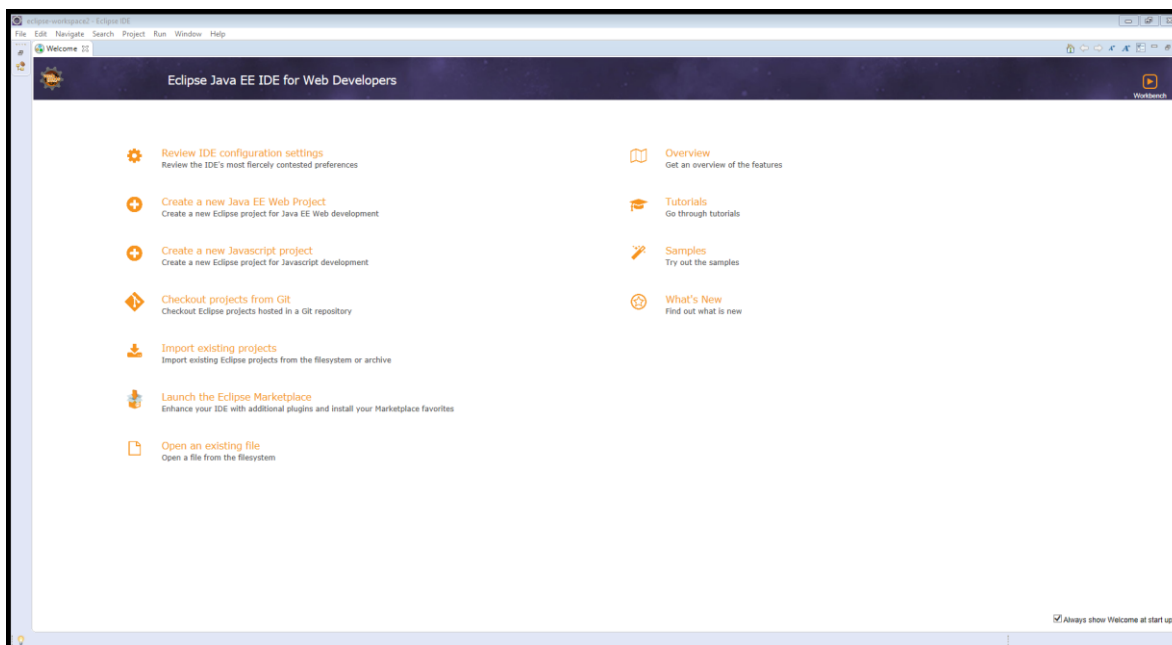
Adjunto captura de pantalla para el ide que estoy utilizando

# MANUAL PARA DESARROLLAR ENDPOINTS EN JAX RS / JERSEY

Escrito por José Morejón

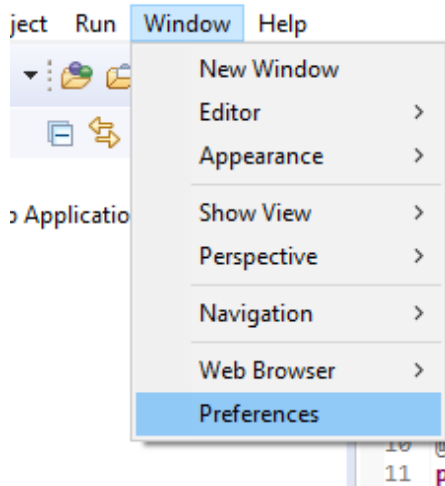
## Inicio

Inicialmente tienen el workspace



## Configurar el JDK

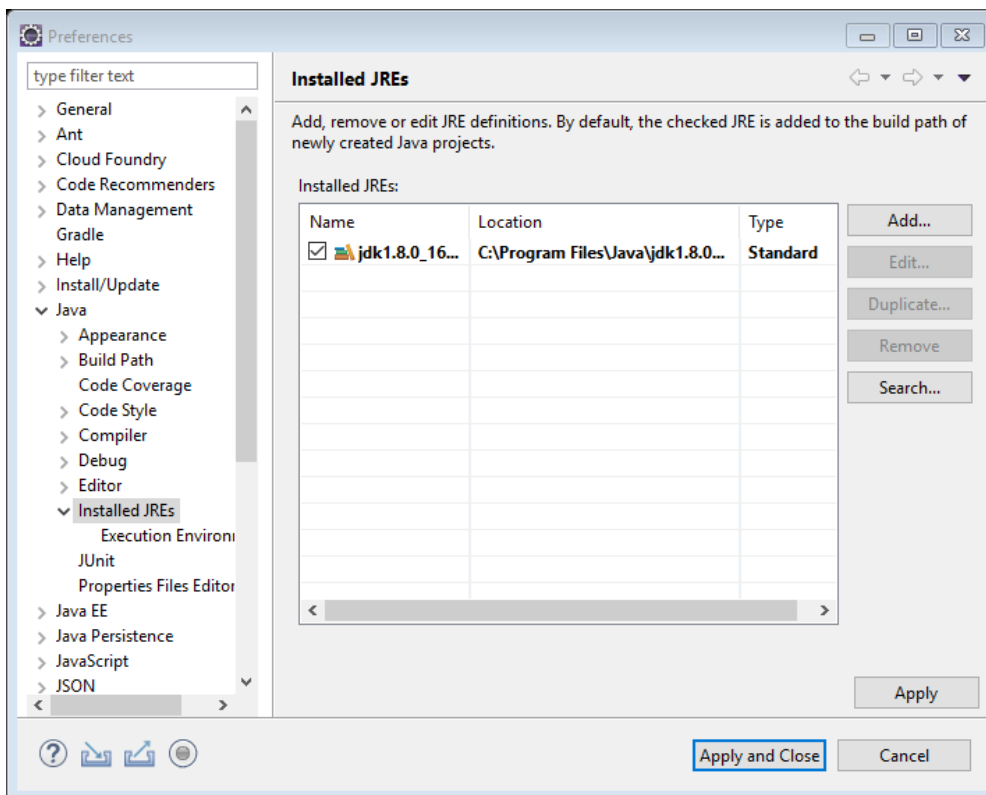
...ed/Services/ServiceClient.java - Eclipse IDE



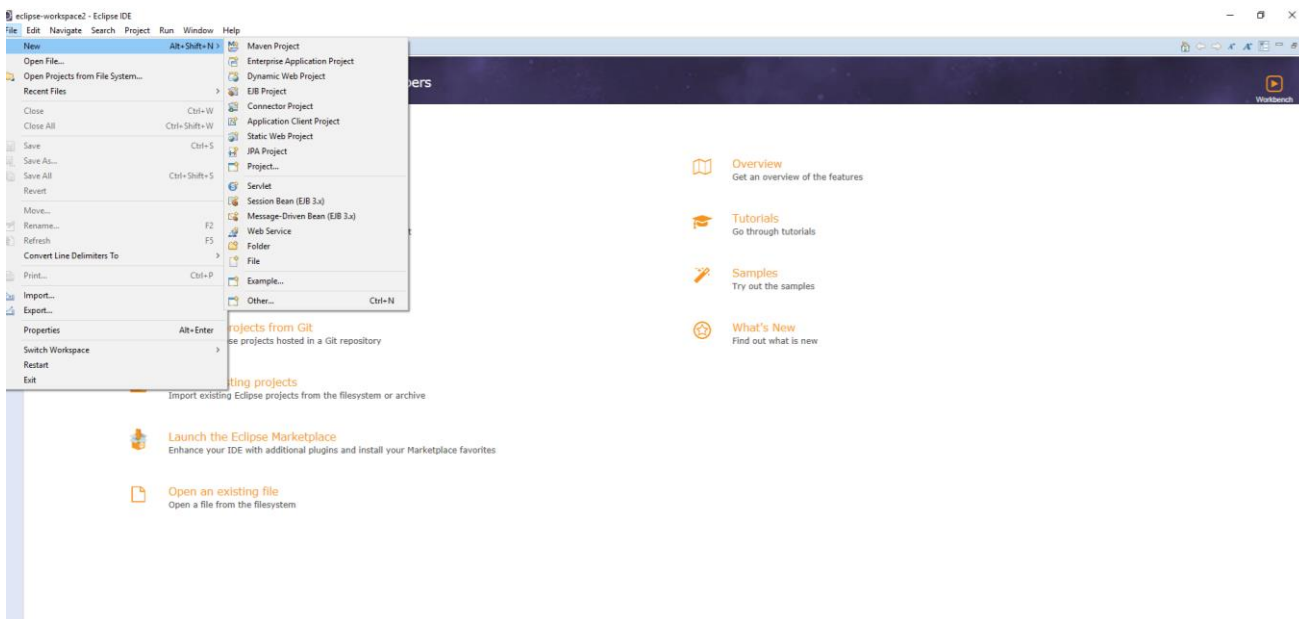
Aquí deben de buscar el path donde tienen al buen JDK dado que el IDE eclipse tiene por defecto al jre. Deben cambiarlo por el jdk como en la siguiente figura.

## MANUAL PARA DESARROLLAR ENDPOINTS EN JAX RS / JERSEY

Escrito por José Morejón



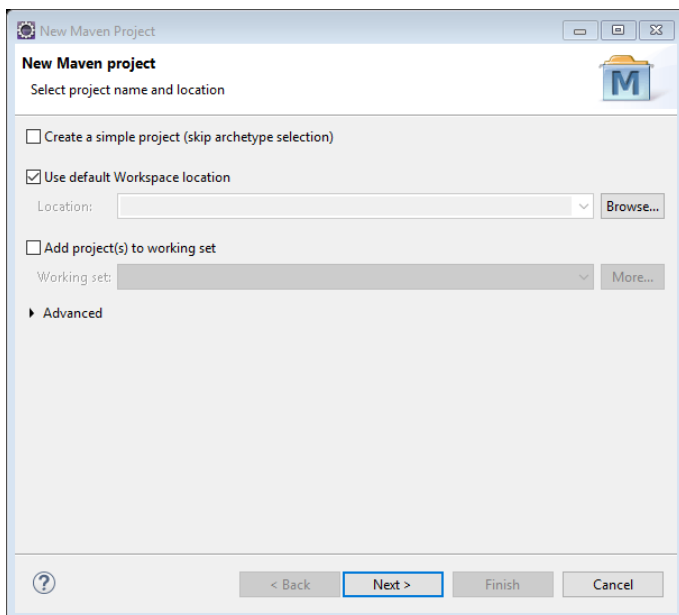
En el apartado de File crean un proyecto maven (Maven Project)



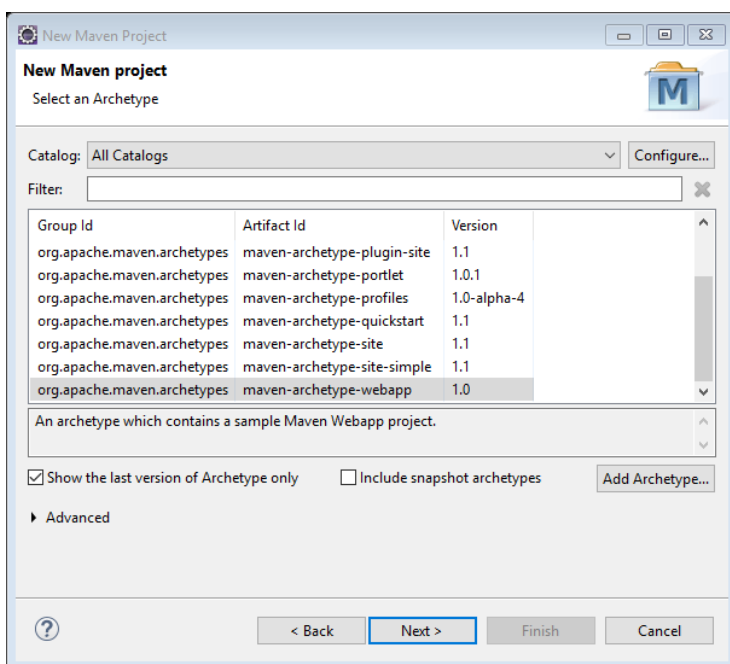
Next

# MANUAL PARA DESARROLLAR ENDPOINTS EN JAX RS / JERSEY

Escrito por José Morejón



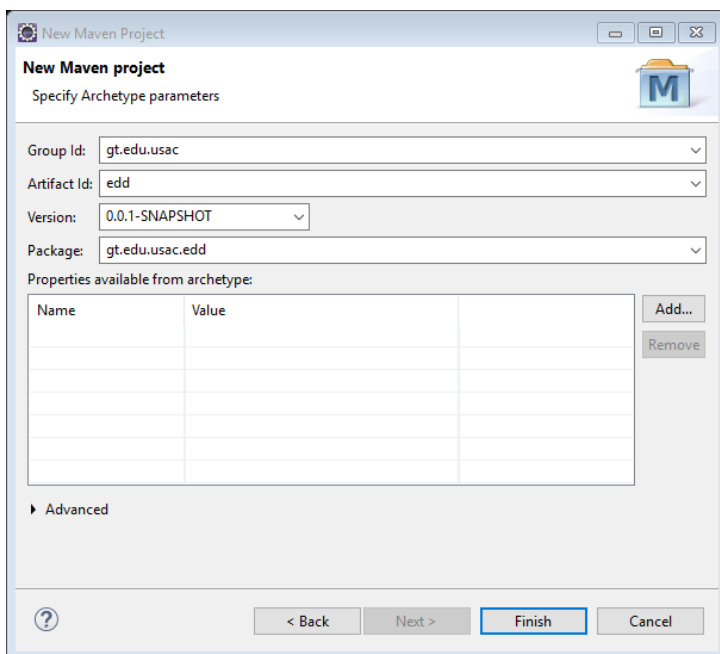
Seleccionamos maven-archetype-webapp



# MANUAL PARA DESARROLLAR ENDPOINTS EN JAX RS / JERSEY

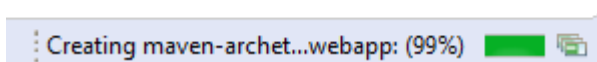
Escrito por José Morejón

Especificamos el Archetype



Al darle finalizar deben de estar conectados al internet para poder descargar las dependencias, caso contrario puede dar algunos errores.

☒ Always show Welcome at start up



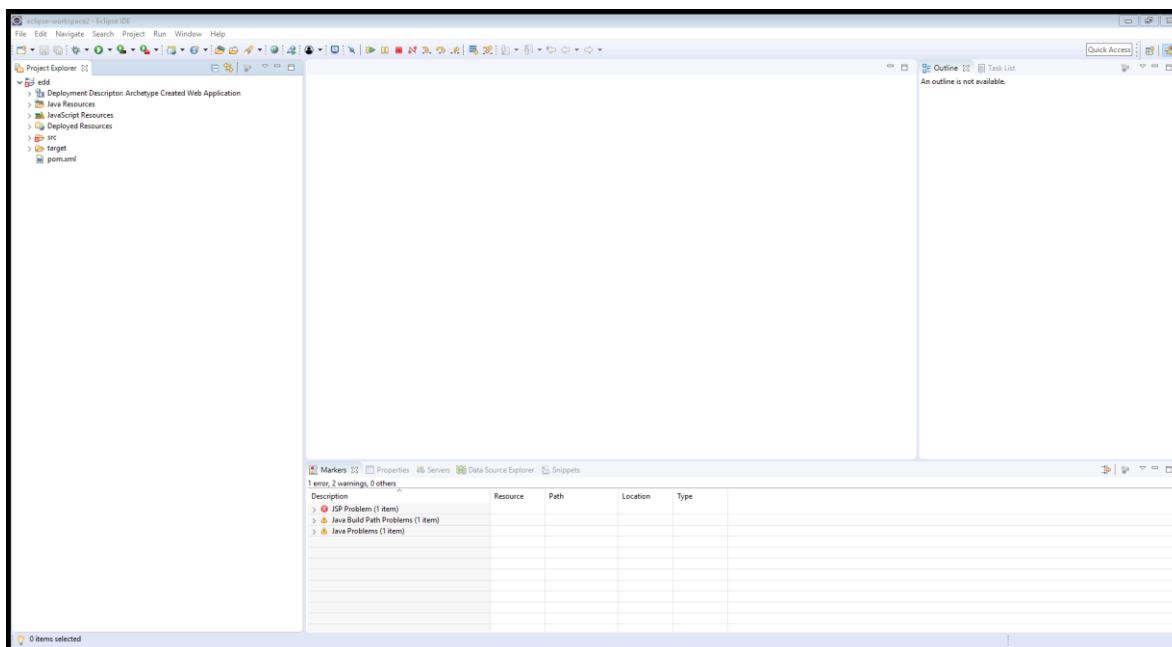
Cerramos welcome



Como pueden observar en la siguiente imagen ya podemos visualizar la carpeta que contiene los fuentes del proyecto.

# MANUAL PARA DESARROLLAR ENDPOINTS EN JAX RS / JERSEY

Escrito por José Morejón



De entrada nos lanza el primer error

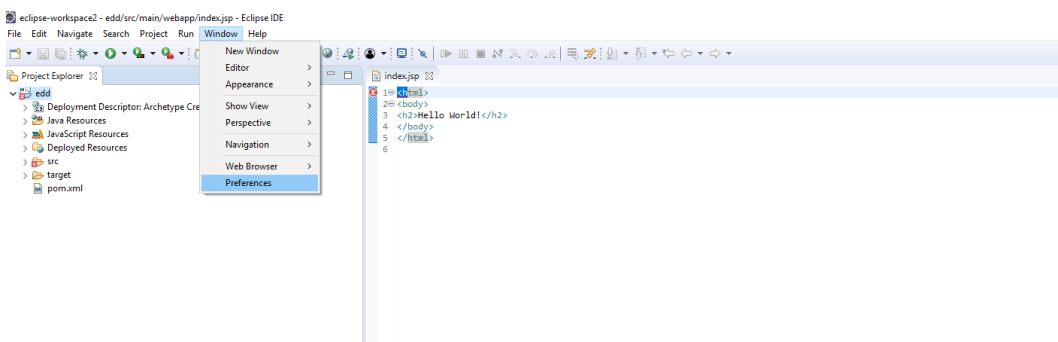
Markers				
1 error, 2 warnings, 0 others				
Description	Resource	Path	Location	Type
✖ JSP Problem (1 item)				
✖ The superclass "javax.servlet.http.HttpServlet"	index.jsp	/edd/src/main/we...	line 1	JSP Problem
> ⚠ Java Build Path Problems (1 item)				
> ⚠ Java Problems (1 item)				

Antes de resolver el error, deben de instalar el servidor de aplicaciones para mi caso apache tomcat.

<https://tomcat.apache.org/download-80.cgi>

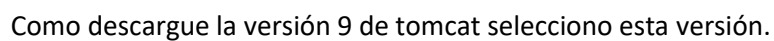
Extraje el comprimido en una carpeta para poder agregarlo a la configuración del proyecto.

Para agregar un servidor, seleccionar la opción window->preferences



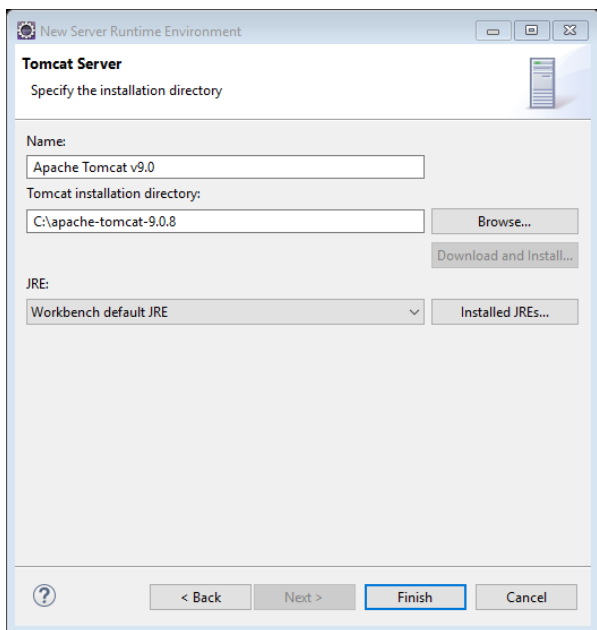
Seleccionar Server->Runtime Environments->Add

## Escrito por José Morejón

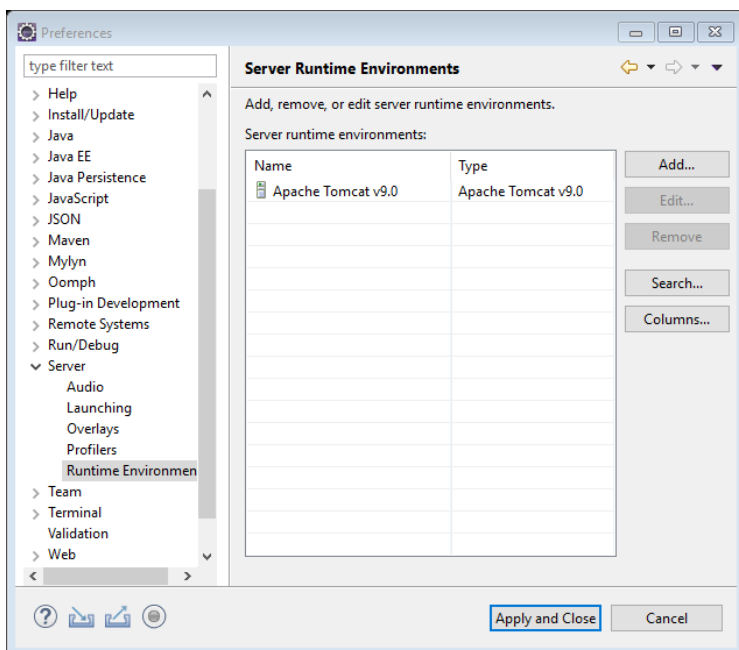


## MANUAL PARA DESARROLLAR ENDPOINTS EN JAX RS / JERSEY

Escrito por José Morejón



Al agregar la ruta, Finish



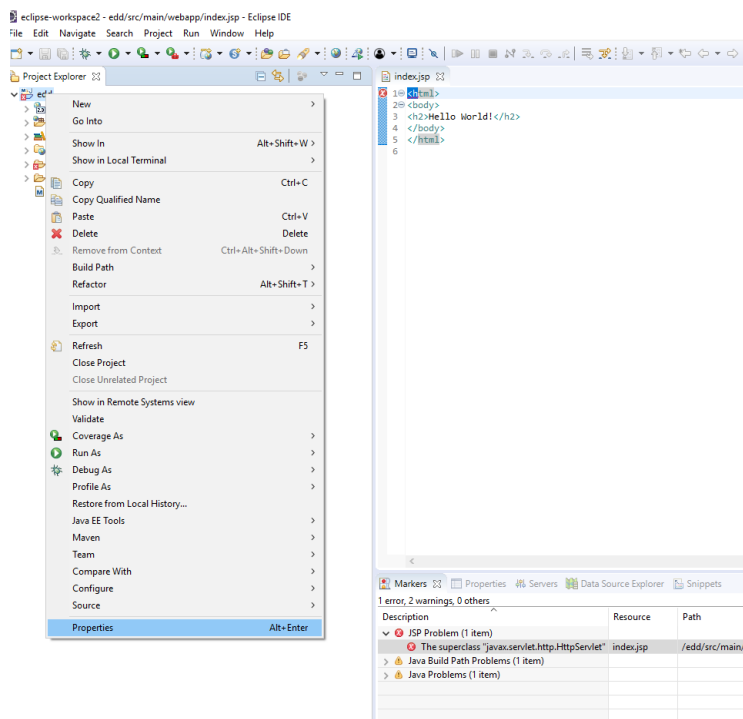
Seleccionar Apply and Close



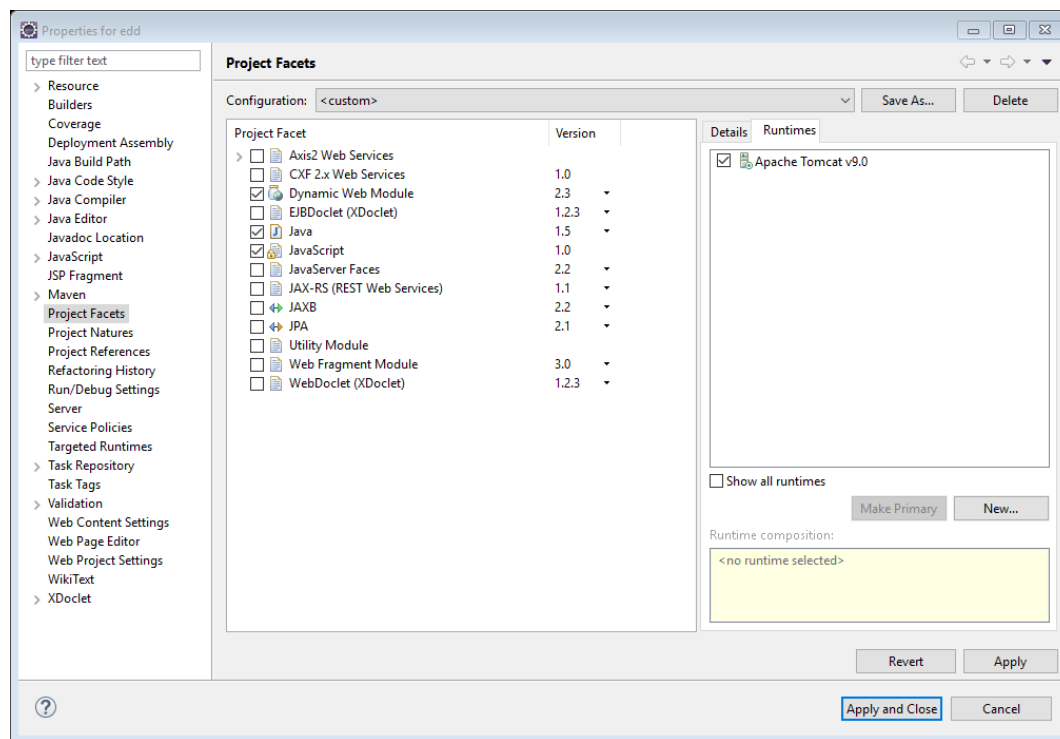
# MANUAL PARA DESARROLLAR ENDPOINTS EN JAX RS / JERSEY

Escrito por José Morejón

Quitar error de jsp: Clic derecho en propiedades de la raíz del proyecto.



Agregar apache tomcat al Project Facets->Runtime

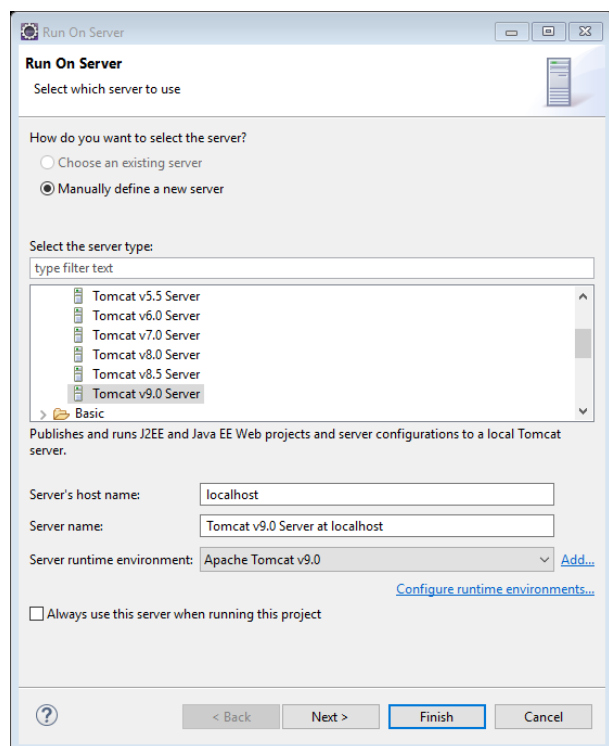
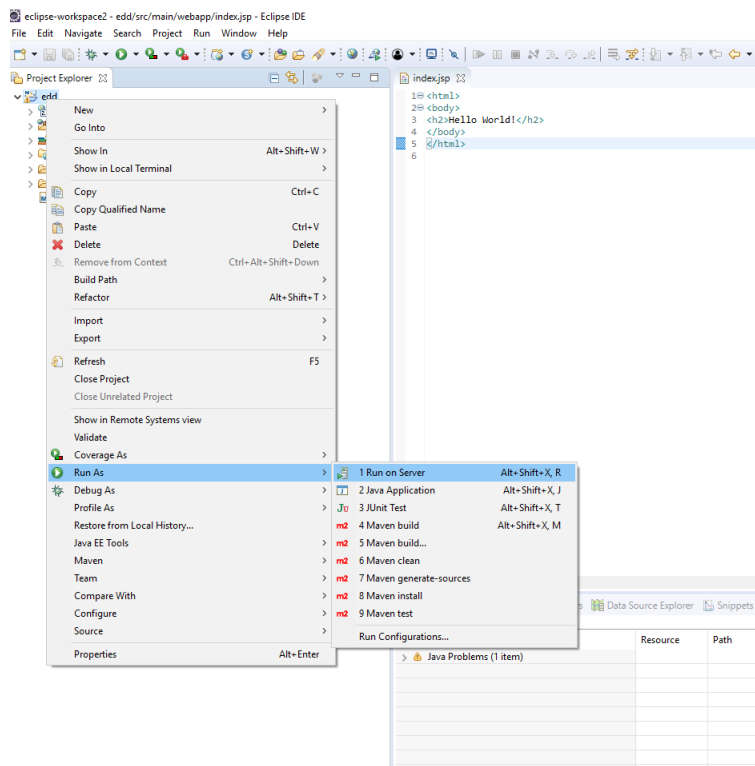


Apply and Close

# MANUAL PARA DESARROLLAR ENDPOINTS EN JAX RS / JERSEY

Escrito por José Morejón

## Construimos el proyecto



Seleccionar el servidor y ejecutamos.

# MANUAL PARA DESARROLLAR ENDPOINTS EN JAX RS / JERSEY

Escrito por José Morejón

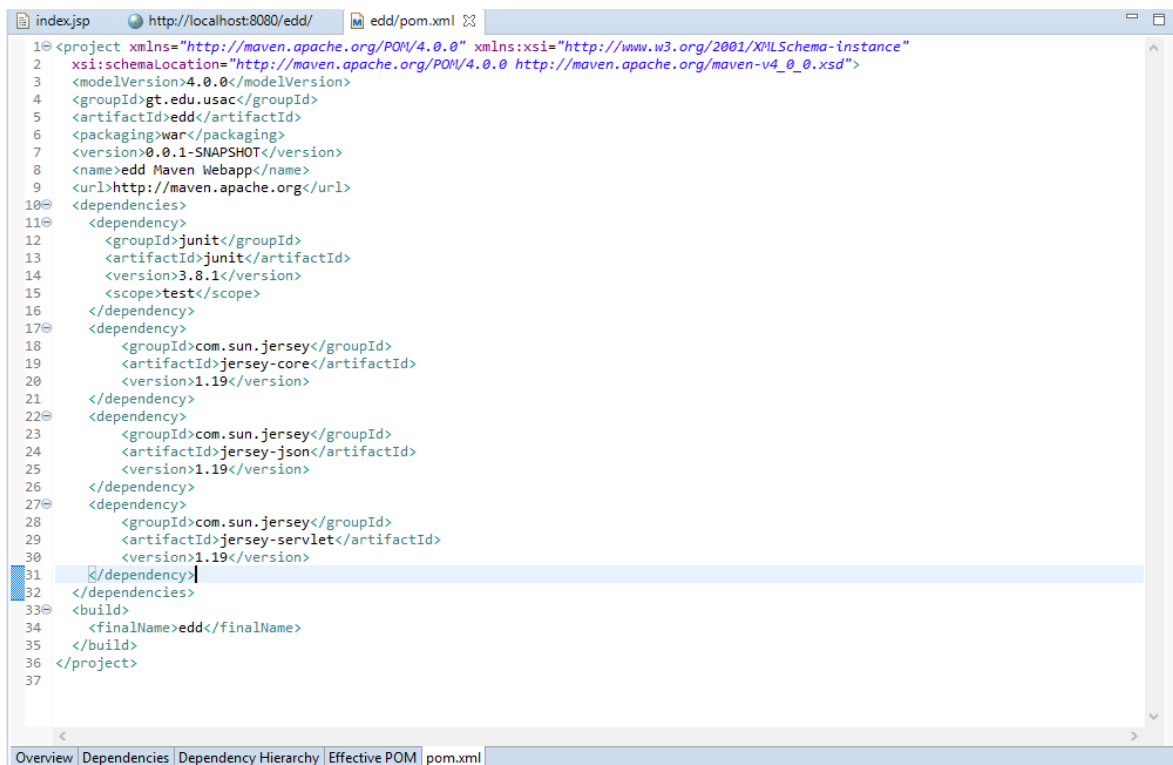
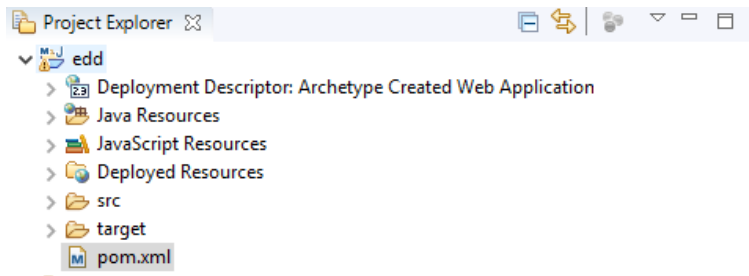
Permitir todas las conexiones que lance el Firewall

Si todo está correcto, puede lanzar un hola mundo.



## Agregar Jersey

Agregar las dependencias al pom.xml

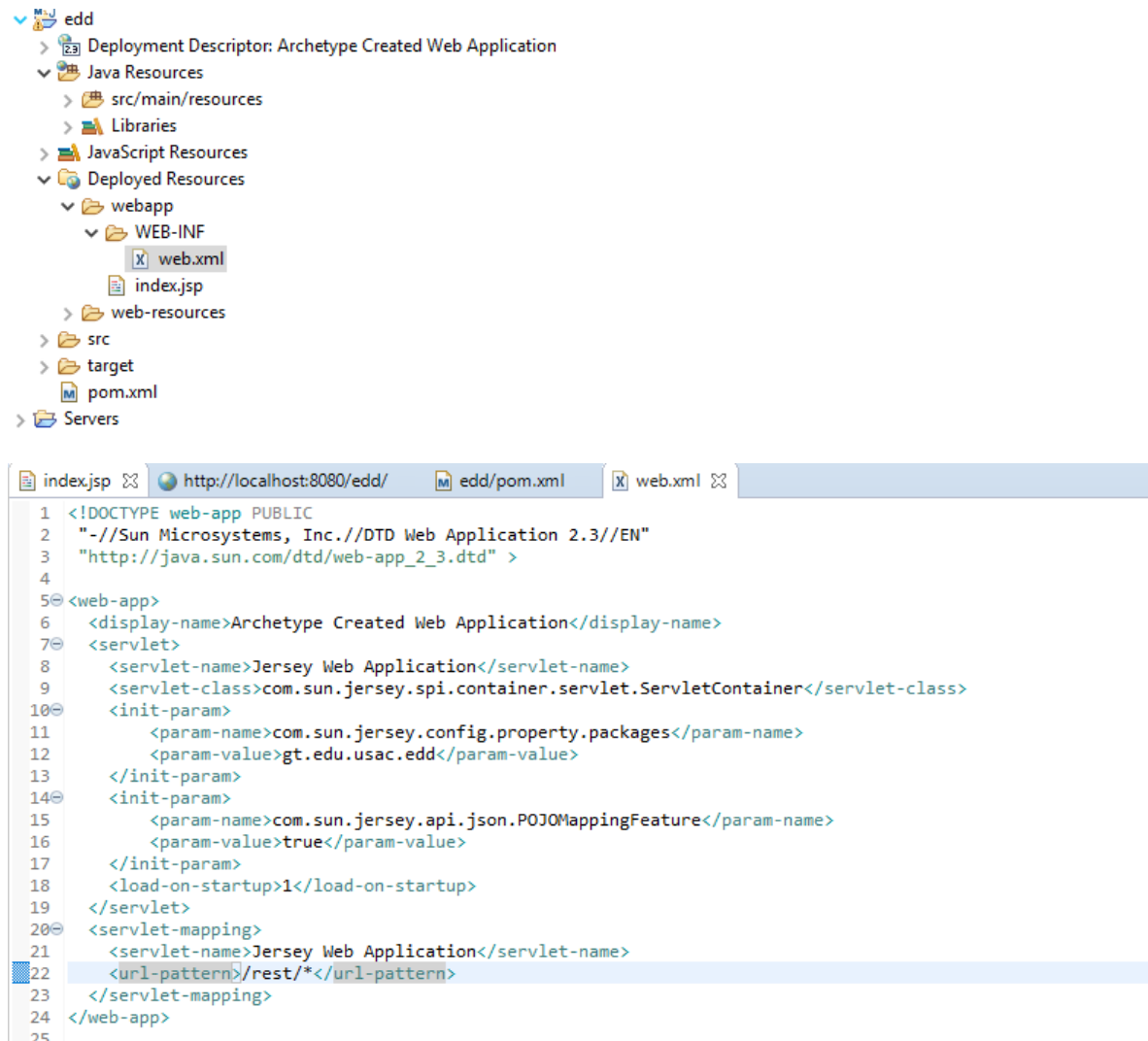


## MANUAL PARA DESARROLLAR ENDPOINTS EN JAX RS / JERSEY

Escrito por José Morejón

### Web.xml

Actualizar el web.xml, declarando el paquete gt.edu.usac.edd y la ruta para acceder a los servicios.  
/rest/\*



The screenshot shows an IDE with a project named 'edd'. The project structure on the left includes 'Deployment Descriptor: Archetype Created Web Application', 'Java Resources', 'JavaScript Resources', 'Deployed Resources', 'webapp', 'WEB-INF', 'web-resources', 'src', 'target', 'pom.xml', and 'Servers'. The 'web.xml' file is selected under 'WEB-INF'. The main editor displays the content of 'web.xml' with the following code:

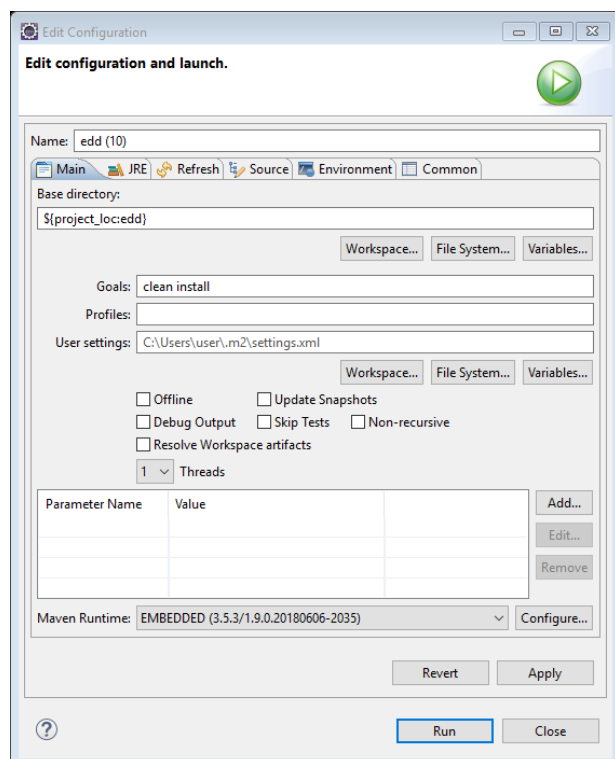
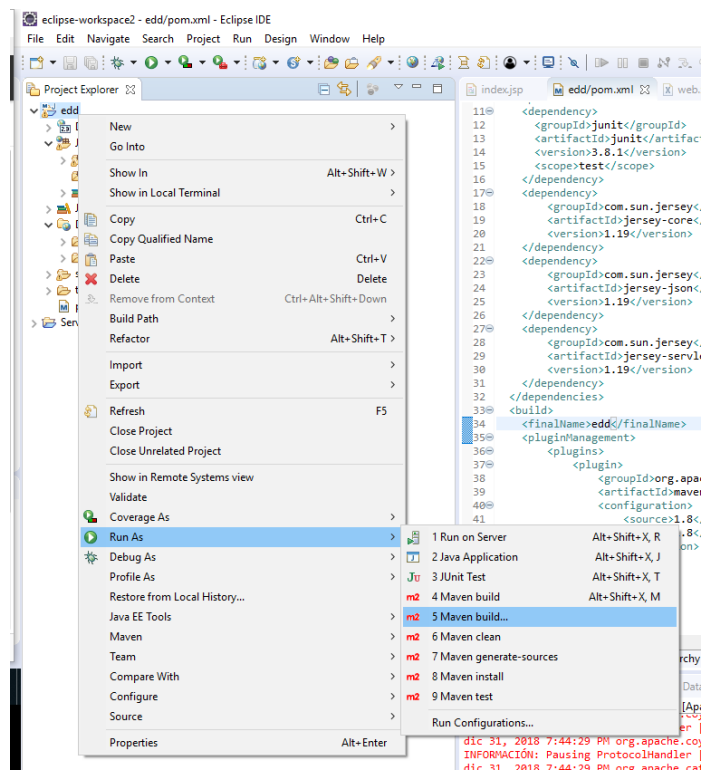
```
1 <!DOCTYPE web-app PUBLIC
2 "-//Sun Microsystems, Inc.//DTD Web Application 2.3//EN"
3 "http://java.sun.com/dtd/web-app_2_3.dtd" >
4
5 <web-app>
6   <display-name>Archetype Created Web Application</display-name>
7   <servlet>
8     <servlet-name>Jersey Web Application</servlet-name>
9     <servlet-class>com.sun.jersey.spi.container.servlet.ServletContainer</servlet-class>
10    <init-param>
11      <param-name>com.sun.jersey.config.property.packages</param-name>
12      <param-value>gt.edu.usac.edd</param-value>
13    </init-param>
14    <init-param>
15      <param-name>com.sun.jersey.api.json.POJOMappingFeature</param-name>
16      <param-value>true</param-value>
17    </init-param>
18    <load-on-startup>1</load-on-startup>
19  </servlet>
20  <servlet-mapping>
21    <servlet-name>Jersey Web Application</servlet-name>
22    <url-pattern>/rest/*</url-pattern>
23  </servlet-mapping>
24 </web-app>
25
```

### Ejecutar Maven

Primer paso, es ejecutar el clean install, de igual manera cuando hay algún problema en su ejecución, pueden detener al tomcat para el clean de maven y volver a ejecutar el proyecto.

# MANUAL PARA DESARROLLAR ENDPOINTS EN JAX RS / JERSEY

Escrito por José Morejón



Si todo está bien configurado, deben de visualizar en su consola lo siguiente

# MANUAL PARA DESARROLLAR ENDPOINTS EN JAX RS / JERSEY

Escrito por José Morejón

```
36</plugins>
37<plugin>
38  <groupId>org.apache.maven.plugins</groupId>
39  <artifactId>maven-compiler-plugin</artifactId>
40</plugin>
41  <configuration>
42    <source>1.8</source>
43    <target>1.8</target>
44  </configuration>
45</plugin>
46</plugins>
47</pluginManagement>
48</build>
49</project>
```

Overview | Dependencies | Dependency Hierarchy | Effective POM | pom.xml

terminated> edd (T0) [Maven Build] C:\Program Files\Java\jdk1.8.0\_161\bin\javaw.exe (31/12/2018 19:46:05)

INFO] Webapp assembled in [1017 msecs]

INFO] Building war: C:\Users\user\workspace2\edd\target\edd.war

INFO] WEB-INF\web.xml already added, skipping

INFO] --- maven-install-plugin:2.4.4:install (default-install) @ edd ---

INFO] Installing C:\Users\user\workspace2\edd\target\edd.war to C:\Users\user\.m2\repository\gt\edu\usac\edd\0.0.1-SNAPSHOT\edd-0.0.1-SNAPSHOT.war

INFO] Installing C:\Users\user\workspace2\edd\pom.xml to C:\Users\user\.m2\repository\gt\edu\usac\edd\0.0.1-SNAPSHOT\edd-0.0.1-SNAPSHOT.pom

INFO] BUILD SUCCESS

INFO] Total time: 10.282 s

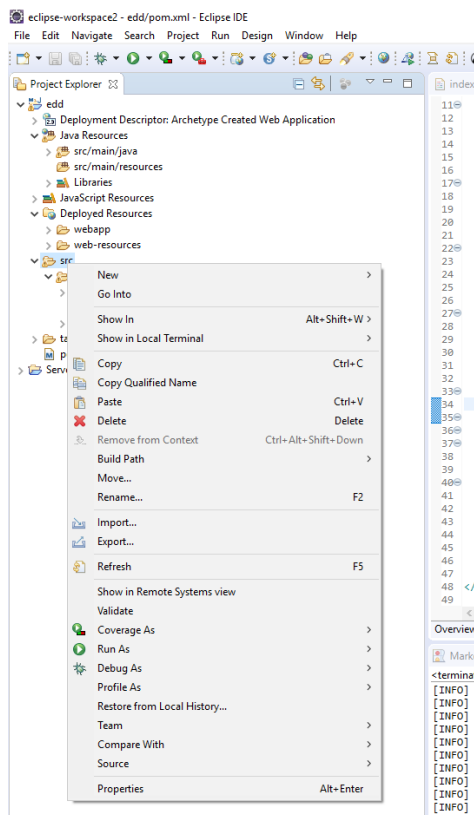
INFO] Finished at: 2018-12-31T19:46:17-06:00

INFO]

## Agregar servicios

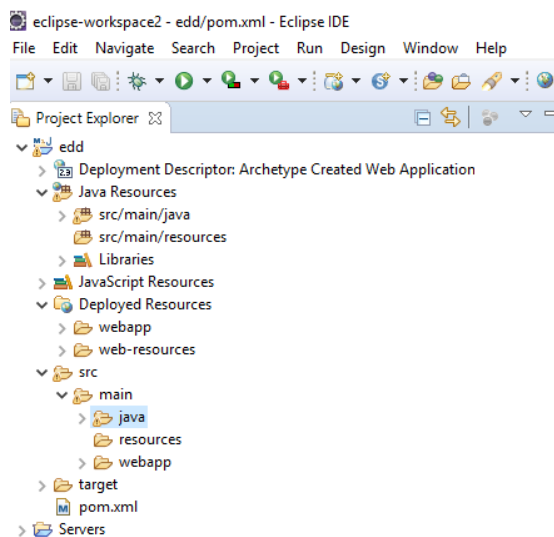
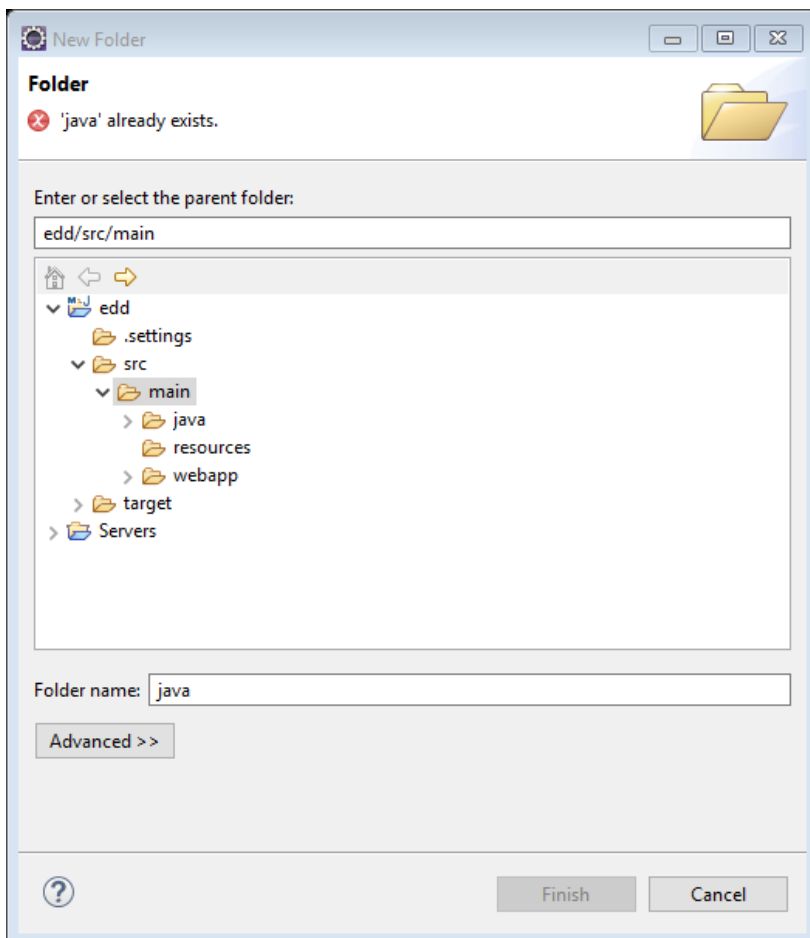
Se deben de crear por lo menos una clase, yo voy a definir dos, una para el servicio y otra para poder observar la clase serializada en JSON.

Deben de crear una carpeta java en el siguiente directorio, src->New->folder



## MANUAL PARA DESARROLLAR ENDPOINTS EN JAX RS / JERSEY

Escrito por José Morejón



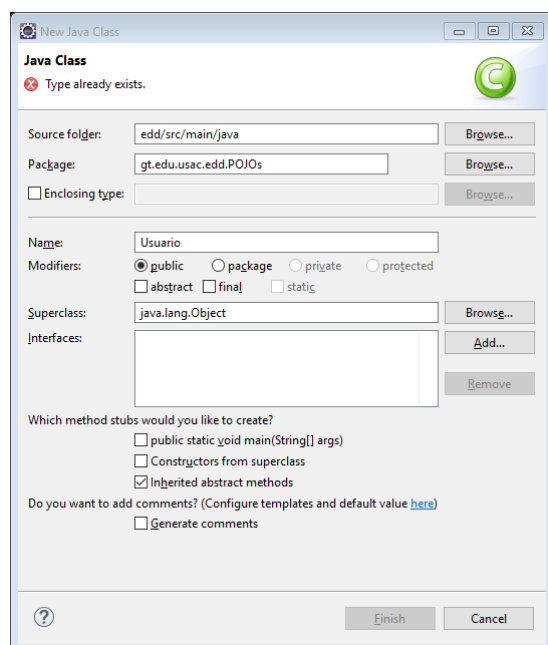
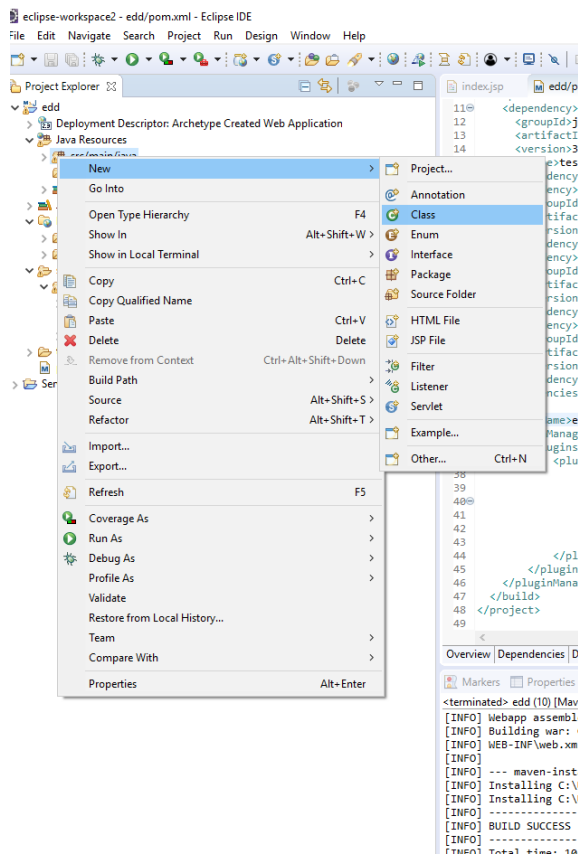
Al crear este folder se genera en el directorio Java Resources la siguiente ruta-> src/main/java.

Ahora este es nuestro workspace donde construiremos nuestros paquetes para trabajar en java.

# MANUAL PARA DESARROLLAR ENDPOINTS EN JAX RS / JERSEY

Escrito por José Morejón

Pasamos a crear las clases en src/main/java

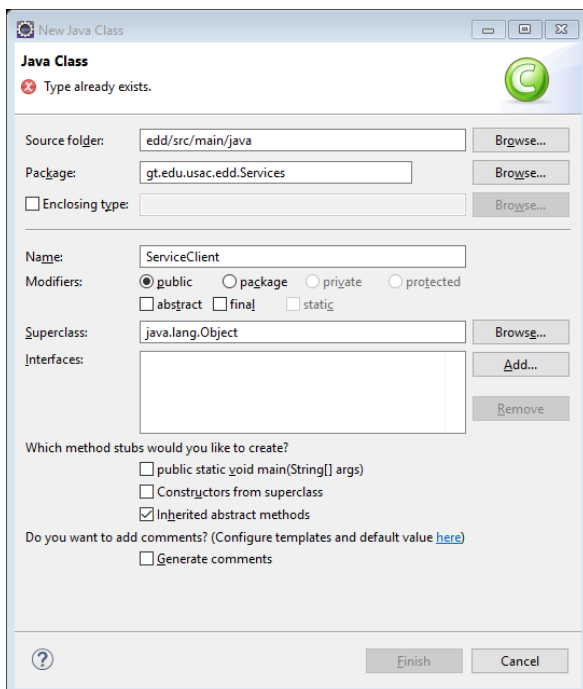


Recuerden que en el web.xml se dejó configurado para que jersey obtenga los recursos de los paquetes nombrados gt.edu.usac.edd

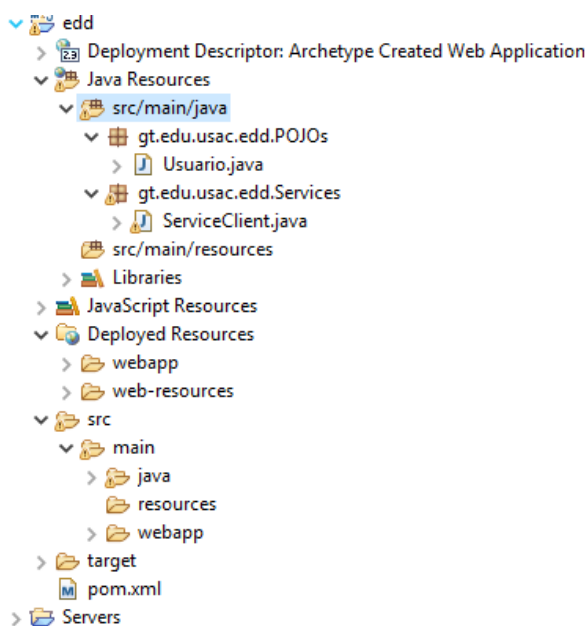


# MANUAL PARA DESARROLLAR ENDPOINTS EN JAX RS / JERSEY

Escrito por José Morejón



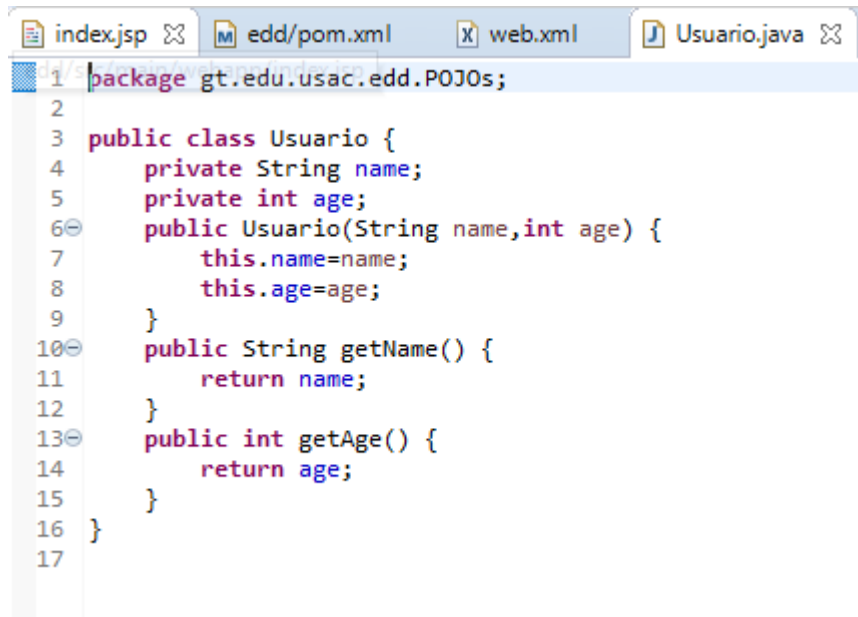
## Resultado en el directorio



## MANUAL PARA DESARROLLAR ENDPOINTS EN JAX RS / JERSEY

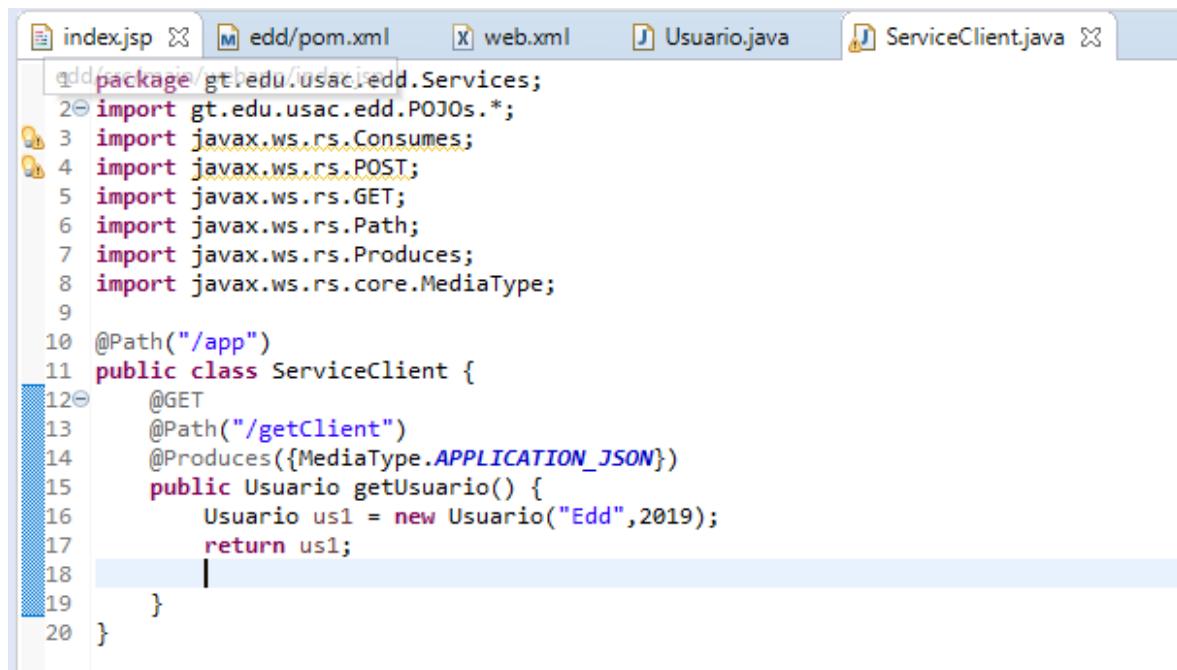
Escrito por José Morejón

### Declaración de la clase usuario



```
1 package gt.edu.usac.edd.POJOs;
2
3 public class Usuario {
4     private String name;
5     private int age;
6     public Usuario(String name,int age) {
7         this.name=name;
8         this.age=age;
9     }
10    public String getName() {
11        return name;
12    }
13    public int getAge() {
14        return age;
15    }
16 }
17
```

### Declaración de la clase ServiceClient



```
1 package gt.edu.usac.edd.Services;
2 import gt.edu.usac.edd.POJOs.*;
3 import javax.ws.rs.Consumes;
4 import javax.ws.rs.POST;
5 import javax.ws.rs.GET;
6 import javax.ws.rs.Path;
7 import javax.ws.rs.Produces;
8 import javax.ws.rs.core.MediaType;
9
10 @Path("/app")
11 public class ServiceClient {
12     @GET
13     @Path("/getClient")
14     @Produces({MediaType.APPLICATION_JSON})
15     public Usuario getUsuario() {
16         Usuario us1 = new Usuario("Edd",2019);
17         return us1;
18     }
19 }
20
```

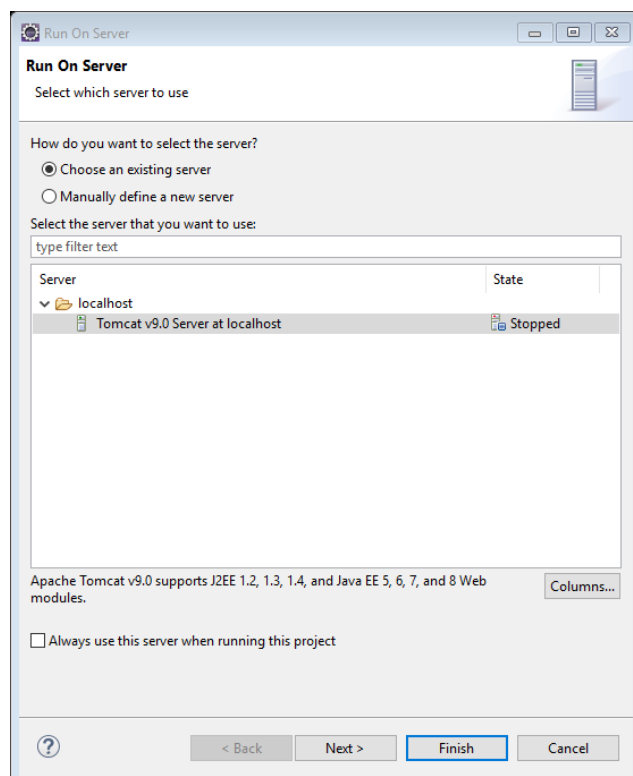
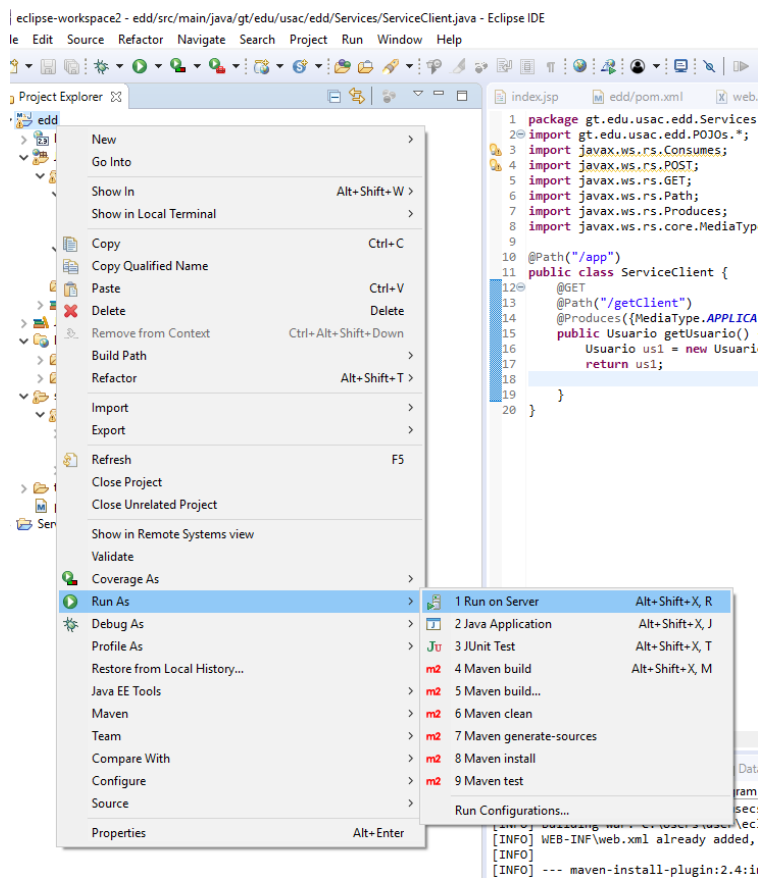
Pequeño resumen del contenido de la clase, en primer lugar se tienen las importaciones, lo relevante sería la parte del javax.ws.rs.Consumes que la utilizamos para pasar objetos relativamente usado para el POST, el ..Produces que retorna los objetos y el MediaType para setear los objetos a JSON, el ..Path ayuda para asignar las rutas tanto de la raíz como la de los endpoints que conforma esa raíz.

En este ejemplo se realiza un get para solo retornar la información quemada en el objeto.

# MANUAL PARA DESARROLLAR ENDPOINTS EN JAX RS / JERSEY

Escrito por José Morejón

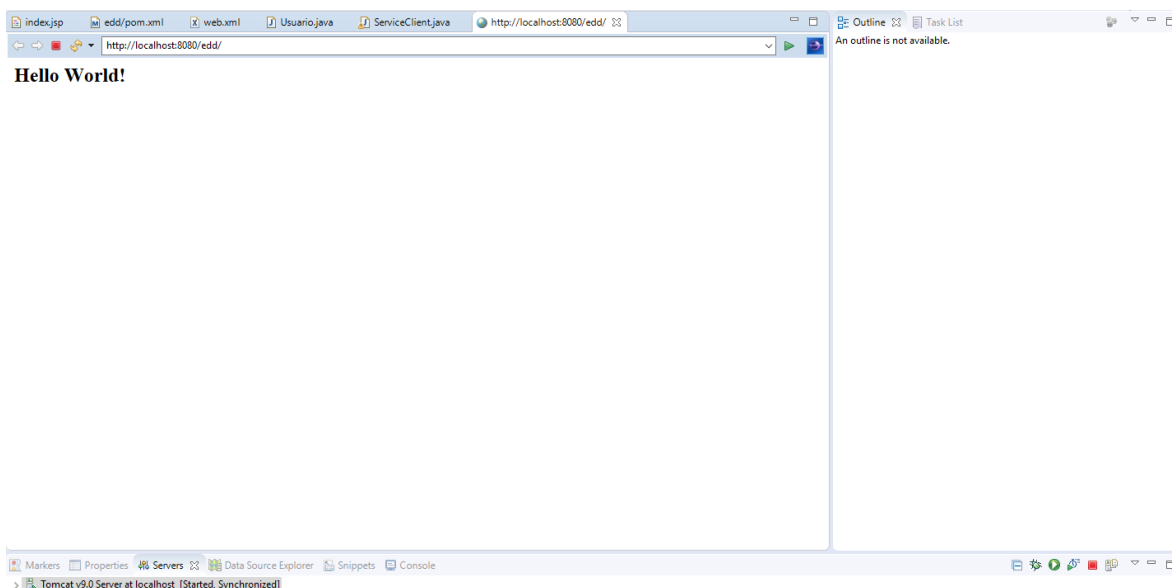
## Corremos la aplicación



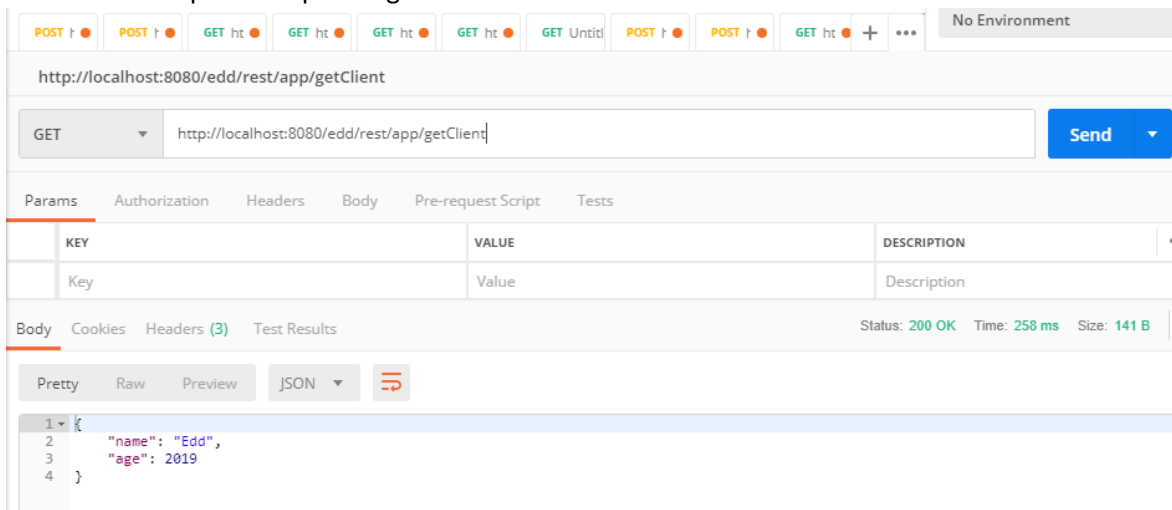
# MANUAL PARA DESARROLLAR ENDPOINTS EN JAX RS / JERSEY

Escrito por José Morejón

Si todo está correcto, muestra el HW.



Vamos al buen postman por un get



Resumiendo la ruta, /edd viene del nombre del proyecto, /rest viene del web.xml, /app al inicio de la clase ServiceClient y por ultimo al getClient que está dentro de la clase ServiceClient.

## Post

Para crear un post agregamos lo siguiente a la clase Usuario

## MANUAL PARA DESARROLLAR ENDPOINTS EN JAX RS / JERSEY

Escrito por José Morejón

```
index.jsp  edd/pom.xml  web.xml  *Usuario.java  X
1 package gt.edu.usac.edd.POJOs;
2
3 public class Usuario {
4     private String name;
5     private int age;
6
7     public Usuario() {}
8
9     public Usuario(String name,int age) {
10         this.name=name;
11         this.age=age;
12     }
13     public String getName() {
14         return name;
15     }
16     public int getAge() {
17         return age;
18     }
19 }
20
21
```

Y a la clase ServiceClient

```
index.jsp  edd/pom.xml  web.xml  Usuario.java  ServiceClient.java  X
1 package gt.edu.usac.edd.Services;
2 import gt.edu.usac.edd.POJOs.*;
3 import javax.ws.rs.Consumes;
4 import javax.ws.rs.POST;
5 import javax.ws.rs.GET;
6 import javax.ws.rs.Path;
7 import javax.ws.rs.Produces;
8 import javax.ws.rs.core.MediaType;
9
10 @Path("/app")
11 public class ServiceClient {
12     @GET
13     @Path("/getClient")
14     @Produces({MediaType.APPLICATION_JSON})
15     public Usuario getUsuario() {
16         Usuario us1 = new Usuario("Edd",2019);
17         return us1;
18     }
19
20     @POST
21     @Path("/validarUser")
22     @Consumes({MediaType.APPLICATION_JSON})
23     @Produces({MediaType.APPLICATION_JSON})
24     public Usuario validarUsuari(Usuario user) {
25         if(user.getName().equals("Edd") && user.getAge()==2019) {
26             return new Usuario("usuario", 2018);
27         }
28         return user;
29     }
30 }
```

# MANUAL PARA DESARROLLAR ENDPOINTS EN JAX RS / JERSEY

Escrito por José Morejón

Vamos nuevamente al buen postman

The image displays two screenshots of the Postman REST client interface, showing the configuration and execution of a POST request to a REST endpoint.

**Top Screenshot:**

- Environment:** No Environment
- URL:** `http://localhost:8080/edd/rest/app/validarUser`
- Method:** POST
- Body Type:** raw (selected), JSON (application/json)
- Body Content:**

```
1 {  
2   "name": "Edd",  
3   "age": 2019  
4 }
```
- Status:** 200 OK, Time: 16 ms, Size: 145 B

**Bottom Screenshot:**

- Environment:** No Environment
- URL:** `http://localhost:8080/edd/rest/app/validarUser`
- Method:** POST
- Body Type:** raw (selected), JSON (application/json)
- Body Content:**

```
1 {  
2   "name": "Edd2",  
3   "age": 2019  
4 }
```
- Status:** 200 OK, Time: 290 ms, Size: 142 B