



El futuro digital
es de todos

MinTIC

»
Api rest
xxx



Mision
TIC2022

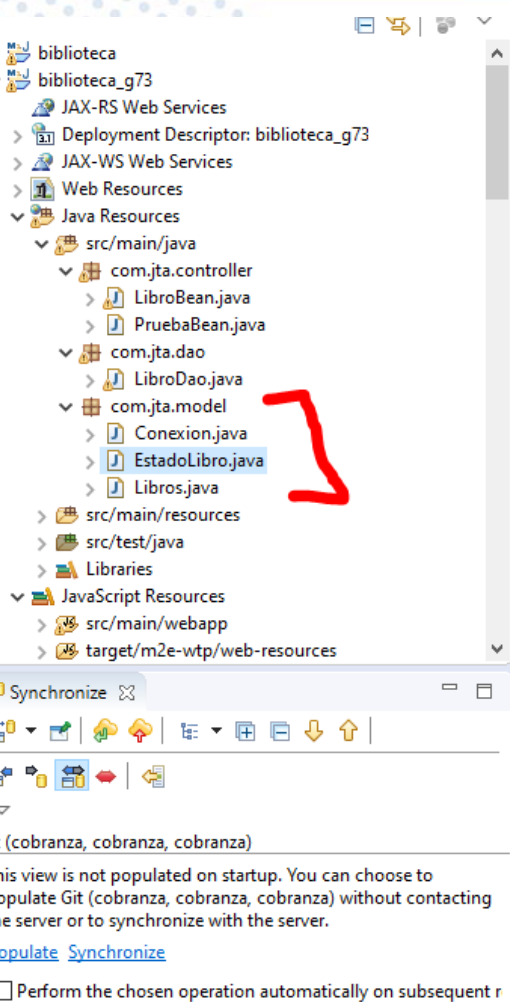


REST es cualquier interfaz entre sistemas que use HTTP para obtener datos o generar operaciones sobre esos datos en todos los formatos posibles, como XML y JSON.

Representational State Transfer- Transferencia de Estado
Representacional



Configuración



```
15 » <servlet-class>javax.faces.webapp.FacesServlet</servlet-class>
16 » <load-on-startup>1</load-on-startup>
17 » </servlet>
18 » <servlet>
19 » <servlet-name>ServiceJT</servlet-name>
20 » <servlet-class>com.sun.jersey.spi.container.servlet.ServletContainer</se
21 » <init-param>
22 » <param-name>com.sun.jersey.config.property.packages</param-name>
23 » <param-value>com.jta.model</param-value>
24 » </init-param>
25 » <init-param>
26 » <param-name>com.sun.jersey.api.json.POJOMappingFeature</param-name>
27 » <param-value>>true</param-value>
28 » </init-param>
29 » <load-on-startup>1</load-on-startup>
30 » </servlet>
31 » <servlet-mapping>
32 » <servlet-name>FacesServlet</servlet-name>
33 » <url-pattern>/faces/*</url-pattern>
34 » </servlet-mapping>
```



El futuro digital
es de todos

MinTIC

Http

<https://javadesde0.com/introduccion-a-http/>

<https://concepto.de/http/>



Http

El **http** (del inglés **HyperText Transfer Protocol** o **Protocolo de Transferencia de Hiper Textos**) es el protocolo de transmisión de información de la World Wide Web, es decir, el código que se establece para que el computador solicitante y el que contiene la información solicitada puedan “hablar” un mismo idioma a la hora de transmitir información por la red.



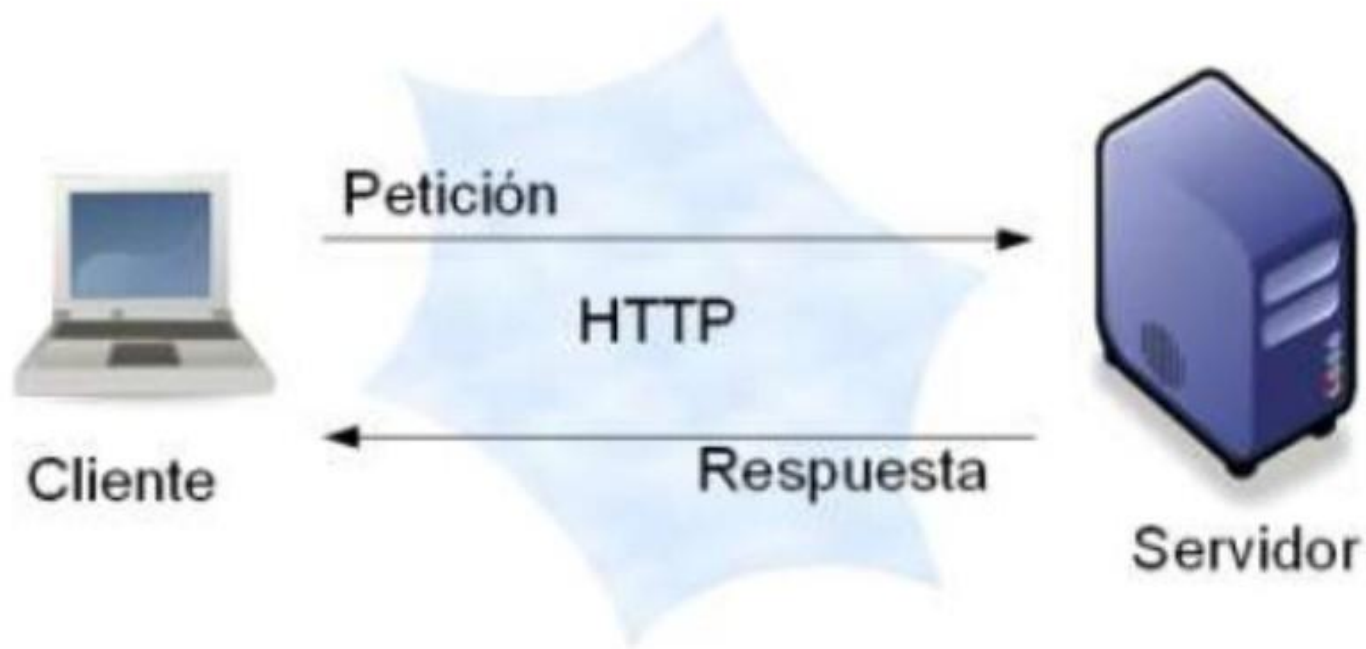
Http

HTTP, consiste en un protocolo que tiene como finalidad la comunicación entre el cliente y servidor. Y que está orientado a transacciones. El cliente, realiza una petición enviando un mensaje, con cierto formato al server. Posteriormente, el server, le envía un mensaje de respuesta.



Http

Arquitectura Cliente - Servidor





Para que sirve el Http

El http, como se ha dicho, es un lenguaje que media entre las peticiones del cliente y las respuestas del servidor en la Internet, para permitir una comunicación fluida y en un mismo “lenguaje”. Este protocolo establece las pautas a seguir, los métodos de petición (llamados “verbos”) y cuenta con cierta flexibilidad para incorporar nuevas peticiones y funcionalidades, en especial a medida que se avanza en sus versiones.



Códigos de estado de HTTP

Una petición vía HTTP, genera una respuesta. Esta respuesta contiene un código que nos permite saber que ha pasado en dicha petición. Cada código tiene un significado concreto.

Los códigos están formados por tres dígitos. El primero empezando por la izquierda (la centena), nos permite identificar el tipo de respuesta que recibimos. Los otros dos (decena y unidad), nos permiten tener un mensaje más conciso, más detallado.



Codigos de estado de HTTP

1XX: Respuestas informativas, indica que la petición ha sido recibida y se está procesando.

Un ejemplo podría ser 102, que significa Processing.

2XX: Respuesta correctas, indica que la petición ha sido procesada satisfactoriamente. Un ejemplo podría ser 200, OK.

3xx: Respuestas de redirección, indica que el cliente necesita realizar más operaciones para finalizar la petición.

Un ejemplo podría ser 301, Moved Permanently.



Códigos de estado de HTTP

4xx: Errores causando por el cliente, indica que ha habido un error en el cliente debido a que este ha hecho algo mal.

Un ejemplo podría ser 404, Not found.

5xx: Errores causados por el servidor, indica que ha habido un error en el procesado a causa de un fallo en el servidor.

Un ejemplo podría ser 504, Gateway Timeout.

<https://www.cheatography.com/kstep/cheat-sheets/http-status-codes/pdf/>



principales métodos de petición

HTTP, tiene una gran variedad de métodos predefinidos. Para identificar a cada uno de ellos se utiliza lo que se conoce como HTTP Verbs (para nosotros a partir de ahora métodos HTTP). El número de métodos va aumentando conforme se desarrollan nuevas versiones del protocolo HTTP.



principales métodos de petición

HTTP, tiene una gran variedad de métodos predefinidos. Para identificar a cada uno de ellos se utiliza lo que se conoce como HTTP Verbs (para nosotros a partir de ahora métodos HTTP). El número de métodos va aumentando conforme se desarrollan nuevas versiones del protocolo HTTP.

Cada método HTTP (HTTP Verb), nos permite identificar la acción que deseamos realizar sobre un recurso determinado.



principales métodos de petición

Las principales operaciones que se utilizan en un sistema HTTP y en los que también utiliza REST son:

GET: Principalmente está enfocado a solicitar, es decir, consultar información existente del servidor.

POST: Principalmente está enfocado a crear un objeto en el servidor.

PUT: Principalmente está enfocado a editar en el servidor.

DELETE: Principalmente está enfocado a eliminar información del servidor.

Aunque existen algunas más, las anteriores son las principales.



El futuro digital
es de todos

MinTIC

Introducción a rest

<https://javadesde0.com/introduccion-a-rest/>



El futuro digital
es de todos

MinTIC

Introducción a rest

Desarrollar nuestra arquitectura de software para crear aplicaciones web basadas en el protocolo HTTP.

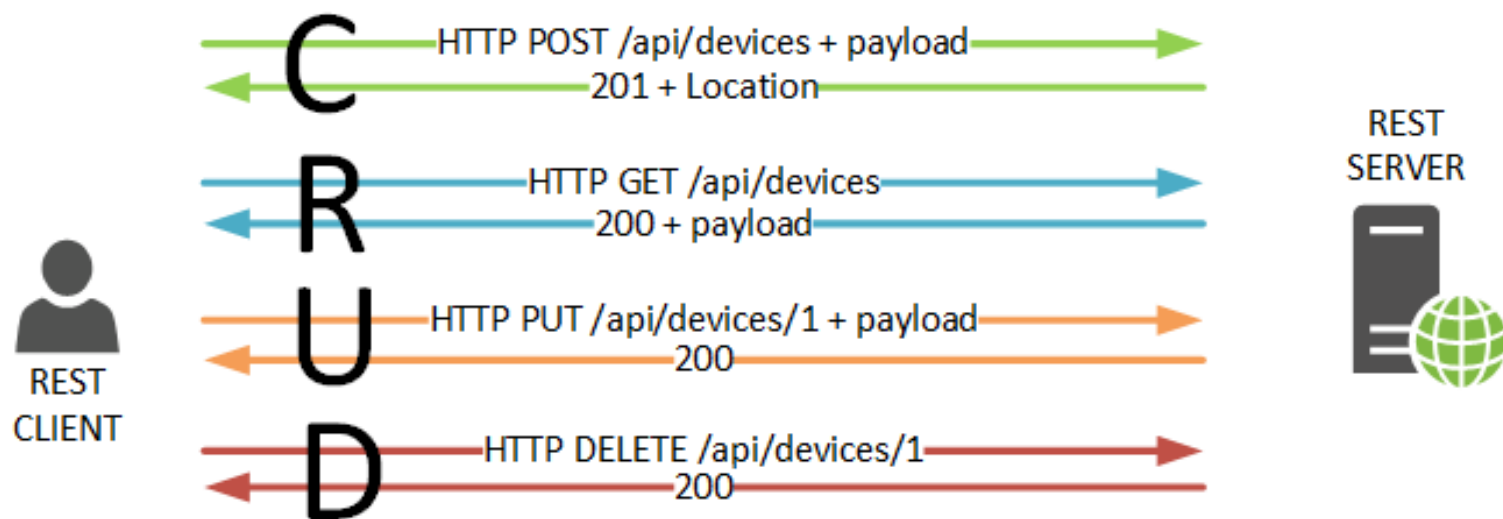


Introducción a rest

Por tanto, API REST, básicamente nos permite realizar peticiones vía HTTP básicas como GET (obtener), POST (acceso), PUT (añadir) y DELETE (eliminar), entre otras. Y usualmente, haremos que la respuesta que nos devuelva dicha petición HTTP sea un archivo JSON o XML, sin plantillas y con el beneficio que posteriormente lo podremos presentar la información en cualquier tecnología independientemente de la tecnología que usemos para realizar la API REST.



Introducción a rest





¿Qué es una API?

API = Application Programming Interface o Interfaz de Programación de Aplicaciones. Una API, es un un software que se comunica con otro software. Y por tanto, se produce una comunicación entre softwares, es decir, sin interacción humana.



Retorno de datos en una API REST

Los principales formatos en los que se suelen devolver los datos de una API REST son XML o JSON. Principalmente JSON, ya que es la evolución de XML y además es más rápido.



Retorno de datos en una API REST

JSON

```
{  
  "siblings": [  
    {"firstName": "Anna", "lastName": "Clayton"},  
    {"firstName": "Alex", "lastName": "Clayton"}  
  ]  
}
```

XML

```
<siblings>  
  <sibling>  
    <firstName>Anna</firstName>  
    <lastName>Clayton</lastName>  
  </sibling>  
  <sibling>  
    <firstName>Alex</firstName>  
    <lastName>Clayton</lastName>  
  </sibling>  
</siblings>
```



Beneficios API REST

Escalabilidad.

Rendimiento.

Seguridad.

Estandarización.

Fácil de usar.

Independencia sobre el lenguaje de implementación ya que las peticiones se responden usualmente mediante a JSON o XML.



El futuro digital
es de todos

MinTIC

Que es REST FULL?

REST FULL, se utiliza para referirse a los servicios web que ejecutan la arquitectura REST.



El futuro digital
es de todos

MinTIC

Instalación SoapUI

<https://www.soapui.org/downloads/soapui/>



SoapUI

Supported by SMARTBEAR

Tools ▾ Learn Docs ▾ Resources



Try ReadyAPI Free



ReadyAPI

Get the most advanced functional testing tool
for REST, SOAP and GraphQL APIs.

Download ReadyAPI

- ✓ Test REST, SOAP, GraphQL, Apache Kafka, gRPC & more
- ✓ Automate process with custom scripting, CI/CD integrations & data generations
- ✓ Definitions: OpenAPI, AsyncAPI, Apache Avro,



SoapUI Open Source

Get the open source version of the most widely
used API testing tool in the world.

Download SoapUI Open Source

- ✓ Test REST & SOAP
- ✓ Basic Property Transfer
- ✓ Definitions: WSDL
- ✓ Basic Functional and performance testing
- ✓ 24-hour customer support

Welcome! Do you need help with anything?



unab
VIGILADA MINEDUCACIÓN

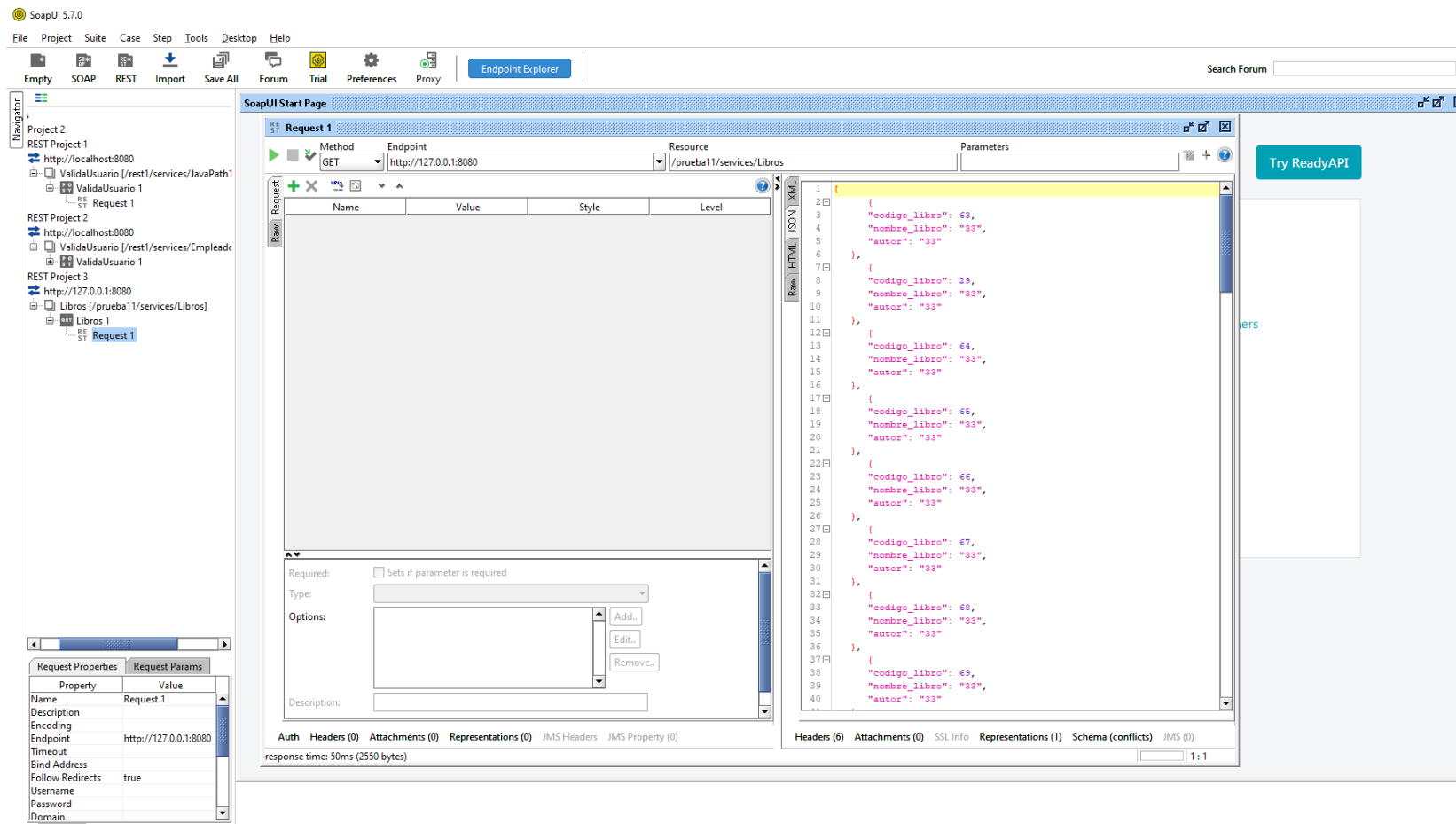
Mision
TIC2022



El futuro digital
es de todos

MinTIC

Instalación SoapUI



VIGILADA MINEDUCACIÓN

Mision
TIC2022

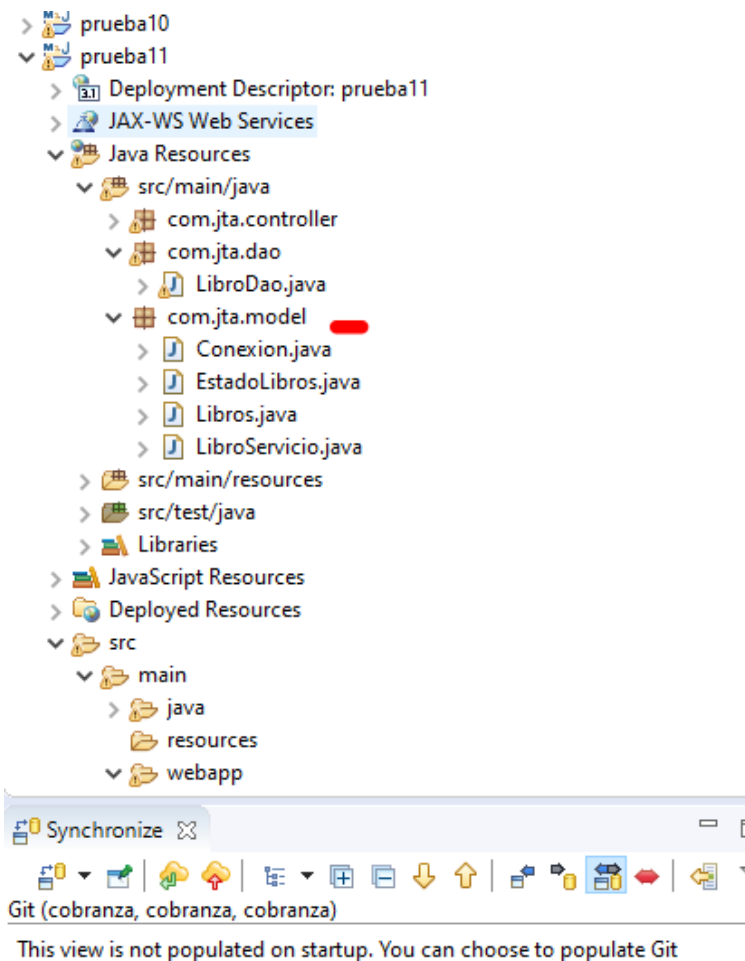


Configuración web.xml

```
</servlet>␣  
<servlet>␣  
»   <servlet-name>ServiceJT</servlet-name>␣  
»   <servlet-class>com.sun.jersey.spi.container.servlet.ServletContainer␣  
»   <init-param>␣  
»       <param-name>com.sun.jersey.config.property.packages</param-name>␣  
»       <param-value>com.jta.model</param-value>␣  
»   </init-param>␣  
»   <init-param>␣  
»       <param-name>com.sun.jersey.api.json.POJOMappingFeature</param-na␣  
»       <param-value>>true</param-value>␣  
»   </init-param>␣  
»   <load-on-startup>1</load-on-startup>␣  
</servlet>␣  
<servlet-mapping>␣  
»   <servlet-name>Faces·Servlet</servlet-name>␣  
»   <url-pattern>/faces/*</url-pattern>␣
```

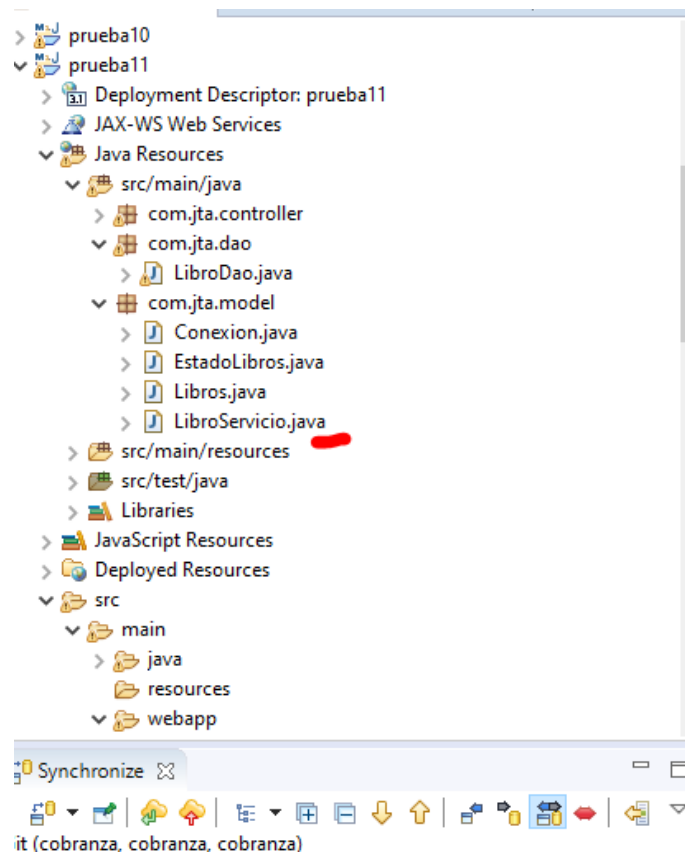


Configuración api rest





Configuración web.xml





Creación del servicio – Return Arraylist

```
    }  
    @Path("/Libros")  
    public class LibroServicio {  
        » @GET  
        » @Produces({ MediaType.APPLICATION_JSON })  
        » public List<Libros> getLibros_JSON() {  
        »     » LibroDao librosdao = new LibroDao();  
        »     » List<Libros> listaLibros = librosdao.getLibros();  
        »     » return listaLibros;  
        » }  
    }  
}
```



Generar la consulta

```
public ArrayList<Libros> getLibros(){  
..... Connection conexion_base_datos=null;  
..... conexion_base_datos=Conexion.getConnection();  
..... ArrayList<Libros> arreglo_libros=new ArrayList();  
.....  
..... PreparedStatement prdStmt=null;  
..... try{  
..... String strSentencia=..... "..... select id, rtrim(titulo), rtrim(au  
..... + "..... from biblioteca.libros....."  
..... + "..... order by titulo";  
.....  
..... prdStmt=conexion_base_datos.prepareStatement(strSentencia);  
.....  
..... ResultSet resultado_consulta=prdStmt.executeQuery();  
..... while(resultado_consulta.next()){  
..... int idTemp=resultado_consulta.getInt(1);  
..... String nombreTemp=resultado_consulta.getString(2);  
..... String autorTemp=resultado_consulta.getString(2);  
..... arreglo_libros.add(new Libros(idTemp, nombreTemp, autorTen  
..... }  
.....  
..... }  
..... }
```



Html

```
.....<br /><br />..xj
.....<h:commandButton id="submitmodcita" styleClass="btn btn-lg btn-outline-
.....onclick="ejercicio1()" value="buscar">xj
.....</h:commandButton>xj
.....xj
...</h:form>xj
```



JavaScript





Configuración de services en el web.xml

```
» </servlet-mapping>␣  
» <servlet-mapping>␣  
»   <servlet-name>ServiceJT</servlet-name>␣  
»   <url-pattern>/services/*</url-pattern>␣  
» </servlet-mapping>␣  
» <session-config>␣  
»   <session-timeout>60</session-timeout>␣  
» </session-config>␣  
» ␣  
» <welcome-file-list>␣
```




Consulta del JSON

← ↻ 🏠 ⓘ localhost:8080/prueba11/services/Libros

🔄 Archer C5 📄 ejemplo base para... 🖼️ localhost:8080/cons... 📄 ::TRANSPORTE BAN... 🖼️ Ingreso Maven JTA 📁 ingles 📁 html5 📁 chicharrones 📁 JAVA 📁 postgres 📁 enviar archivos 📁

```
[{"codigo_libro":63,"nombre_libro":"33","autor":"33"}, {"codigo_libro":29,"nombre_libro":"33","autor":"33"}, {"codigo_libro":64,"nombre_libro":"33","autor":"33"}, {"codigo_libro":67,"nombre_libro":"33","autor":"33"}, {"codigo_libro":68,"nombre_libro":"33","autor":"33"}, {"codigo_libro":69,"nombre_libro":"33","autor":"33"}, {"codigo_libro":39,"nombre_libro":"4","autor":"4"}, {"codigo_libro":38,"nombre_libro":"423423","autor":"423423"}, {"codigo_libro":26,"nombre_libro":"44","autor":"44"}, {"codigo_libro":30,"nombre_libro":"555","autor":"555"}, {"codigo_libro":36,"nombre_libro":"66","autor":"66"}, {"codigo_libro":25,"nombre_libro":"CIEN AÑOS","autor":"CIEN AÑOS"}, {"codigo_libro":23,"nombre_libro":"CIEN AÑOS DE SOLEDAD","autor":"CIEN AÑOS DE SOLEDAD"}, {"codigo_libro":21,"nombre_libro":"d","autor":"d"}, {"codigo_libro":22,"nombre_libro":"d","autor":"d"}, {"codigo_libro":20,"nombre_libro":"dsfdsfdf","autor":"dsfdsfdf"}, {"codigo_libro":24,"nombre_libro":"EL GENERAL","autor":"EL GENERAL"}, {"codigo_libro":15,"nombre_libro":"HAMILI HOJARASCA"}, {"codigo_libro":2,"nombre_libro":"LO QUE EL VIENTO SE LLEVO","autor":"LO QUE EL VIENTO SE LLEVO"}, {"codigo_libro":16,"nombre_libro":"prueba","autor":"prueba"}, {"codigo_libro":85,"nombre_libro":"PRUEBA 150","autor":"PRUEBA 150"}, {"codigo_libro":33,"nombre_libro":"PRUEBA G74 1","autor":"PRUEBA G74 1"}, {"codigo_libro":40,"nombre_libro":"prueba101","autor":"prueba101"}, {"codigo_libro":41,"nombre_libro":"prueba101","autor":"prueba101"}, {"codigo_libro":42,"nombre_libro":"prueba102","autor":"prueba102"}, {"codigo_libro":19,"nombre_libro":"prueba102","autor":"prueba102"}, {"codigo_libro":4,"nombre_libro":"romeo y julieta","autor":"romeo y julieta"}, {"codigo_libro":1,"nombre_libro":"ROMEO Y JULIETA","autor":"ROMEO Y JULIETA"}, {"codigo_libro":34,"nombre_libro":"w","autor":"w"}, {"codigo_libro":28,"nombre_libro":"w","autor":"w"}]
```