

UNIVERSIDAD CENTRAL DEL ECUADOR
Facultad de Ingeniería, Ciencias Físicas y Matemática
Ingeniería Informática

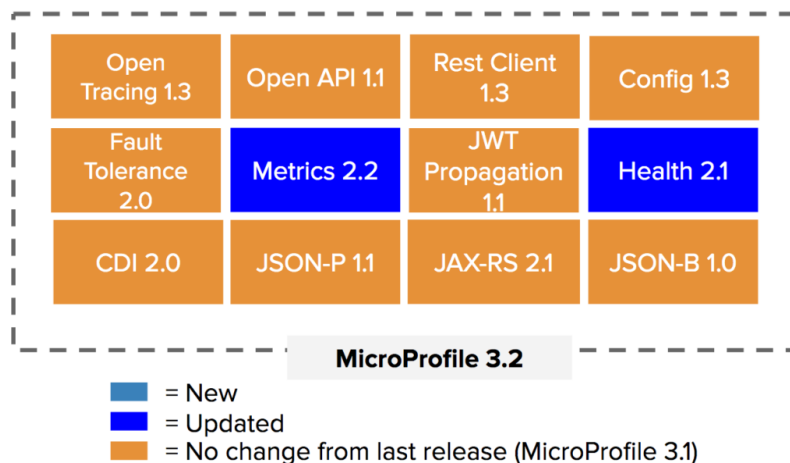
NOMBRE: Luis Pantoja

MATERIA: Programación Distribuida

FECHA: 2020-01-15

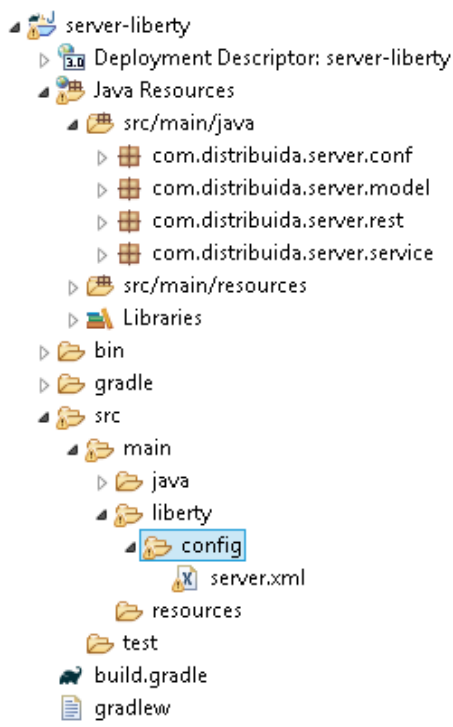
MICROPROFILE

Eclipse MicroProfile es una colección de librerías Java EE y tecnologías que juntas forman la línea base para microservicios que apunta a entregar aplicaciones portables a través de múltiples sistemas operativos.



OpenLiberty

Open Liberty es un entorno de ejecución Java ligero de código abierto construido mediante características modulares. Con MicroProfile, un proyecto de código abierto que define nuevos estándares y API para acelerar y simplificar la creación de microservicios.



Los archivos más importantes son el build.gradle y el server.xml donde están las dos configuraciones del proyecto. Este es el build.gradle

```
apply plugin: 'eclipse'
apply plugin: 'war'
apply plugin: 'liberty'

buildscript {
    repositories {
        mavenCentral()
    }
    dependencies {
        classpath 'net.wasdev.wlp.gradle.plugins:liberty-gradle-plugin:2.6.3'
    }
}

repositories {
    mavenCentral()
}

dependencies {
    providedCompile group: 'org.eclipse.microprofile', name: 'microprofile', version: '3.2'
    //compile group: 'javax', name: 'javaee-api', version: '8.0.1'
    libertyRuntime group: 'io.openliberty', name: 'openliberty-runtime', version: '19.0.0.12'
    //BD
    compile group: 'commons-dbcp', name: 'commons-dbcp', version: '1.2.2'
    compile group: 'org.postgresql', name: 'postgresql', version: '42.2.8'
}
```

Aquí podemos ver que necesitamos el plugin de Liberty, además de 'net.wasdev.wlp.gradle.plugins:liberty-gradle-plugin:2.6.3' en las dependencias usamos la dependencia de microprofile y Liberty, luego definimos los puertos de Liberty y donde se encuentra el archivo de configuración server.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<server description="server-liberty">

    <featureManager>
        <feature>microProfile-3.2</feature>
    </featureManager>

    <httpEndpoint httpPort="9080" httpsPort="9443" id="defaultHttpEndpoint" host="*" />
</server>
```

Por último, ejecutamos los siguientes comandos para que funcione la aplicación:

Primero con cmd, nos ubicamos en la carpeta raíz del proyecto:

```
cd C:\Users\Luis\Documents\workspace-spring-tool-suite-4-4.2.1.RELEASE\server-liberty
```

En esta carpeta donde este el archivo gradlew ejecutamos los comandos:

gradlew build

```
C:\Users\Luis\Documents\workspace-spring-tool-suite-4-4.2.1.RELEASE\server-liberty>gradlew build
Starting a Gradle Daemon, 2 incompatible and 1 stopped Daemons could not be reused, use --status for details

BUILD SUCCESSFUL in 24s
2 actionable tasks: 2 up-to-date
```

gradlew clean

```
C:\Users\Luis\Documents\workspace-spring-tool-suite-4-4.2.1.RELEASE\server-liberty>gradlew clean
BUILD SUCCESSFUL in 3s
2 actionable tasks: 2 executed
```

gradlew eclipse

```
C:\Users\Luis\Documents\workspace-spring-tool-suite-4-4.2.1.RELEASE\server-liberty>gradlew eclipse
BUILD SUCCESSFUL in 2s
7 actionable tasks: 5 executed, 2 up-to-date
```

gradlew libertyStart

```
C:\Users\Luis\Documents\workspace-spring-tool-suite-4-4.2.1.RELEASE\server-liberty>gradlew libertyStart

> Task :installApps
At least one application is not defined in the server configuration but the build file indicates it should be installed in the apps folder. Liberty will use additional application configuration added to the the target server configuration dropins folder by the plug-in.

BUILD SUCCESSFUL in 58s
6 actionable tasks: 2 executed, 4 up-to-date
```

gradlew libertyStop

```
C:\Users\Luis\Documents\workspace-spring-tool-suite-4-4.2.1.RELEASE\server-liberty>gradlew libertyStop

BUILD SUCCESSFUL in 5s
1 actionable task: 1 executed
```

Y, por último, en el navegador escribimos la dirección <http://localhost:9080/server-liberty/singers>

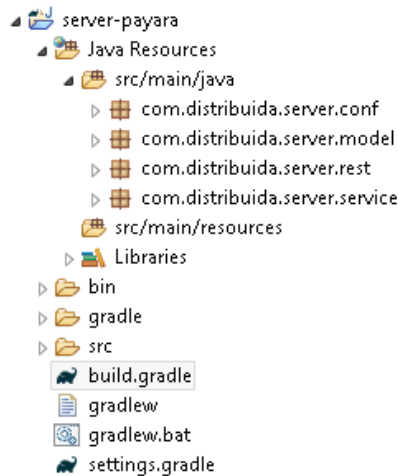


The screenshot shows a web browser window with the address bar displaying `localhost:9080/server-liberty/singers`. The browser's address bar includes navigation icons (back, forward, refresh, home) and a search icon. Below the address bar is a toolbar with various icons for different services like Wolfram, YouTube, Facebook, Mail UCE, Yahoo, and others. The main content area of the browser displays a JSON array of objects, each representing a singer with fields for firstName, id, and lastName. The JSON data is as follows:

```
[{"firstName":"A","id":2,"lastName":"A"}, {"firstName":"A","id":3,"lastName":"AB"}, {"firstName":"ab","id":4,"lastName":"ab1"}, {"firstName":"Luis","id":5,"lastName":"Pantoja"}, {"firstName":"NarIn","id":7,"lastName":"Abb"}, {"firstName":"U","id":8,"lastName":"U"}, {"firstName":"NarIn","id":10,"lastName":"ADAA"}, {"firstName":"marIn","id":11,"lastName":"Ada"}]
```

Payara

Payara Server es una plataforma de middleware nativa en la nube y de código abierto (reemplaza a GlassFish Server Open Source) que admite implementaciones confiables y seguras de aplicaciones Java EE (Jakarta EE) en las instalaciones, en la nube o en entornos híbridos.



El archivo build.gradle es el siguiente:

```
apply plugin: 'eclipse'
apply plugin: 'war'

repositories {
    jcenter()
    mavenCentral()
}

dependencies {
    providedCompile group: 'org.eclipse.microprofile', name: 'microprofile', version: '3.2'
    //compile group: 'javax', name: 'javaee-api', version: '8.0.1'
    //BD
    compile group: 'commons-dbcp', name: 'commons-dbcp', version: '1.2.2'
    compile group: 'org.postgresql', name: 'postgresql', version: '42.2.8'
}
```

Para ejecutar la aplicación escribimos los siguientes comandos:

Descargar payara del siguiente enlace <https://www.payara.fish/software/downloads/> en la versión micro

Payara Micro 194

This microservices-ready version of Payara Server is perfect for cloud environments. Compatible with Eclipse MicroProfile 3.2, small (<80MB) and incredibly simple to use, it enables you to run war files from the command line without any application server installation.

[See the Data Sheet](#)

Payara Micro 194 Download

```
cd C:\Users\Luis\Documents\workspace-spring-tool-suite-4-4.2.1.RELEASE\server-payara
```

gradlew build

```
C:\Users\Luis\Documents\workspace-spring-tool-suite-4-4.2.1.RELEASE\server-payara>gradlew build
BUILD SUCCESSFUL in 2s
2 actionable tasks: 2 up-to-date
```

cd C:\Users\Luis\Downloads

```
java -jar payara-micro-5.194.jar --deploy
C:\Users\Luis\Documents\workspace-spring-tool-suite-4-4.2.1.RELEASE\server-payara\build\libs\server-payara.war
```

```
C:\Users\Luis\Downloads>java -jar payara-micro-5.194.jar --deploy C:\Users\Luis\Documents\workspace-spring-tool-suite-4-4.2.1.RELEASE\server-payara\build\libs\server-payara.war
```

```
[2020-01-14T13:42:22.775-0500] [] [1;92mINFORMACIEN[0m] [] [1;94mPayaraMicro[0m] [tid: _ThreadID=1 _ThreadName=main] [timeMillis: 1579027342775] [levelValue: 800]
Payara Micro URLs:
http://PANTOJA-LAP:8080/server-payara

'server-payara' REST Endpoints:
GET    /server-payara/application.wadl
GET    /server-payara/singers
GET    /server-payara/singers/{id}

[2020-01-14T13:42:22.776-0500] [] [1;92mINFORMACIEN[0m] [] [1;94mPayaraMicro[0m] [tid: _ThreadID=1 _ThreadName=main] [timeMillis: 1579027342776] [levelValue: 800] Payara Micro 5.194 #badassmicrofish (build 327) ready in 35.651 (ms)
```

Por último, en el navegador escribimos <http://localhost:8080/server-payara/singers>

```
localhost:8080/server-payara/singers
Wolfram|Otras YouTube RHCP Facebook Mail UCE Yahoo Series Tutoriales Programacion Matematicas Juegos Informatica gifs Otros
[{"firstName":"A","id":2,"lastName":"A"}, {"firstName":"A","id":3,"lastName":"AB"}, {"firstName":"ab","id":4,"lastName":"ab1"}, {"firstName":"Luis","id":5,"lastName":"Pantoja"}, {"firstName":"NarIn","id":7,"lastName":"Abb"}, {"firstName":"U","id":8,"lastName":"U"}, {"firstName":"NarIn","id":10,"lastName":"ADAA"}, {"firstName":"marin","id":11,"lastName":"Ada"}]
```