



Universidad Central del Ecuador

Programación Distribuida

Título: Informe de despliegue del
aplicativo completo



Integrantes Grupo 8: Alexander Columba
Alexis Masache
Santiago Vega



Contenido

1. Objetivo	3
2. Objetivos Específicos	3
3. Marco Teórico Herramientas Tecnológicas.....	¡Error! Marcador no definido.
3.1. Apache TomEE	¡Error! Marcador no definido.
3.2. JPA e Hibernate	¡Error! Marcador no definido.
3.3. Servidor de Registro y Configuración Zookeeper:..	¡Error! Marcador no definido.
3.4. Spring Cloud Gateway y Spring Cloud Load Balancer	¡Error! Marcador no definido.
4. Entorno de Desarrollo	¡Error! Marcador no definido.
5. Implementación de las Aplicaciones:	¡Error! Marcador no definido.
5.1. Servidor 1: Customer	¡Error! Marcador no definido.
5.2. Servidor 2: Orders	¡Error! Marcador no definido.
5.3. Creación del Cliente Web	¡Error! Marcador no definido.
6. Servidor de Registro y Configuración Apache Zookeeper	¡Error! Marcador no definido.
Instalación del Servidor Zookeeper	¡Error! Marcador no definido.
Servidor Spring Cloud Gateway	¡Error! Marcador no definido.
7. Conclusiones:	¡Error! Marcador no definido.
8. Referencias:	¡Error! Marcador no definido.



1. Objetivo

- Dar una guía detallada para desplegar el Aplicativo del Proyecto completo con todas las tecnologías y frameworks descritos en el documento de implementación.

2. Objetivos Específicos

- Configurar y desplegar el servidor de Configuración y Registro Zookeeper.
- Crear el Uber jar y desplegar 3 instancias del servidor 1 Customers .
- Crear el Uber jar y desplegar 2 instancias del servidor 2 Orders .
- Desplegar el cliente web.
- Desplegar el servidor Spring Cloud Gateway.
- Realizar pruebas funcionales de todo el Aplicativo

3. Despliegue de las Aplicaciones

3.1. Instalación de Maven.

En nuestro caso utilizamos Maven para la construcción de los servidores, y también instalamos Maven localmente:

- Maven puede ser descargado de aquí: <https://maven.apache.org/download.cgi>, como un Binary zip archive.
- Este archivo solo debe ser descomprimido en una ruta de nuestro gusto, en nuestro caso escogimos la ruta C:/
- La ruta de la carpeta que acabamos de descomprimir debe ser agregada en la variable de sistema que se denomina *Path*
- Con todo esto realizado ya podemos ejecutar el comando *MVN* en alguna ventana de CMD (Símbolo del Sistema (CMD) que servirá, específicamente en este caso, para construir el Uber Jar de ambos servidores.

3.2. Proceso de despliegue de 2 instancias del servidor Customer.

- 1- Debemos dirigirnos a la carpeta del proyecto en nuestro caso será en el directorio de trabajo de STS, ...`\workspace-spring-tool-suite-4-4.7.0.RELEASE\servidor-customer-tomee`



Nombre	Fecha de modificación	Tipo	Tamaño
.distribution	12/09/2020 20:30	Carpeta de archivos	
.settings	16/09/2020 13:41	Carpeta de archivos	
src	11/09/2020 13:38	Carpeta de archivos	
target	16/09/2020 18:48	Carpeta de archivos	
.classpath	16/09/2020 18:39	Archivo CLASSPATH	3 KB
.project	16/09/2020 18:37	Archivo PROJECT	2 KB
pom.xml	16/09/2020 18:36	Documento XML	3 KB
readme.md	11/09/2020 18:37	Archivo MD	1 KB

- 2- Sobre esta ruta vamos a ejecutar el siguiente comando en una ventana de CMD para construir el Uber Jar de la aplicación correspondiente al servicio Customer-Server: *mvn clean package*

```
workspace-spring-tool-suite-4-4.7.0.RELEASE\servidor-customer-tomee>mvn clean package
```

- 3- Después de esto en el directorio *target* se creará un archivo con el nombre: *servidor-customer-tomee-exec.jar*

Nombre	Fecha de modificación	Tipo	Tamaño
.distribution	16/09/2020 18:48	Carpeta de archivos	
apache-tomee	16/09/2020 18:47	Carpeta de archivos	
classes	16/09/2020 18:47	Carpeta de archivos	
generated-sources	16/09/2020 18:47	Carpeta de archivos	
maven-archiver	16/09/2020 18:47	Carpeta de archivos	
maven-status	16/09/2020 18:47	Carpeta de archivos	
servidor-customer-tomee	16/09/2020 18:47	Carpeta de archivos	
servidor-customer-tomee.war	16/09/2020 18:47	Archivo WAR	24.821 KB
servidor-customer-tomee.zip	16/09/2020 18:48	Archivo WinRAR Z...	69.201 KB
servidor-customer-tomee-exec.jar	16/09/2020 18:48	Executable Jar File	69.258 KB

- 4- Ahora vamos a mandar un comando para ejecutar el archivo *.jar*, además de especificar los puertos que necesita la aplicación para “correr”, necesitamos especificar dos puertos, uno para la comunicación http, que será por donde accederemos a los distintos servicios, y el otro puerto funciona para apagar la aplicación; el comando es el siguiente: *java -DadditionalSystemProperties="-Dhttp.port=8083 -Dshutdown.port=8009" -jar "target\servidor-customer-tomee-exec.jar"*



```
C:\Users\Famil\Documents\workspace-spring-tool-suite-4-4.7.0.RELEASE\servidor-customer-tomee\target>java -DadditionalSystemProperties="-Dhttp.port=8083 -Dshutdown.port=8089" -jar servidor-customer-tomee-exec.jar
Extracting tomee to C:\Users\Famil\Documents\workspace-spring-tool-suite-4-4.7.0.RELEASE\servidor-customer-tomee\target\distribution
Using CATALINA_BASE: "C:\Users\Famil\Documents\workspace-spring-tool-suite-4-4.7.0.RELEASE\servidor-customer-tomee\target\distribution\apache-tomee"
Using CATALINA_HOME: "C:\Users\Famil\Documents\workspace-spring-tool-suite-4-4.7.0.RELEASE\servidor-customer-tomee\target\distribution\apache-tomee"
Using CATALINA_TMPDIR: "C:\Users\Famil\Documents\workspace-spring-tool-suite-4-4.7.0.RELEASE\servidor-customer-tomee\target\distribution\apache-tomee\temp"
Using JRE_HOME: "C:\Program Files\Java\jdk-11.0.8"
Using CLASSPATH: "C:\Users\Famil\Documents\workspace-spring-tool-suite-4-4.7.0.RELEASE\servidor-customer-tomee\target\distribution\apache-tomee\bin\bootstrap.jar;C:\Users\Famil\Documents\workspace-spring-tool-suite-4-4.7.0.RELEASE\servidor-customer-tomee\target\distribution\apache-tomee\bin\tomcat-juli.jar"
NOTE: Picked up JDK_JAVA_OPTIONS: --add-opens=java.base/java.lang=ALL-UNNAMED --add-opens=java.base/java.io=ALL-UNNAMED --add-opens=java.rmi/sun.rmi.transport=ALL-UNNAMED
WARNING: An illegal reflective access operation has occurred
WARNING: Illegal reflective access by org.apache.tomee.catalina.ServerListener (file:C:\Users\Famil\Documents\workspace-spring-tool-suite-4-4.7.0.RELEASE\servidor-customer-tomee\target\distribution\apache-tomee\lib\tomee-catalina-8.0.0-M3.jar) to field java.lang.reflect.Field.modifiers
WARNING: Please consider reporting this to the maintainers of org.apache.tomee.catalina.ServerListener
WARNING: Use --illegal-access=warn to enable warnings of further illegal reflective access operations
WARNING: All illegal access operations will be denied in a future release
16-Sep-2020 18:48:42.635 INFO [main] jdk.internal.reflect.NativeMethodAccessorImpl.invoke Inicializando el manejador de protocolo ["http-nio-8083"]
```

- 5- La primera instancia en este caso se levantará en el puerto 8083, lo verificaremos posteriormente en el servidor de registro.
- 6- Levantaremos otra instancia en el puerto 8084.

```
C:\Users\Famil\Documents\workspace-spring-tool-suite-4-4.7.0.RELEASE\servidor-customer-tomee>java -DadditionalSystemProperties="-Dhttp.port=8084 -Dshutdown.port=8010" -jar "target\servidor-customer-tomee-exec.jar"
Deleting C:\Users\Famil\Documents\workspace-spring-tool-suite-4-4.7.0.RELEASE\servidor-customer-tomee\distribution
Extracting tomee to C:\Users\Famil\Documents\workspace-spring-tool-suite-4-4.7.0.RELEASE\servidor-customer-tomee\distribution
Using CATALINA_BASE: "C:\Users\Famil\Documents\workspace-spring-tool-suite-4-4.7.0.RELEASE\servidor-customer-tomee\distribution\apache-tomee"
Using CATALINA_HOME: "C:\Users\Famil\Documents\workspace-spring-tool-suite-4-4.7.0.RELEASE\servidor-customer-tomee\distribution\apache-tomee"
Using CATALINA_TMPDIR: "C:\Users\Famil\Documents\workspace-spring-tool-suite-4-4.7.0.RELEASE\servidor-customer-tomee\distribution\apache-tomee\temp"
Using JRE_HOME: "C:\Program Files\Java\jdk-11.0.8"
Using CLASSPATH: "C:\Users\Famil\Documents\workspace-spring-tool-suite-4-4.7.0.RELEASE\servidor-customer-tomee\distribution\apache-tomee\bin\bootstrap.jar;C:\Users\Famil\Documents\workspace-spring-tool-suite-4-4.7.0.RELEASE\servidor-customer-tomee\distribution\apache-tomee\bin\tomcat-juli.jar"
NOTE: Picked up JDK_JAVA_OPTIONS: --add-opens=java.base/java.lang=ALL-UNNAMED --add-opens=java.base/java.io=ALL-UNNAMED --add-opens=java.rmi/sun.rmi.transport=ALL-UNNAMED
WARNING: An illegal reflective access operation has occurred
WARNING: Illegal reflective access by org.apache.tomee.catalina.ServerListener (file:C:\Users\Famil\Documents\workspace-spring-tool-suite-4-4.7.0.RELEASE\servidor-customer-tomee\distribution\apache-tomee\lib\tomee-catalina-8.0.0-M3.jar) to field java.lang.reflect.Field.modifiers
WARNING: Please consider reporting this to the maintainers of org.apache.tomee.catalina.ServerListener
WARNING: Use --illegal-access=warn to enable warnings of further illegal reflective access operations
WARNING: All illegal access operations will be denied in a future release
16-Sep-2020 21:21:06.003 INFO [main] jdk.internal.reflect.NativeMethodAccessorImpl.invoke Inicializando el manejador de protocolo ["http-nio-8084"]
```

- 7- En este punto mostraré las instancias registradas en el servidor Zookeeper y posteriormente indicaremos qué comandos pueden servir para observar estos registros.

```
[zk: localhost:2181(CONNECTED) 23] ls /services/customer-server
[customer-server-8083, customer-server-8084]
[zk: localhost:2181(CONNECTED) 24]
```

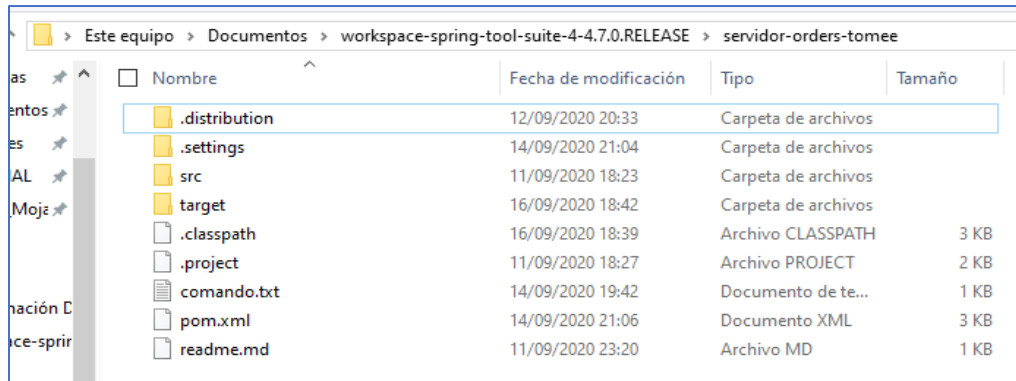
- 8- Podemos ir a un navegador y probar cualquier instancia del microservicio:

```
localhost:8083/customers
[{"id":1,"name":"Alexis","surname":"Masache"}, {"id":2,"name":"Fernando","surname":"Bonilla"}, {"id":3,"name":"Aldo","surname":"Palacios"}]
```

```
localhost:8084/customers
[{"id":1,"name":"Alexis","surname":"Masache"}, {"id":2,"name":"Fernando","surname":"Bonilla"}, {"id":3,"name":"Aldo","surname":"Palacios"}]
```

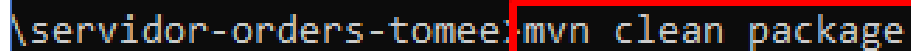
3.3. Proceso de despliegue de 3 instancias del servidor Orders.

- 1- Debemos dirigirnos a la carpeta del proyecto en nuestro caso será en el directorio de trabajo de STS, ...*workspace-spring-tool-suite-4-4.7.0.RELEASE\servidor-orders-tomee*



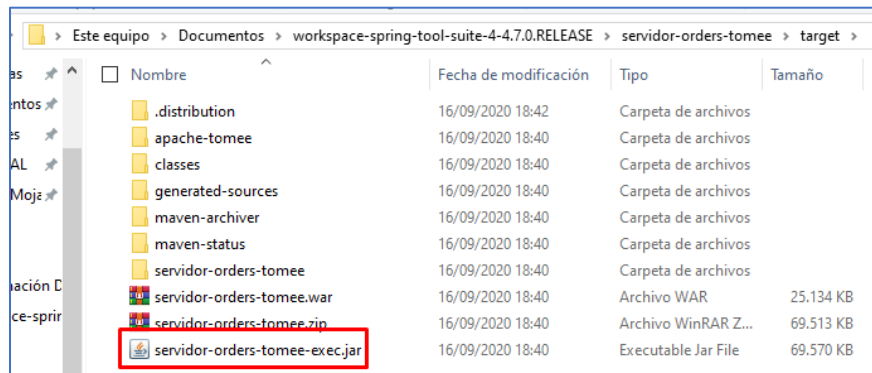
Nombre	Fecha de modificación	Tipo	Tamaño
.distribution	12/09/2020 20:33	Carpeta de archivos	
.settings	14/09/2020 21:04	Carpeta de archivos	
src	11/09/2020 18:23	Carpeta de archivos	
target	16/09/2020 18:42	Carpeta de archivos	
.classpath	16/09/2020 18:39	Archivo CLASSPATH	3 KB
.project	11/09/2020 18:27	Archivo PROJECT	2 KB
comando.txt	14/09/2020 19:42	Documento de te...	1 KB
pom.xml	14/09/2020 21:06	Documento XML	3 KB
readme.md	11/09/2020 23:20	Archivo MD	1 KB

- 2- Sobre esta ruta vamos a ejecutar el siguiente comando en una ventana de CMD para construir el Uber Jar de la aplicación correspondiente al servicio Orders-Server:
mvn clean package



```
\servidor-orders-tomee:mvn clean package
```

- 3- Después de esto en el directorio *target* se creará un archivo con el nombre: *servidor-orders-tomee-exec.jar*



Nombre	Fecha de modificación	Tipo	Tamaño
.distribution	16/09/2020 18:42	Carpeta de archivos	
apache-tomee	16/09/2020 18:40	Carpeta de archivos	
classes	16/09/2020 18:40	Carpeta de archivos	
generated-sources	16/09/2020 18:40	Carpeta de archivos	
maven-archiver	16/09/2020 18:40	Carpeta de archivos	
maven-status	16/09/2020 18:40	Carpeta de archivos	
servidor-orders-tomee	16/09/2020 18:40	Carpeta de archivos	
servidor-orders-tomee.war	16/09/2020 18:40	Archivo WAR	25.134 KB
servidor-orders-tomee.zip	16/09/2020 18:40	Archivo WinRAR Z...	69.513 KB
servidor-orders-tomee-exec.jar	16/09/2020 18:40	Executable Jar File	69.570 KB

- 4- En este punto vamos a necesitar un comando similar al que utilizamos para desplegar las instancias del servicio Customer-Server, pero ahora referido al servicio Orders-Server: *java -DadditionalSystemProperties="-Dhttp.port=8085 -Dshutdown.port=8011" -jar "target\servidor-orders-tomee-exec.jar"*



```
C:\Windows\System32\cmd.exe - java -DAdditionalSystemProperties="-Dhttp.port=8085 -Dshutdown.port=8011" -jar "target\servidor-orders-tomee-exec.jar"
C:\Users\Famil\Documents\workspace-spring-tool-suite-4-4.7.0.RELEASE\servidor-orders-tomee>java -DAdditionalSystemProperties="-Dhttp.port=8085 -Dshutdown.port=8011" -j
ar "target\servidor-orders-tomee-exec.jar"
Deleting C:\Users\Famil\Documents\workspace-spring-tool-suite-4-4.7.0.RELEASE\servidor-orders-tomee\distribution
Extracting tomee to C:\Users\Famil\Documents\workspace-spring-tool-suite-4-4.7.0.RELEASE\servidor-orders-tomee\distribution
Using CATALINA_BASE: "C:\Users\Famil\Documents\workspace-spring-tool-suite-4-4.7.0.RELEASE\servidor-orders-tomee\distribution\apache-tomee"
Using CATALINA_HOME: "C:\Users\Famil\Documents\workspace-spring-tool-suite-4-4.7.0.RELEASE\servidor-orders-tomee\distribution\apache-tomee"
Using CATALINA_TMPDIR: "C:\Users\Famil\Documents\workspace-spring-tool-suite-4-4.7.0.RELEASE\servidor-orders-tomee\distribution\apache-tomee\temp"
Using JRE_HOME: "C:\Program Files\Java\jdk-11.0.8"
Using CLASSPATH: "C:\Users\Famil\Documents\workspace-spring-tool-suite-4-4.7.0.RELEASE\servidor-orders-tomee\distribution\apache-tomee\bin\bootstrap.jar;C:\User
s\Famil\Documents\workspace-spring-tool-suite-4-4.7.0.RELEASE\servidor-orders-tomee\distribution\apache-tomee\bin\tomcat-juli.jar"
NOTE: Picked up JDK_JAVA_OPTIONS: --add-opens=java.base/java.lang=ALL-UNNAMED --add-opens=java.base/java.io=ALL-UNNAMED --add-opens=java.rmi/sun.rmi.transport=ALL-UNN
AMED
WARNING: An illegal reflective access operation has occurred
WARNING: Illegal reflective access by org.apache.tomee.catalina.ServerListener (file:C:\Users\Famil\Documents\workspace-spring-tool-suite-4-4.7.0.RELEASE\servidor-ord
ers-tomee\distribution\apache-tomee\lib\tomee-catalina-8.0.0-M3.jar) to field java.lang.reflect.Field.modifiers
WARNING: Please consider reporting this to the maintainers of org.apache.tomee.catalina.ServerListener
WARNING: Use --illegal-access=warn to enable warnings of further illegal reflective access operations
WARNING: All illegal access operations will be denied in a future release
16-Sep-2020 21:37:27.131 INFO [main] jdk.internal.reflect.NativeMethodAccessorImpl.invoke Inicializando el manejador de protocolo ["http-nio-8085"]
```

- 5- La primera instancia en este caso se levantará en el puerto 8085, lo verificaremos posteriormente en el servidor de registro.
- 6- Levantaremos otra instancia en el puerto 8086.

```
C:\Windows\System32\cmd.exe - java -DAdditionalSystemProperties="-Dhttp.port=8086 -Dshutdown.port=8012" -jar "target\servidor-orders-tomee-exec.jar"
Microsoft Windows [Versión 10.0.18362.720]
(c) 2019 Microsoft Corporation. Todos los derechos reservados.
C:\Users\Famil\Documents\workspace-spring-tool-suite-4-4.7.0.RELEASE\servidor-orders-tomee>java -DAdditionalSystemProperties="-Dhttp.port=8086 -Dshutdown.port=8012" -j
ar "target\servidor-orders-tomee-exec.jar"
Using CATALINA_BASE: "C:\Users\Famil\Documents\workspace-spring-tool-suite-4-4.7.0.RELEASE\servidor-orders-tomee\distribution\apache-tomee"
Using CATALINA_HOME: "C:\Users\Famil\Documents\workspace-spring-tool-suite-4-4.7.0.RELEASE\servidor-orders-tomee\distribution\apache-tomee"
Using CATALINA_TMPDIR: "C:\Users\Famil\Documents\workspace-spring-tool-suite-4-4.7.0.RELEASE\servidor-orders-tomee\distribution\apache-tomee\temp"
Using JRE_HOME: "C:\Program Files\Java\jdk-11.0.8"
Using CLASSPATH: "C:\Users\Famil\Documents\workspace-spring-tool-suite-4-4.7.0.RELEASE\servidor-orders-tomee\distribution\apache-tomee\bin\bootstrap.jar;C:\User
s\Famil\Documents\workspace-spring-tool-suite-4-4.7.0.RELEASE\servidor-orders-tomee\distribution\apache-tomee\bin\tomcat-juli.jar"
NOTE: Picked up JDK_JAVA_OPTIONS: --add-opens=java.base/java.lang=ALL-UNNAMED --add-opens=java.base/java.io=ALL-UNNAMED --add-opens=java.rmi/sun.rmi.transport=ALL-UNN
AMED
WARNING: An illegal reflective access operation has occurred
WARNING: Illegal reflective access by org.apache.tomee.catalina.ServerListener (file:C:\Users\Famil\Documents\workspace-spring-tool-suite-4-4.7.0.RELEASE\servidor-ord
ers-tomee\distribution\apache-tomee\lib\tomee-catalina-8.0.0-M3.jar) to field java.lang.reflect.Field.modifiers
WARNING: Please consider reporting this to the maintainers of org.apache.tomee.catalina.ServerListener
WARNING: Use --illegal-access=warn to enable warnings of further illegal reflective access operations
WARNING: All illegal access operations will be denied in a future release
16-Sep-2020 21:39:21.048 INFO [main] jdk.internal.reflect.NativeMethodAccessorImpl.invoke Inicializando el manejador de protocolo ["http-nio-8086"]
```

- 7- Levantaremos otra instancia en el puerto 8087.

```
C:\Windows\System32\cmd.exe - java -DAdditionalSystemProperties="-Dhttp.port=8087 -Dshutdown.port=8013" -jar "target\servidor-orders-tomee-exec.jar"
Microsoft Windows [Versión 10.0.18362.720]
(c) 2019 Microsoft Corporation. Todos los derechos reservados.
C:\Users\Famil\Documents\workspace-spring-tool-suite-4-4.7.0.RELEASE\servidor-orders-tomee>java -DAdditionalSystemProperties="-Dhttp.port=8087 -Dshutdown.port=8013" -j
ar "target\servidor-orders-tomee-exec.jar"
Using CATALINA_BASE: "C:\Users\Famil\Documents\workspace-spring-tool-suite-4-4.7.0.RELEASE\servidor-orders-tomee\distribution\apache-tomee"
Using CATALINA_HOME: "C:\Users\Famil\Documents\workspace-spring-tool-suite-4-4.7.0.RELEASE\servidor-orders-tomee\distribution\apache-tomee"
Using CATALINA_TMPDIR: "C:\Users\Famil\Documents\workspace-spring-tool-suite-4-4.7.0.RELEASE\servidor-orders-tomee\distribution\apache-tomee\temp"
Using JRE_HOME: "C:\Program Files\Java\jdk-11.0.8"
Using CLASSPATH: "C:\Users\Famil\Documents\workspace-spring-tool-suite-4-4.7.0.RELEASE\servidor-orders-tomee\distribution\apache-tomee\bin\bootstrap.jar;C:\User
s\Famil\Documents\workspace-spring-tool-suite-4-4.7.0.RELEASE\servidor-orders-tomee\distribution\apache-tomee\bin\tomcat-juli.jar"
NOTE: Picked up JDK_JAVA_OPTIONS: --add-opens=java.base/java.lang=ALL-UNNAMED --add-opens=java.base/java.io=ALL-UNNAMED --add-opens=java.rmi/sun.rmi.transport=ALL-UNN
AMED
WARNING: An illegal reflective access operation has occurred
WARNING: Illegal reflective access by org.apache.tomee.catalina.ServerListener (file:C:\Users\Famil\Documents\workspace-spring-tool-suite-4-4.7.0.RELEASE\servidor-ord
ers-tomee\distribution\apache-tomee\lib\tomee-catalina-8.0.0-M3.jar) to field java.lang.reflect.Field.modifiers
WARNING: Please consider reporting this to the maintainers of org.apache.tomee.catalina.ServerListener
WARNING: Use --illegal-access=warn to enable warnings of further illegal reflective access operations
WARNING: All illegal access operations will be denied in a future release
16-Sep-2020 21:40:08.782 INFO [main] jdk.internal.reflect.NativeMethodAccessorImpl.invoke Inicializando el manejador de protocolo ["http-nio-8087"]
```

- 8- En este punto mostraré las instancias registradas en el servidor Zookeeper y posteriormente indicaremos qué comandos pueden servir para observar estos registros.

```
[zk: localhost:2181(CONNECTED) 51] ls /services/orders-server
orders-server-8085, orders-server-8086, orders-server-8087]
[zk: localhost:2181(CONNECTED) 52]
```

- 9- Ahora podemos probar las instancias por separado en un navegador.



Este equipo > Disco local (C:) > apache-zookeeper-3.6.2-bin > bin				
Nombre	Fecha de modificación	Tipo	Tamaño	
README.txt	04/09/2020 7:43	Documento de te...	1 KB	
zkCleanup.sh	04/09/2020 7:43	Shell Script	3 KB	
zkCli.cmd	04/09/2020 7:43	Script de comand...	2 KB	
zkCli.sh	04/09/2020 7:43	Shell Script	2 KB	
zkEnv.cmd	13/09/2020 19:24	Script de comand...	2 KB	
zkEnv.sh	04/09/2020 7:43	Shell Script	4 KB	
zkServer.cmd	04/09/2020 7:43	Script de comand...	2 KB	
zkServer.sh	04/09/2020 7:43	Shell Script	11 KB	
zkServer-initialize.sh	04/09/2020 7:43	Shell Script	5 KB	
zkSnapShotToolkit.cmd	04/09/2020 7:43	Script de comand...	1 KB	
zkSnapShotToolkit.sh	04/09/2020 7:43	Shell Script	2 KB	
zkTxnLogToolkit.cmd	04/09/2020 7:43	Script de comand...	1 KB	
zkTxnLogToolkit.sh	04/09/2020 7:43	Shell Script	2 KB	

3- Los más importantes entre estos archivos son *zkServer.cmd*, *zkCli.cmd* y *zkEnv.cmd*

- zkServer.cmd*: Al ejecutar este archivo iniciaremos nuestro servidor Zookeeper y se mantendrá activo hasta que cerremos la ventana de CMD o presionemos la combinación de teclas Ctrl+C. El servidor Zookeeper utiliza el puerto 8080 y el 2181 como entrada para que un cliente se conecte al servidor.
- zkCli.cmd*: Cuando ejecutamos este archivo, se inicia un cliente que se conecta a nuestro servidor Zookeeper, es necesario haber ejecutado antes el servidor, y abrir otra ventana de CMD para ejecutar este archivo.
- zkEnv.cmd*: Este archivo, al ejecutarse, agrega y lee algunas variables de entorno necesarias para el funcionamiento del servidor, ya que el servidor ejecuta este archivo mientras se inicia.

```
C:\Windows\System32\cmd.exe - zkServer.cmd
Microsoft Windows [Versión 10.0.18362.720]
(c) 2019 Microsoft Corporation. Todos los derechos reservados.

C:\apache-zookeeper-3.6.2-bin\bin>zkServer.cmd

C:\apache-zookeeper-3.6.2-bin\bin>call "C:\Program Files\Java\jdk-11.0.8\bin\java.exe" "-Dzookeeper.log.
t.logger=INFO,CONSOLE" "-Dzookeeper.log.file=zookeeper-Famil-server-MASACHEBONILLA.log" "-XX:+HeapDumpOn
%p /t /f" -cp "C:\apache-zookeeper-3.6.2-bin\bin\..\build\classes;C:\apache-zookeeper-3.6.2-bin\bin\..\
ookeeper-3.6.2-bin\bin\..\lib\*;C:\apache-zookeeper-3.6.2-bin\bin\..\conf" org.apache.zookeeper.server.qu
zoo.cfg"
2020-09-16 10:49:43,972 [myid:] - INFO [main:QuorumPeerConfig@174] - Reading configuration from: C:\apa
2020-09-16 10:49:43,995 [myid:] - INFO [main:QuorumPeerConfig@460] - clientPortAddress is 0.0.0.0:2181
2020-09-16 10:49:43,995 [myid:] - INFO [main:QuorumPeerConfig@464] - secureClientPort is not set
2020-09-16 10:49:43,995 [myid:] - INFO [main:QuorumPeerConfig@480] - observerMasterPort is not set
2020-09-16 10:49:43,997 [myid:] - INFO [main:QuorumPeerConfig@497] - metricsProvider.className is org.a
2020-09-16 10:49:44,007 [myid:] - INFO [main:DatadirCleanupManager@78] - autopurge.snapRetainCount set
2020-09-16 10:49:44,008 [myid:] - INFO [main:DatadirCleanupManager@79] - autopurge.purgeInterval set to
2020-09-16 10:49:44,008 [myid:] - INFO [main:DatadirCleanupManager@101] - Purge task is not scheduled.
2020-09-16 10:49:44,008 [myid:] - WARN [main:QuorumPeerMain@138] - Either no config or no quorum define
2020-09-16 10:49:44,012 [myid:] - INFO [main:ManagedUtil@44] - Log4j 1.2 jmx support found and enabled.
2020-09-16 10:49:44,152 [myid:] - INFO [main:QuorumPeerConfig@174] - Reading configuration from: C:\apa
2020-09-16 10:49:44,153 [myid:] - INFO [main:QuorumPeerConfig@460] - clientPortAddress is 0.0.0.0:2181
2020-09-16 10:49:44,153 [myid:] - INFO [main:QuorumPeerConfig@464] - secureClientPort is not set
```



```
Selecciónar C:\Windows\System32\cmd.exe - zkCli.cmd
Microsoft Windows [Versión 10.0.18362.720]
(c) 2019 Microsoft Corporation. Todos los derechos reservados.

C:\apache-zookeeper-3.6.2-bin\bin>zkCli.cmd
Connecting to localhost:2181
2020-09-16 10:50:06,927 [myid:] - INFO [main:Environment@98] - Client environment:zookeeper.version=3.6.2--803c7f1a12f85978cb
20 12:44 GMT
2020-09-16 10:50:06,930 [myid:] - INFO [main:Environment@98] - Client environment:host.name=MasacheBonilla
2020-09-16 10:50:06,931 [myid:] - INFO [main:Environment@98] - Client environment:java.version=11.0.8
2020-09-16 10:50:06,932 [myid:] - INFO [main:Environment@98] - Client environment:java.vendor=Oracle Corporation
2020-09-16 10:50:06,933 [myid:] - INFO [main:Environment@98] - Client environment:java.home=C:\Program Files\Java\jdk-11.0.8
2020-09-16 10:50:06,933 [myid:] - INFO [main:Environment@98] - Client environment:java.class.path=C:\apache-zookeeper-3.6.2-b
er-3.6.2-bin\bin\..\build\lib*;C:\apache-zookeeper-3.6.2-bin\bin\..\*;C:\apache-zookeeper-3.6.2-bin\bin\..\lib\audience-annot
2-bin\bin\..\lib\commons-cli-1.2.jar;C:\apache-zookeeper-3.6.2-bin\bin\..\lib\commons-lang-2.6.jar;C:\apache-zookeeper-3.6.2-b
jar;C:\apache-zookeeper-3.6.2-bin\bin\..\lib\jackson-core-2.10.3.jar;C:\apache-zookeeper-3.6.2-bin\bin\..\lib\jackson-databind
\bin\..\lib\javax.servlet-api-3.1.0.jar;C:\apache-zookeeper-3.6.2-bin\bin\..\lib\jetty-http-9.4.24.v20191120.jar;C:\apache-zoo
24.v20191120.jar;C:\apache-zookeeper-3.6.2-bin\bin\..\lib\jetty-security-9.4.24.v20191120.jar;C:\apache-zookeeper-3.6.2-bin\bi
r;C:\apache-zookeeper-3.6.2-bin\bin\..\lib\jetty-servlet-9.4.24.v20191120.jar;C:\apache-zookeeper-3.6.2-bin\bin\..\lib\jetty-u
per-3.6.2-bin\bin\..\lib\jline-2.14.6.jar;C:\apache-zookeeper-3.6.2-bin\bin\..\lib\json-simple-1.1.1.jar;C:\apache-zookeeper-3
apache-zookeeper-3.6.2-bin\bin\..\lib\metrics-core-3.2.5.jar;C:\apache-zookeeper-3.6.2-bin\bin\..\lib\netty-buffer-4.1.50.Final
\lib\netty-codec-4.1.50.Final.jar;C:\apache-zookeeper-3.6.2-bin\bin\..\lib\netty-common-4.1.50.Final.jar;C:\apache-zookeeper-3
```

- 4- Con el cliente de Zookeeper iniciado podemos ejecutar algunos comandos y la forma en que se guardan valores aquí es jerárquica, inicialmente solo hay un nodo que es el nodo *zookeeper*, pero claro podemos crear más y de hecho en un nodo al que llamaremos *services* es en donde registraremos nuestros servicios posteriormente.
- 5- Con el comando *help* en el cliente, podemos obtener una lista de comandos válidos para que el cliente interactúe con el servidor zookeeper.

```
WATCHER::

WatchedEvent state:SyncConnected type:None path:null
[zk: localhost:2181(CONNECTED) 0] help
ZooKeeper -server host:port -client-configuration properties-file cmd args
  addwatch [-m mode] path # optional mode is one of [PERSISTENT, PERS
  addauth scheme auth
  close
  config [-c] [-w] [-s]
  connect host:port
  create [-s] [-e] [-c] [-t ttl] path [data] [acl]
  delete [-v version] path
  deleteall path [-b batch size]
  delquota [-n|-b] path
  get [-s] [-w] path
  getAcl [-s] path
  getAllChildrenNumber path
  getEphemerals path
  history
  listquota path
  ls [-s] [-w] [-R] path
  printwatches on|off
  quit
  reconfig [-s] [-v version] [[-file path] | [-members serverID=host:
[,...]*]
  redo cmdno
  removewatches path [-c|-d|-a] [-l]
  set [-s] [-v version] path data
  setAcl [-s] [-v version] [-R] path acl
  setquota -n|-b val path
  stat [-w] path
  sync path
  version
Command not found: Command not found help
[zk: localhost:2181(CONNECTED) 1] _
```

- 6- Los comandos que más nos interesan para el propósito de configuración y de registro, son:



- a. *ls*: Para realizar comprobaciones de que nuestros servicios se han registrado correctamente, específicamente utilizaremos *ls /services*. Podemos listar cada vez dentro de cualquier “directorio” que tenga nodos dentro, en caso de que esté vacío solo retornará un par de corchetes []. Si en este punto también listamos un servicio en específico nos devolverá una lista con las instancias levantadas de ese servicio.

```
[zk: localhost:2181(CONNECTED) 5] ls /services  
[customer-server, orders-server]  
[zk: localhost:2181(CONNECTED) 6]
```

```
[zk: localhost:2181(CONNECTED) 18] ls /services/customer-server  
[customer-server-8083, customer-server-8085]  
[zk: localhost:2181(CONNECTED) 19]
```

- b. *create*: Para poder guardar algunas variables de configuración en un nodo que podemos llamarlo como deseemos, utilizamos la siguiente lista de comandos para guardar cierta información de configuración en Zookeeper:

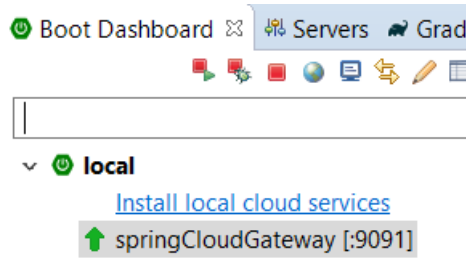
```
[customer-server, orders-server]  
[zk: localhost:2181(CONNECTED) 6] create /conf-db  
Created /conf-db  
[zk: localhost:2181(CONNECTED) 7] create /conf-db/url jdbc:postgresql://127.0.0.1:5432/distribuidadb  
Created /conf-db/url  
[zk: localhost:2181(CONNECTED) 8] create /conf-db/username postgres  
Created /conf-db/username  
[zk: localhost:2181(CONNECTED) 9] create /conf-db/password clave  
Created /conf-db/password  
[zk: localhost:2181(CONNECTED) 10] _
```

- c. *get*: Este comando nos permite obtener el valor guardado en un path, si lo tiene, retornará el contenido, si no retornará null.

```
[zk: localhost:2181(CONNECTED) 13] get /conf-db  
null  
[zk: localhost:2181(CONNECTED) 14] get /conf-db/url  
jdbc:postgresql://127.0.0.1:5432/distribuidadb  
[zk: localhost:2181(CONNECTED) 15] get /conf-db/username  
postgres  
[zk: localhost:2181(CONNECTED) 16] get /conf-db/password  
clave  
[zk: localhost:2181(CONNECTED) 17] _
```

4. Despliegue de Spring Cloud Gateway

Usamos el Spring tolos Suite para ejecutar el Gateway de la siguiente manera , damos clic en la icono de Run del la ventana de Boot Dashboard



Y se nos levantara en nuestro caso en el puerto 9091

```
Spring Boot (v2.3.3.RELEASE)
2020-09-16 20:32:20.834 INFO 18200 --- [main] c.d.gateway.GatewayApplication : No active profile set, falling back to default profiles: def
2020-09-16 20:32:21.500 INFO 18200 --- [main] o.s.c.cloud.context.scope.GenericScope : BeanFactory id=13c8bd33-a322-3658-b5c8-4e7e81b826f9
2020-09-16 20:32:21.550 INFO 18200 --- [main] trationDelegateBeanPostProcessorChecker : Bean 'org.springframework.cloud.client.loadbalancer.reactive
2020-09-16 20:32:21.551 INFO 18200 --- [main] trationDelegateBeanPostProcessorChecker : Bean 'org.springframework.cloud.client.loadbalancer.reactive
2020-09-16 20:32:21.552 INFO 18200 --- [main] trationDelegateBeanPostProcessorChecker : Bean 'reactorDeferingLoadBalancerExchangeFilterFunction' of
2020-09-16 20:32:21.552 INFO 18200 --- [main] org.apache.zookeeper.ZooKeeper : Client environment:zookeeper.version=3.5.3-beta-8ce24f9e675c
2020-09-16 20:32:21.552 INFO 18200 --- [main] org.apache.zookeeper.ZooKeeper : Client environment:host.name=DESKTOP-BJHNVSC
2020-09-16 20:32:21.552 INFO 18200 --- [main] org.apache.zookeeper.ZooKeeper : Client environment:java.version=11.0.7
2020-09-16 20:32:21.552 INFO 18200 --- [main] org.apache.zookeeper.ZooKeeper : Client environment:java.vendor=Oracle Corporation
2020-09-16 20:32:21.552 INFO 18200 --- [main] org.apache.zookeeper.ZooKeeper : Client environment:java.home=C:\Program Files\Java\jdk-11.0
2020-09-16 20:32:21.552 INFO 18200 --- [main] org.apache.zookeeper.ZooKeeper : Client environment:java.class.path=D:\universidad\ultimo sem
2020-09-16 20:32:21.552 INFO 18200 --- [main] org.apache.zookeeper.ZooKeeper : Client environment:java.library.path=C:\Program Files\Java\
2020-09-16 20:32:21.552 INFO 18200 --- [main] org.apache.zookeeper.ZooKeeper : Client environment:java.io.tmpdir=C:\Users\alxx\AppData\Loc
2020-09-16 20:32:21.552 INFO 18200 --- [main] org.apache.zookeeper.ZooKeeper : Client environment:java.compiler=DAW
2020-09-16 20:32:21.552 INFO 18200 --- [main] org.apache.zookeeper.ZooKeeper : Client environment:os.name=Windows 10
2020-09-16 20:32:21.552 INFO 18200 --- [main] org.apache.zookeeper.ZooKeeper : Client environment:os.arch=amd64
2020-09-16 20:32:21.552 INFO 18200 --- [main] org.apache.zookeeper.ZooKeeper : Client environment:os.version=10.0
2020-09-16 20:32:21.552 INFO 18200 --- [main] org.apache.zookeeper.ZooKeeper : Client environment:user.name=alxx
2020-09-16 20:32:21.552 INFO 18200 --- [main] org.apache.zookeeper.ZooKeeper : Client environment:user.home=C:\Users\alxx
2020-09-16 20:32:21.552 INFO 18200 --- [main] org.apache.zookeeper.ZooKeeper : Client environment:user.dir=D:\universidad\ultimo semestre\D
2020-09-16 20:32:21.552 INFO 18200 --- [main] org.apache.zookeeper.ZooKeeper : Client environment:os.memory.free=519M
2020-09-16 20:32:21.552 INFO 18200 --- [main] org.apache.zookeeper.ZooKeeper : Client environment:os.memory.max=2024M
2020-09-16 20:32:21.552 INFO 18200 --- [main] org.apache.zookeeper.ZooKeeper : Client environment:os.memory.total=1289M
2020-09-16 20:32:21.552 INFO 18200 --- [main] org.apache.zookeeper.ZooKeeper : Starting
2020-09-16 20:32:21.552 INFO 18200 --- [main] org.apache.zookeeper.ZooKeeper : Initiating client connection, connectString=localhost:2181 s
2020-09-16 20:32:21.552 INFO 18200 --- [main] org.apache.zookeeper.ZooKeeper : jute.maxbuffer value is 4194304 bytes
2020-09-16 20:32:21.552 INFO 18200 --- [main] org.apache.zookeeper.ZooKeeper : Opening socket connection to server localhost/127.0.0.1:2181
2020-09-16 20:32:21.552 INFO 18200 --- [main] org.apache.zookeeper.ZooKeeper : Socket connection established, initiating session, client: /
```

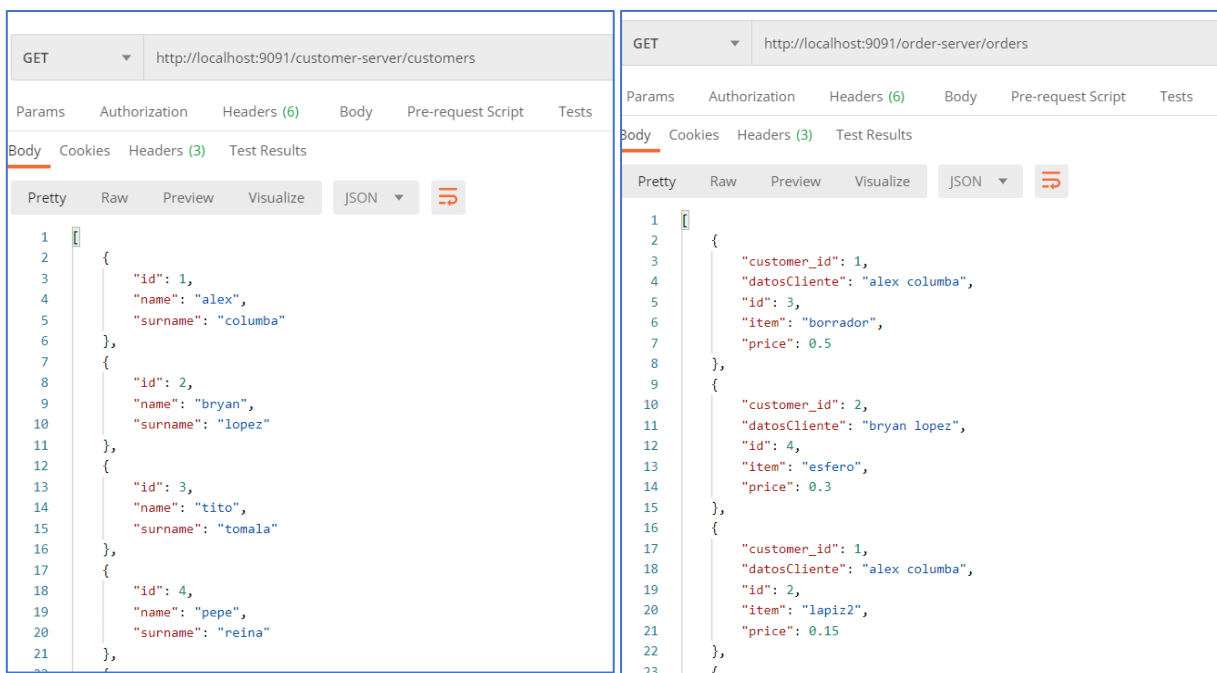
Y ahora ya podremos acceder desde nuestro Gateway a los servicios de órdenes y customers en este caso desde Postman.

La forma de acceder es la siguiente para clientes y ordenes:

<http://localhost:9191/{nombre-servicio}/{metodo-Rest}>

customers : <http://localhost:9091/customer-server/customers>

ordenes : <http://localhost:9091/order-server/orders>





5. Despliegue del Cliente Web

Ahora nuestro cliente web podrá acceder a los servicios desde el Gateway con esto nos aseguramos que el Gateway balancee la carga a las diferentes instancias de los clientes y ordenes.

Para esto se realizará mediante el IDE STS y el servidor apache tomcat 9.0.36 antes de desplegar se deberá cambiar de puertos ya que el 8080 está ocupado por ZooKeeper para esto vamos donde se encuentra instalado en servidor tomcat y se debe dirigir a conf/server.xml y cambiar los puertos que sean necesarios.

Luego se procede a correr el proyecto con apache 9.x

```
grupo8 [Started, Synchronized]
cliente-resteasy-jsf [Synchronized]

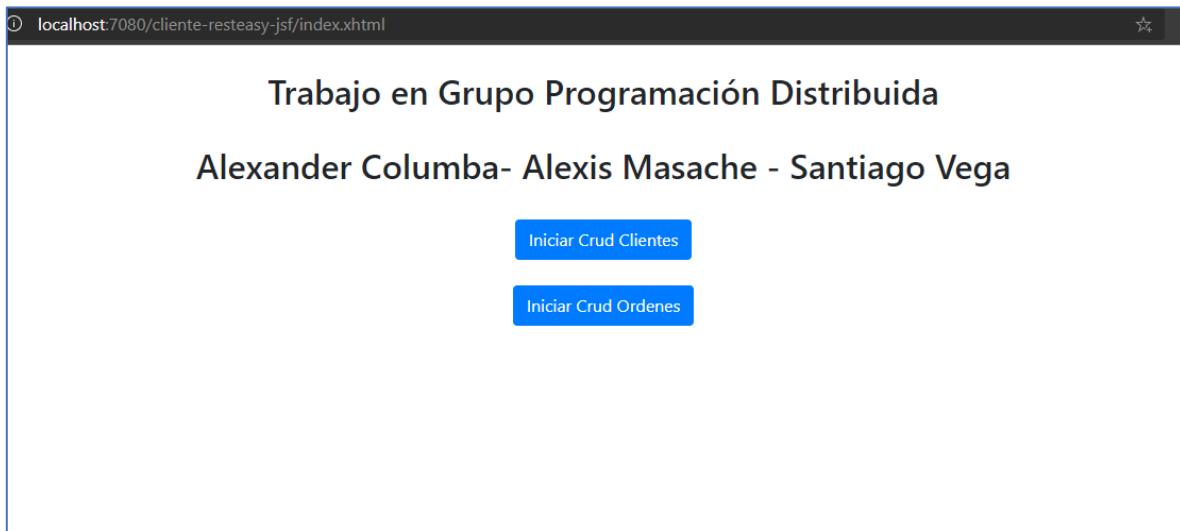
Console
Gradle Executions

grupo8 [Apache Tomcat] C:\Program Files\Java\jdk-11.0.7\bin\javaw.exe (16 sep. 2020 20:32:42)
INFO: Command line argument: -Dfile.encoding=Cp1252
sep. 16, 2020 8:32:43 P.M. org.apache.catalina.startup.VersionLoggerListener log
INFO: Command line argument: -Dfile.encoding=Cp1252
sep. 16, 2020 8:32:43 P.M. org.apache.catalina.core.AprLifecycleListener lifecycleEvent
INFO: La biblioteca nativa de Apache Tomcat basada en ARP que permite un rendimiento óptimo en entornos de desar
sep. 16, 2020 8:32:44 P.M. org.apache.coyote.AbstractProtocol init
INFO: Inicializando el manejador de protocolo ["http-nio-7080"]

sep. 16, 2020 8:32:44 P.M. org.apache.catalina.startup.Catalina load
INFO: Server initialization in [756] milliseconds
sep. 16, 2020 8:32:44 P.M. org.apache.catalina.core.StandardService startInternal
INFO: Arrancando servicio [Catalina]
sep. 16, 2020 8:32:44 P.M. org.apache.catalina.core.StandardEngine startInternal
INFO: Starting Servlet engine: [Apache Tomcat/9.0.36]
sep. 16, 2020 8:32:44 P.M. org.apache.catalina.util.SessionIdGeneratorBase createSecureRandom
WARNING: Creation of SecureRandom instance for session ID generation using [SHA1PRNG] took [140] milliseconds.
sep. 16, 2020 8:32:47 P.M. org.apache.jasper.servlet.TldScanner scanJars
INFO: Al menos un JAR, que se ha explorado buscando TLDs, aún no contenía TLDs. Activar historial de depuración
sep. 16, 2020 8:32:47 P.M. org.jboss.weld.environment.servlet.EnhancedListener onStartUp
INFO: WELD-ENV-001008: Initialize Weld using ServletContainerInitializer
sep. 16, 2020 8:32:47 P.M. org.jboss.weld.bootstrap.WeldStartup <clinit>
INFO: WELD-000900: 3.1.0 (Final)
sep. 16, 2020 8:32:47 P.M. org.jboss.weld.environment.deployment.discovery.ReflectionDiscoveryStrategy processAr
INFO: WELD-ENV-000014: Falling back to Java Reflection for bean-discovery-mode="annotated" discovery. Add org.j
sep. 16, 2020 8:32:48 P.M. org.jboss.weld.bootstrap.WeldStartup startContainer
INFO: WELD-000101: Transactional services not available. Injection of @Inject UserTransaction not available. Tra
sep. 16, 2020 8:32:48 P.M. org.jboss.weld.bootstrap.events.BeforeBeanDiscoveryImpl addAnnotatedType
WARN: WELD-000146: BeforeBeanDiscovery.addAnnotatedType(AnnotatedType<?>) used for class com.sun.faces.flow.Flow
sep. 16, 2020 8:32:48 P.M. org.jboss.weld.environment.tomcat.TomcatContainer initialize
INFO: WELD-ENV-001100: Tomcat 7+ detected, CDI injection will be available in Servlets, Filters and Listeners.
sep. 16, 2020 8:32:48 P.M. com.sun.faces.config.ConfigureListener contextInitialized
INFO: Inicializando Mojarra 2.4.0 para el contexto '/cliente-resteasy-jsf'
sep. 16, 2020 8:32:49 P.M. com.sun.faces.spi.InjectionProviderFactory createInstance
INFO: JSF1048: hay presentes anotaciones PostConstruct/PreDestroy. Los métodos de beans administrados marcados
sep. 16, 2020 8:32:49 P.M. org.apache.coyote.AbstractProtocol start
INFO: Starting ProtocolHandler ["http-nio-7080"]
sep. 16, 2020 8:32:49 P.M. org.apache.catalina.startup.Catalina start
INFO: Server startup in [5.342] milliseconds
```

Se nos abrirá un navegador web automáticamente o ir a :

<http://localhost:7080/cliente-resteasy-jsf/index.xhtml>



Damos clic en Iniciar Crud Clientes y se nos mostrara una tabla con todos los clientes



Podremos buscar por sus apellidos desde el combo box



Seleccione un Apellido		columba	▼	Buscar
Id	Nombre	Apellido	Opciones	
1	alex	columba	Editar	Eliminar
12	ramiro	columba	Editar	Eliminar

Podremos crear , editar y eliminar clientes :

EDITAR CLIENTE

Nombre	<input type="text" value="alex"/>
Apellido	<input type="text" value="columba"/>

GuardarCancelar

CREAR NUEVA CLIENTE

Nombre	<input type="text"/>
Apellido	<input type="text"/>

GuardarCancelar

Ahora procedemos a probar el microservicio de Ordenes

GESTION DE ORDENES

Listado de las Ordenes

Crear Ordenes

Seleccione un Cliente		todos	▼	Buscar	
Id	Producto	Precio	Cliente	Datos Cliente	Opciones
3	borrador	0.5	1	alex columba	Editar Eliminar
4	esfero	0.3	2	bryan lopez	Editar Eliminar
2	lapiz2	0.15	1	alex columba	Editar Eliminar
7	mesa	30.4	2	bryan lopez	Editar Eliminar
9	cama	0.89	2	bryan lopez	Editar Eliminar
10	computadora	358.0	3	tito tomala	Editar Eliminar

Regresar menu principal



Así mismo podremos filtrar las ordenes por los clientes del combo box

GESTION DE ORDENES

Listado de las Ordenes

Crear Ordenes

Seleccione un Cliente

1 alex columba

Buscar

Id	Producto	Precio	Cliente	Datos Cliente	Opciones
3	borrador	0.5	1	alex columba	<div>Editar</div> <div>Eliminar</div>
2	lapiz2	0.15	1	alex columba	<div>Editar</div> <div>Eliminar</div>

Regresar menu principal

Y podremos crear, actualizar y eliminar ordenes

CREAR NUEVA ORDEN

Producto

Precio

Cliente

Guardar

Cancelar

EDITAR ORDEN

Producto

borrador

Precio

0.5

Cliente

Guardar

Cancelar



6. Conclusiones

EL Gateway de Spring Cloud integrado con ZooKeeper nos provisionaron de las partes más fundamentales de la arquitectura de microservicios como registro, descubrimiento, configuración, Gateway y balanceador de carga.

Hemos visto que una gran ventaja de los microservicios es que si se quiere modificar un servicio no es necesario alterar el resto de la infraestructura, cada microservicio se puede desplegar y modificar sin que ello afecte a otros microservicios, y que además se puede escalar horizontalmente según sea necesario.

Se pueden usar diferentes tecnologías y lenguajes de programación. Lo que permite adaptar cada funcionalidad a la tecnología más adecuada y rentable, así pues, los microservicios puede usar contruidos en distinto framework o lenguajes de programación, una parte negativa que observamos es que es más difícil la gestión y para esto necesitaremos de más tecnologías para poder administrarlos y monitorear correctamente