

Manual de Implementación de arquitectura Micro-Servicios

Laura Cristina Becerra González

lauracriss@gmail.com

Universidad Distrital Francisco José de Caldas

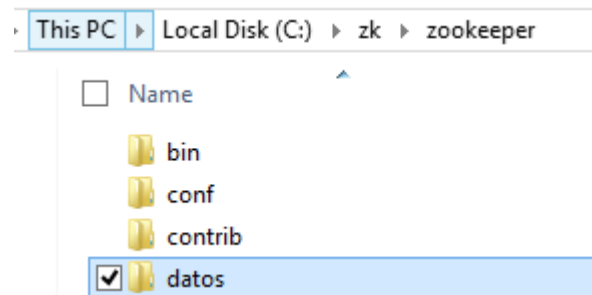
Tendencias de Software – Mayo 2016

Contenido

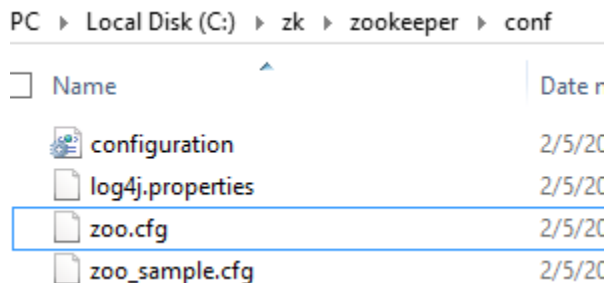
Instalación de zeekeeper	3
Configuración servidor Apache	8
Configuración servicio en PHP	10
Creación de base de datos:	12
Configuración del Balanceador De Carga en PHP:	12
Prueba de uso y transferencia las otras maquinas.....	14

Instalación de zookeeper

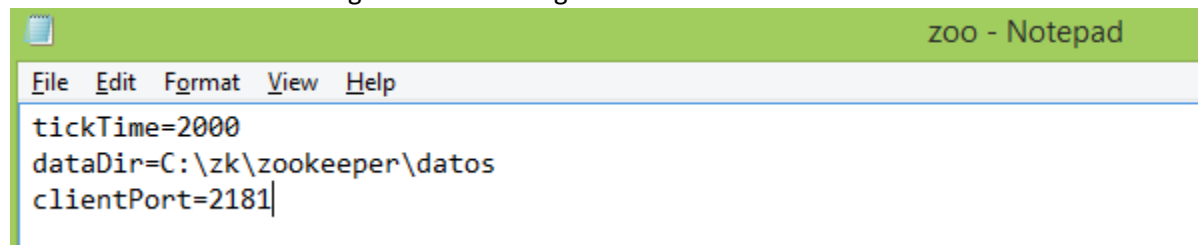
1. Se debe descargar el balanceador de carga, para este caso, se va a utilizar zookeeper, se descarga de la página oficial: <http://www.apache.org/dyn/closer.cgi/zookeeper/>
2. Después de descargar descomprimir, en una carpeta raíz c:\ crear una carpeta denominada zk y adentro copiar los archivos en la carpeta zookeeper. Seguido a ello, nuevamente estando en la carpeta zookeeper crear una carpeta denominada datos:



3. Se debe crear el archivo de configuración ubicado en la carpeta conf, es posible utilizar el archivo de ejemplo:



4. Modificar el archivo de configuración con la siguiente información:



5. Descargar el JDK de java de: <https://www.java.com/es/download/>

Descarga gratuita de Java

Descargue Java para su computadora de escritorio ahora

Version 8 Update 91

Fecha de lanzamiento: 19 de abril de 2016

[Descarga gratuita de Java](#)

[» ¿Qué es Java?](#) [» ¿Tengo Java?](#) [» ¿Necesita ayuda?](#)

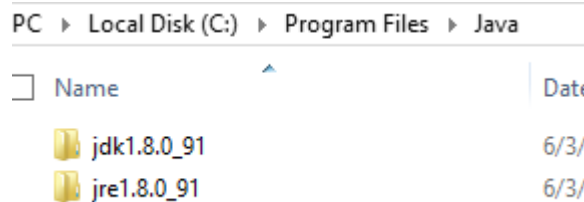
Java SE Development Kit 8u91

You must accept the [Oracle Binary Code License Agreement for Java SE](#) to download this software.

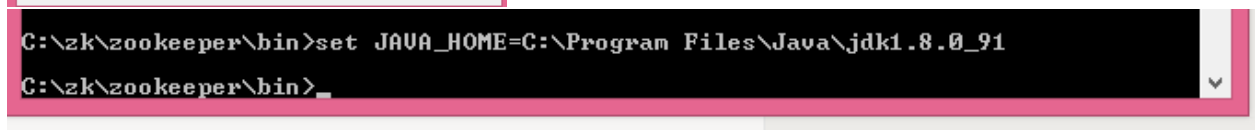
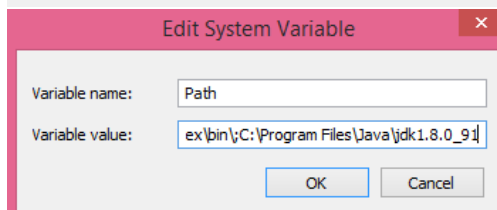
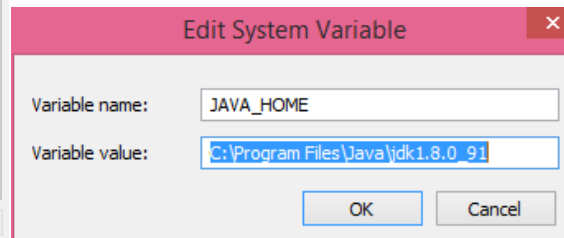
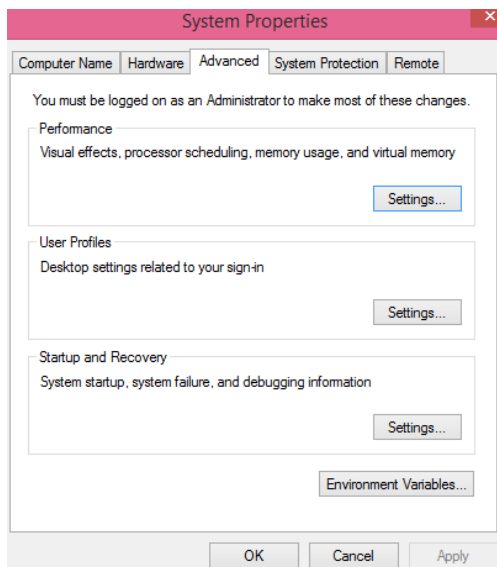
Thank you for accepting the Oracle Binary Code License Agreement for Java SE; you may now download this software.

Product / File Description	File Size	Download
Linux ARM 32 Hard Float ABI	77.72 MB	jdk-8u91-linux-arm32-vfp-hflt.tar.gz
Linux ARM 64 Hard Float ABI	74.69 MB	jdk-8u91-linux-arm64-vfp-hflt.tar.gz
Linux x86	154.74 MB	jdk-8u91-linux-i586.rpm
Linux x86	174.92 MB	jdk-8u91-linux-i586.tar.gz
Linux x64	152.74 MB	jdk-8u91-linux-x64.rpm
Linux x64	172.97 MB	jdk-8u91-linux-x64.tar.gz
Mac OS X	227.29 MB	jdk-8u91-macosx-x64.dmg
Solaris SPARC 64-bit (SVR4 package)	139.59 MB	jdk-8u91-solaris-sparcv9.tar.Z
Solaris SPARC 64-bit	98.95 MB	jdk-8u91-solaris-sparcv9.tar.gz
Solaris x64 (SVR4 package)	140.29 MB	jdk-8u91-solaris-x64.tar.Z
Solaris x64	96.78 MB	jdk-8u91-solaris-x64.tar.gz
Windows x86	182.29 MB	jdk-8u91-windows-i586.exe
Windows x64	187.4 MB	jdk-8u91-windows-x64.exe

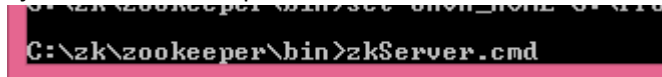
6. Se debe instalar java y luego establecer la variable java_home:



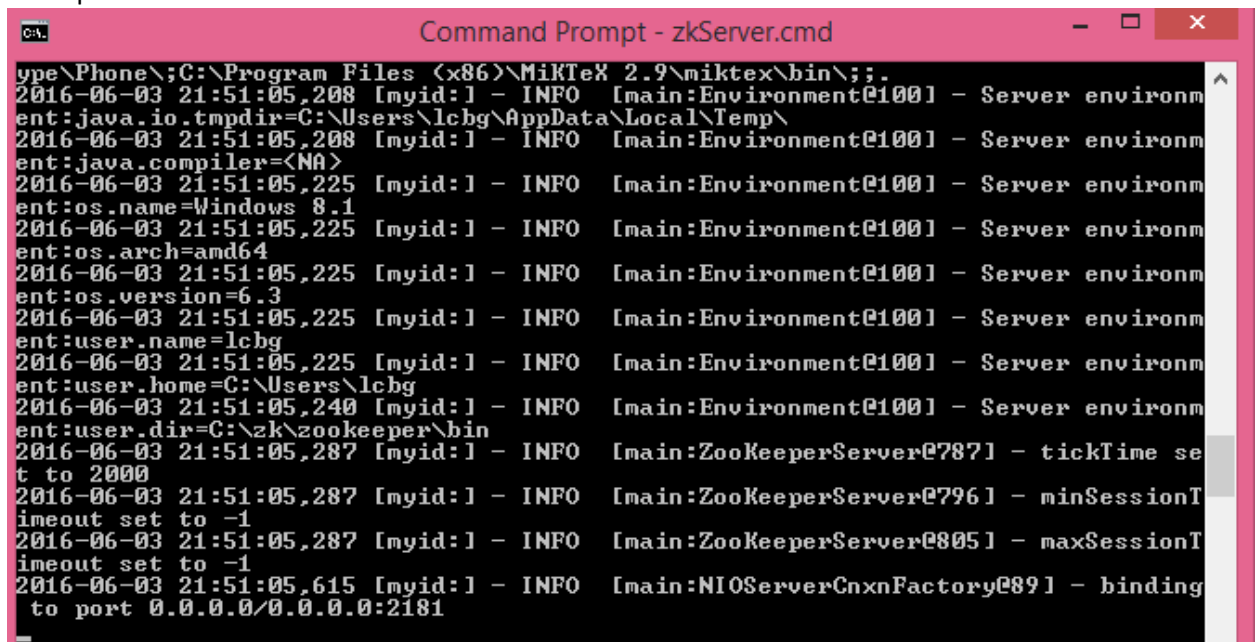
- clic derecho en el icono de "mi PC" y clic en la opción "propiedades".
- clic en la pestaña "avanzado"
- clic en el botón "variables de entorno"
- Bajo la variable "sistema", clic en "nuevo"
- Ingresa el nombre de la variable como JAVA_HOME
- Ingresa el valor de variable como la ruta de instalación, "C:\Program Files\Java\jdk1.8.0_91"
- clic en "ok"
- clic en "aplicar cambios"
- Reinicia la computadora para asegurarte de que los cambios estén reflejados.



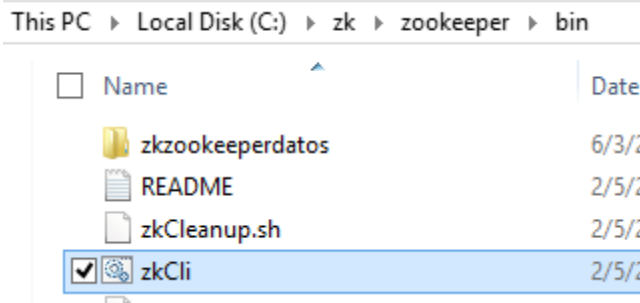
7. Ejecutar el servidor por medio del archivo zkServer.cmd del directorio raíz, bin:



La respuesta debe ser:



8. Ejecutar el cliente por medio del siguiente archivo:



```
Command Prompt - zkServer.cmd
2016-06-03 21:51:05.287 [myid:1] - INFO [main:ZooKeeperServer@796] - minSessionT
imeout set to -1
2016-06-03 21:51:05.287 [myid:1] - INFO [main:ZooKeeperServer@805] - maxSessionT
imeout set to -1
2016-06-03 21:51:05.615 [myid:1] - INFO [main:NIOServerCnxnFactory@89] - binding
to port 0.0.0.0/0.0.0.0:2181
2016-06-03 21:53:03.567 [myid:1] - INFO [NIOServerCnxn.Factory:0.0.0.0/0.0.0.0:21
81:NIOServerCnxnFactory@192] - Accepted socket connection from /0:0:0:0:0:0:1:
59731
2016-06-03 21:53:03.646 [myid:1] - INFO [NIOServerCnxn.Factory:0.0.0.0/0.0.0.0:21
81:ZooKeeperServer@900] - Client attempting to establish new session at /0:0:0:0
:0:0:0:1:59731
2016-06-03 21:53:03.661 [myid:1] - INFO [SyncThread:0:FileTxnLog@199] - Creating
new log file: log.1
2016-06-03 21:53:03.771 [myid:1] - INFO [SyncThread:0:ZooKeeperServer@645] - Est
ablished session 0x1551952344e0000 with negotiated timeout 30000 for client /0:0
:0:0:0:0:0:1:59731
2016-06-03 21:55:15.663 [myid:1] - WARN [NIOServerCnxn.Factory:0.0.0.0/0.0.0.0:21
81:NIOServerCnxn@362] - Exception causing close of session 0x1551952344e0000 due
to java.io.IOException: An existing connection was forcibly closed by the remot
e host
2016-06-03 21:55:15.680 [myid:1] - INFO [NIOServerCnxn.Factory:0.0.0.0/0.0.0.0:21
81:NIOServerCnxn@1008] - Closed socket connection for client /0:0:0:0:0:0:1:59
731 which had sessionid 0x1551952344e0000
2016-06-03 21:55:44.013 [myid:1] - INFO [SessionTracker:ZooKeeperServer@355] - E
xpiring session 0x1551952344e0000, timeout of 30000ms exceeded
2016-06-03 21:55:44.013 [myid:1] - INFO [ProcessThread(sid:0 cport:2181)::PrepRe
questProcessor@489] - Processed session termination for sessionid: 0x1551952344e
0000
```

Al ejecutar el cliente el servidor debe activar la petición y cambiar con respecto a la toma del punto 7:

```
C:\Windows\system32\cmd.exe
2016-06-03 21:56:16.403 [myid:1] - INFO [main:Environment@100] -
ent:user.name=lcby
2016-06-03 21:56:16.418 [myid:1] - INFO [main:Environment@100] -
ent:user.home=C:\Users\lcby
2016-06-03 21:56:16.418 [myid:1] - INFO [main:Environment@100] -
ent:user.dir=C:\zk\zookeeper\bin
2016-06-03 21:56:16.418 [myid:1] - INFO [main:ZooKeeper@438] - In
connection, connectString=localhost:2181 sessionTimeout=30000 us
e.zookeeper.ZooKeeperMain$MyWatcher@506c589e
Welcome to ZooKeeper!
2016-06-03 21:56:16.825 [myid:1] - INFO [main:SendThread@127.0.0.
1:181] - Opening socket connection to server 127.0.0.1:2181. Will not attempt to authenticate using SASL (unknown error)
2016-06-03 21:56:16.825 [myid:1] - INFO [main:SendThread@127.0.0.
1:181] - Socket connection established to 127.0.0.1:2181. Initiating session
JLine support is enabled
2016-06-03 21:56:16.887 [myid:1] - INFO [main:SendThread@127.0.0.
1:181] - Session establishment complete on server 127.0.0.1:2181, sessionid = 0x1551952344e0001, negotiated timeout = 30000
WATCHER::
WatchedEvent state:SyncConnected type:None path:null
[zookeeper:localhost:2181(CONNECTED) 0]
```

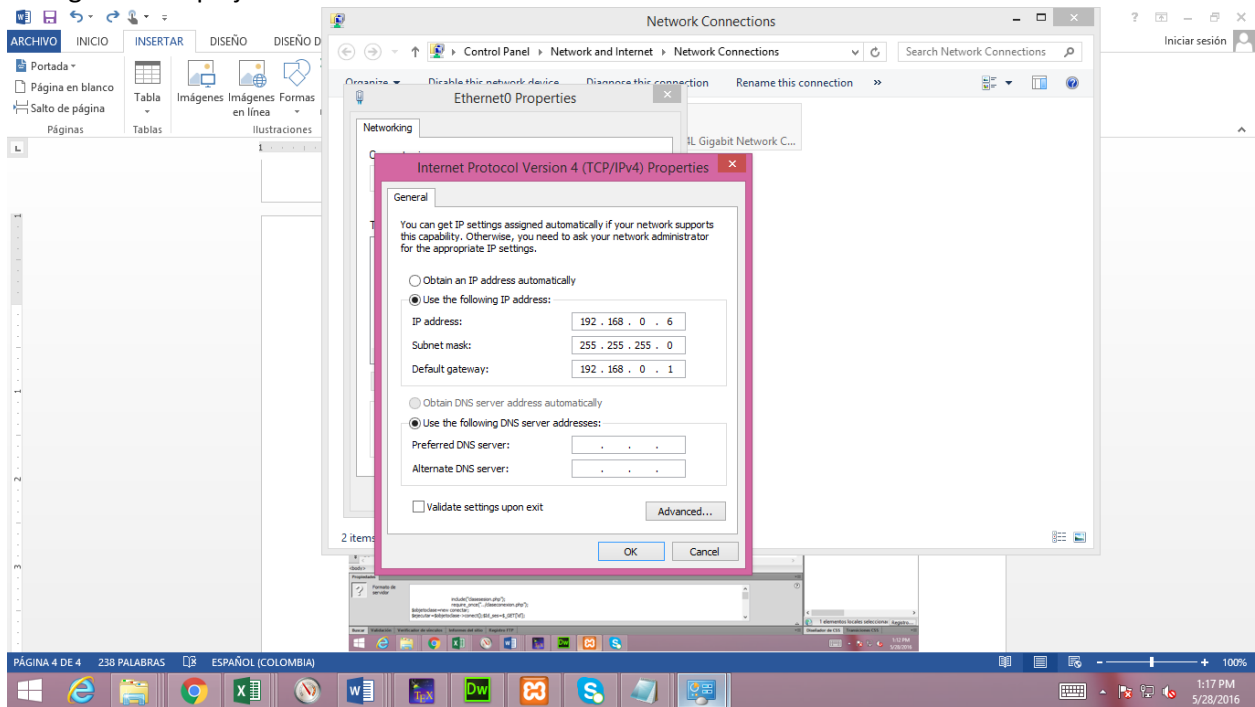
Y al cerrar el cliente, el servidor debe mostrar:

```
C:\Windows\system32\cmd.exe
2016-06-03 21:53:03.130 [myid:1] - INFO [main:Environment@100] - Client environment: user.home=C:\Users\lcbg
2016-06-03 21:53:03.130 [myid:1] - INFO [main:Environment@100] - Client environment: user.dir=C:\zk\zookeeper\bin
2016-06-03 21:53:03.146 [myid:1] - INFO [main:ZooKeeper@438] - Initiating client connection, connectString=localhost:2181 sessionTimeout=30000 watcher=org.apache.zookeeper.ZooKeeperMain$MyWatcher@506c589e
Welcome to ZooKeeper!
2016-06-03 21:53:03.552 [myid:1] - INFO [main:SendThread@0:0:0:0:0:1:2181] - ClientConn$SendThread@10321 - Opening socket connection to server 0:0:0:0:0:0:0:0:1:2181. Will not attempt to authenticate using SASL (unknown error)
2016-06-03 21:53:03.567 [myid:1] - INFO [main:SendThread@0:0:0:0:0:1:2181] - Socket connection established to 0:0:0:0:0:0:0:0:1:2181. Initiating session
2016-06-03 21:53:03.771 [myid:1] - INFO [main:SendThread@0:0:0:0:0:1:2181] - ClientConn$SendThread@1292 - Session establishment complete on server 0:0:0:0:0:0:0:0:1:2181, sessionId = 0x1551952344e0000, negotiated timeout = 30000
WATCHER::
WatchedEvent state:SyncConnected type:None path:null
[ek: localhost:2181(CONNECTED) 0]
```

```
Command Prompt - zkServer.cmd
C:\zk\zookeeper\bin\.\lib\log4j-1.6.1.jar;C:\zk\zookeeper\bin\.\conf
2016-06-03 21:51:05.193 [myid:1] - INFO [main:Environment@100] - Server environment: java.library.path=C:\Program Files\Java\jdk1.8.0_91\bin;C:\Windows\Sun\Java\bin;C:\Windows\system32\bin;C:\Windows\system32\cmd;C:\Windows\system32\WindowsPowerShell\v1.0\cmd;C:\Program Files (x86)\Skype\Phone\bin;C:\Program Files (x86)\MIKTeX 2.9\miktex\bin\;
2016-06-03 21:51:05.208 [myid:1] - INFO [main:Environment@100] - Server environment: java.io.tmpdir=C:\Users\lcbg\AppData\Local\Temp\
2016-06-03 21:51:05.208 [myid:1] - INFO [main:Environment@100] - Server environment: java.compile=NO
2016-06-03 21:51:05.225 [myid:1] - INFO [main:Environment@100] - Server environment: os.name=Windows 8.1
2016-06-03 21:51:05.225 [myid:1] - INFO [main:Environment@100] - Server environment: os.arch=amd64
2016-06-03 21:51:05.225 [myid:1] - INFO [main:Environment@100] - Server environment: os.version=6.3
2016-06-03 21:51:05.225 [myid:1] - INFO [main:Environment@100] - Server environment: user.name=lcbg
2016-06-03 21:51:05.225 [myid:1] - INFO [main:Environment@100] - Server environment: user.home=C:\Users\lcbg
2016-06-03 21:51:05.240 [myid:1] - INFO [main:Environment@100] - Server environment: user.dir=C:\zk\zookeeper\bin
2016-06-03 21:51:05.287 [myid:1] - INFO [main:ZooKeeperServer@787] - tickTime set to 2000
2016-06-03 21:51:05.287 [myid:1] - INFO [main:ZooKeeperServer@796] - minSessionTimeout set to -1
2016-06-03 21:51:05.287 [myid:1] - INFO [main:ZooKeeperServer@805] - maxSessionTimeout set to -1
2016-06-03 21:51:05.615 [myid:1] - INFO [main:NIOServerCnxnFactory@889] - binding to port 0.0.0.0:0.0.0:2181
```

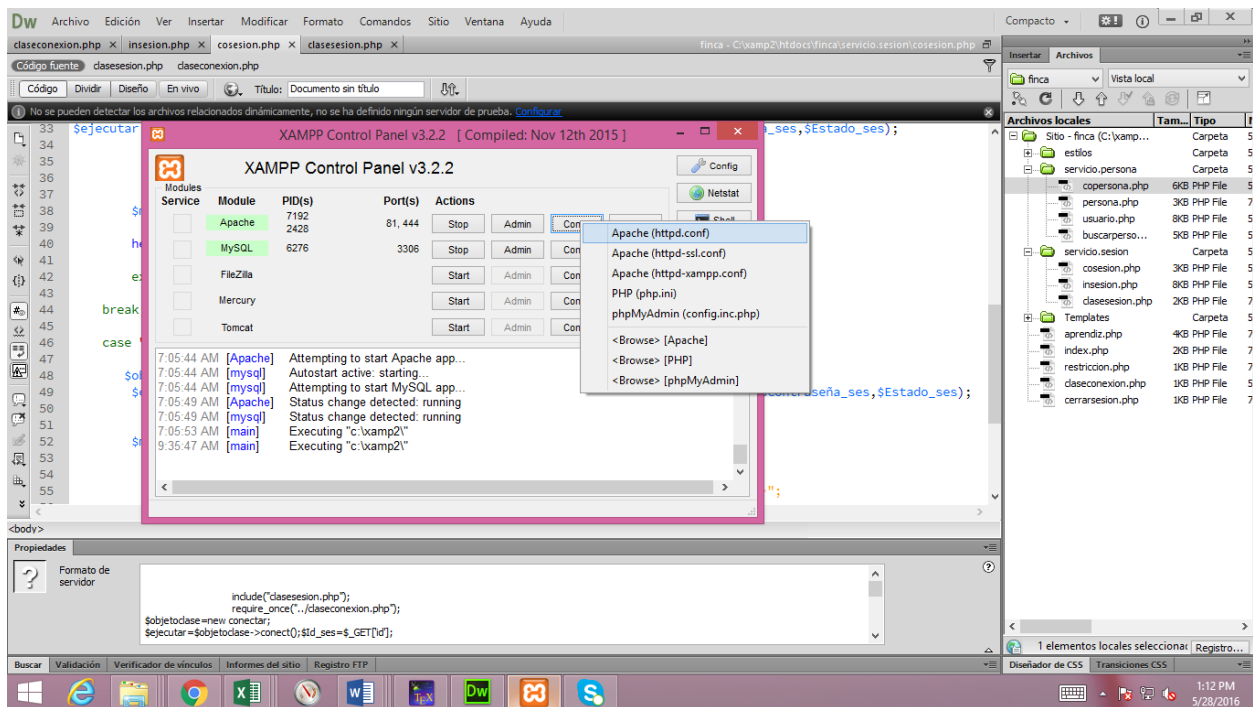
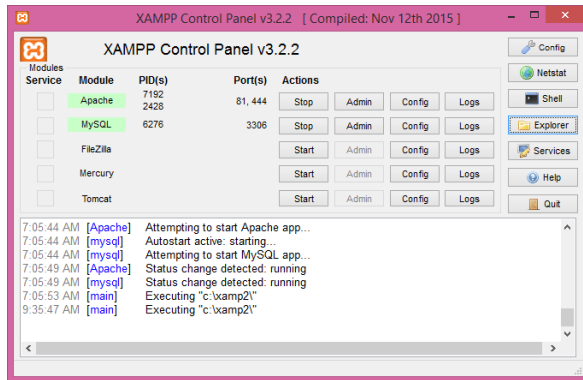
Primero se debe configurar la máquina virtual con una tarjeta tipo NAT y dirección ip estática:

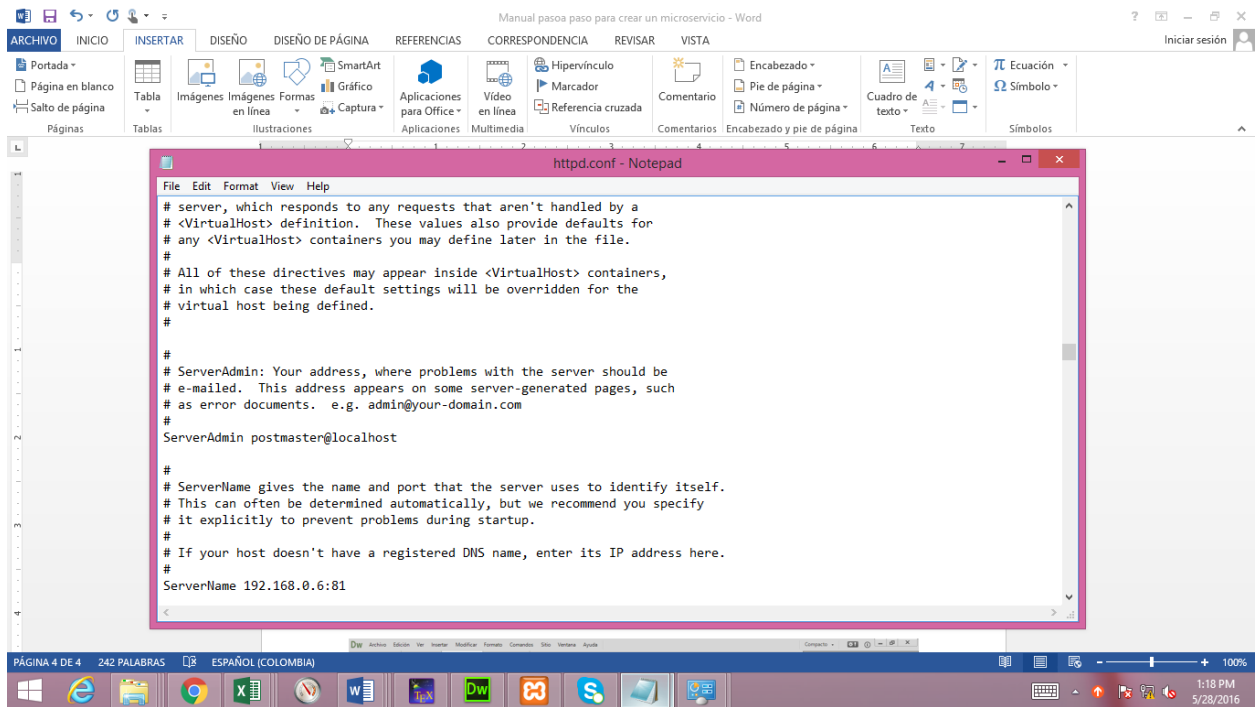
Configurar una ip fija:



Configuración servidor Apache

1. Se debe instalar un servidor apache, y los servicios deben estar corriendo, en este caso se utilizó xampp, es importante tener en cuenta que el puerto utilizado es el 81:



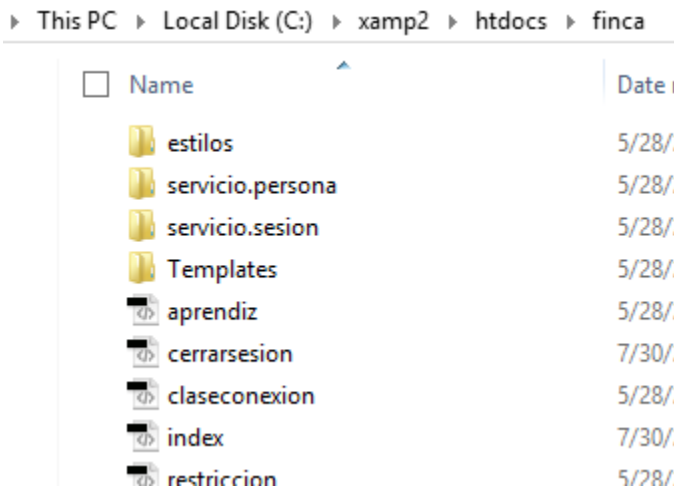


Configuración servicio en PHP

1. Se debe crear el archivo de conexión a la base de datos:

```
//Código para conectar a la base de datos
$conexion=mysql_connect("localhost","root","");// Cadena de conexión usuario y contraseña
mysql_query("SET NAMES 'utf8'");
mysql_select_db("finca");//nombre de la base de datos
return $conexion;
```

2. Organizar las carpetas en estructura jerárquica por cada servicio, es importante, la ubicación del proyecto debe estar en la carpeta htdocs del servidor apache.



3. Para la ejecución del microservicio, se necesita los 4 pasos siguientes:
 1. Script de conexión e interfaz

Cada servicio cuenta con el acceso a la base de datos por medio de REST ful: para configurar el envío ingresar al servicio y configurar el tipo : GET,POST...

```
//el formulario envía los datos por medio del método get
?>
<form action="cosecion.php" method="get" name="formulario" id="formulario" >
    <p>&nbsp;</p>
    <p>&nbsp;</p>
    <table class="tabla" bgcolor="#fc7323">
        <tr>
            <td colspan="2">Ingresa Los Datos</td>
        </tr>
        <tr>
            <td>Consecutivo</td>
            <td><input type="number" id="id" name="id" placeholder="No Ingresar" readonl
$_GET['id'];?>" />
            <input type="hidden" id="id" name="id" value="<?php echo @$_GET['id'];?>" />
        </tr>
```

2. Agregar Solicitud cliente Avance Rest :

Para este caso es GET , pero este dato puede cambiar según el tipo de solicitud: POST por ejemplo

3. Script en PHP para obtener información del usuario:

```
$Nombre_ses=$_GET['nombre'];  
$Numeroid_ses=$_GET['identi'];  
$Rol_ses=$_GET['rol'];  
$Contraseña_ses=$_GET['password1'];  
$Estado_ses=$_GET['permiso'];  
$enviar=$_GET['enviar'];
```


4. Script en PHP para obtener la condición de usuario

```
<form action="cosesion.php" method="get" name="formulario" id="formulario" >  
  .  
  .  
</form>
```

Creación de base de datos:

Por medio de phpmyadmin se crea la base de datos con una tabla usuarios cómo lo indica la figura, para poder conectar el servicio:

Bases de datos

Crear base de datos 

finca Cotejamiento ▼ Crear

finca persona

- Id_per : int(20)
- Nombre_per : varchar(100)
- Apellidos_per : varchar(50)
- Correo_per : varchar(100)
- Telefono_per : varchar(100)
- Direccion_per : varchar(100)
- Profesion_per : varchar(100)
- Rol_per : varchar(100)
- Cargo_per : varchar(100)
- Centroforma_per : varchar(100)

finca sesion

- Id_ses : int(20)
- Nombre_ses : varchar(100)
- Numeroid_ses : int(20)
- Rol_ses : varchar(100)
- Contraseña_ses : varchar(100)
- Estado_ses : varchar(10)

Configuración del Balanceador De Carga en PHP:

1. Se debe descargar el componente para php, en este caso ingresar a github al siguiente link: <https://github.com/andreiz/php-zookeeper> y descargar dicho componente
2. Copiar el api de php en la ruta del servidor apache configurado:

Local Disk (C:) > xamp2 > php > ext > zookeeper-0.2.2

<input type="checkbox"/> Name	Date modified
examples	6/4/2016 12:38
ChangeLog	7/30/2013 12:1
config.m4	7/30/2013 12:1
CREDITS	7/30/2013 12:1
LICENSE	7/30/2013 12:1
php_zookeeper	7/30/2013 12:1
php_zookeeper.h	7/30/2013 12:1
php_zookeeper_private.h	7/30/2013 12:1
php_zookeeper_session	7/30/2013 12:1
php_zookeeper_session.h	7/30/2013 12:1
README.markdown	7/30/2013 12:1
zoo_lock	7/30/2013 12:1
zoo_lock.h	7/30/2013 12:1
<input checked="" type="checkbox"/> zookeeper-api	7/30/2013 12:1



zookeeper-api.php

PC > Local Disk (C:) > xamp2 > htdocs > finca

<input type="checkbox"/> Name	Date modified
estilos	5/28/2013
servicio.persona	5/28/2013
servicio.sesion	5/28/2013
Templates	5/28/2013
cerrarsesion	7/30/2013
claseconexion	5/28/2013
index	5/28/2013
restriccion	5/28/2013
<input checked="" type="checkbox"/> zookeeper-api	7/30/2013

Para cada servicio creado en la página principal cargar se debe crear el archivo de configuración denominado myid:

PC > Local Disk (C:) > zk > zookeeper > datos

<input type="checkbox"/> Name	
<input checked="" type="checkbox"/> myid	<div>File Edit Format View Help 1,192.168.0.6:8889 2,192.168.0.7:8888 3,192.168.0.8:8887</div>

En cada vista principal de los servicios se deben crear los nodos:



```
139
140
141 $zk = new serviciofinca('localhost:2181');
142 var_dump($zk->get('/'));
143 var_dump($zk->getChildren('/'));
144 var_dump($zk->set('/servicio.persona', 'nodohijoejempl
145 var_dump($zk->get('/servicio.persona'));
146 var_dump($zk->getChildren('/'));
147
148 var_dump($zk->set('/servicio/sesion'));
149 var_dump($zk->set('/servicio/sesion'));
150 var_dump($zk->getChildren('/servicio'));
151
```

Prueba de uso y transferencia las otras maquinas

Lo primero que se debe hacer es clonar las maquinas, e cuanto al servidor apache xampp, pues el framework zeekeer solo debe estar en una maquina principal, cada vez que se accede a una url de la vista tanto de servicios.persona cómo de servicios.sesion debe hacer una carga al servidor en consola y en una aplicación grande debe reducir el tiempo de ejecución.