

Anotaciones generales sobre el despliegue

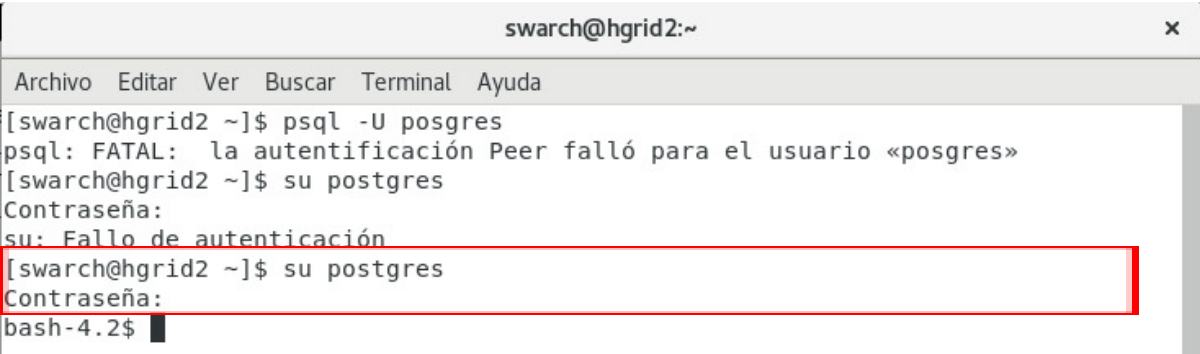
Una **aclaración importante**, a lo largo de este documento se habla de los hostnames de los computadores donde se despliegan los diferentes componentes. Sin embargo, si se encuentra realizando el despliegue desde una red externa a la de la Universidad Icesi, pero cuenta con acceso a la red del laboratorio, debe preceder el hostname con una x, es decir, dentro de la red de la Universidad, hgrid2, y por fuera de la red de la universidad xhgrid2.

Por otro lado, si el despliegue se realiza de manera remota deberá contar con Xming[1], ZeroTier[2], y Putty[3] en su computadora, además de tener acceso a la red del laboratorio y la configuración de hostnames que esto implica. Además, se da por hecho que el proyecto ha sido compilado utilizando el comando **gradle build** previamente.

INSTRUCCIONES DE DESPLIEGUE:

1. DESPLIEGUE DE BASE DE DATOS:

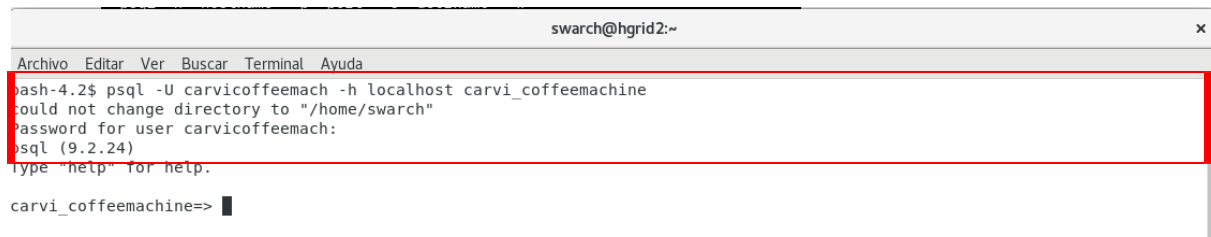
Para realizar el despliegue de la base de datos se utilizó el computador **hgrid2**. Primero, es necesario acceder al **usuario postgres** haciendo uso del comando **su postgres**, la contraseña que deberá ingresar es **postgres**, como se puede ver en la siguiente imagen.



```
swarch@hgrid2:~  
Archivo Editar Ver Buscar Terminal Ayuda  
[swarch@hgrid2 ~]$ psql -U postgres  
psql: FATAL: la autenticación Peer falló para el usuario «postgres»  
[swarch@hgrid2 ~]$ su postgres  
Contraseña:  
su: Fallo de autenticación  
[swarch@hgrid2 ~]$ su postgres  
Contraseña:  
bash-4.2$
```

Después de iniciar sesión con el usuario postgres, puede acceder a la consola bash utilizando el comando **psql postgres**. Una vez allí, utilice el script sql ubicado en el proyecto, en la dirección **./scripts/postgres**, para crear un nuevo usuario y la correspondiente base de datos asociada. Además, encontrará otros scripts en esta ubicación para la creación de las tablas del modelo.

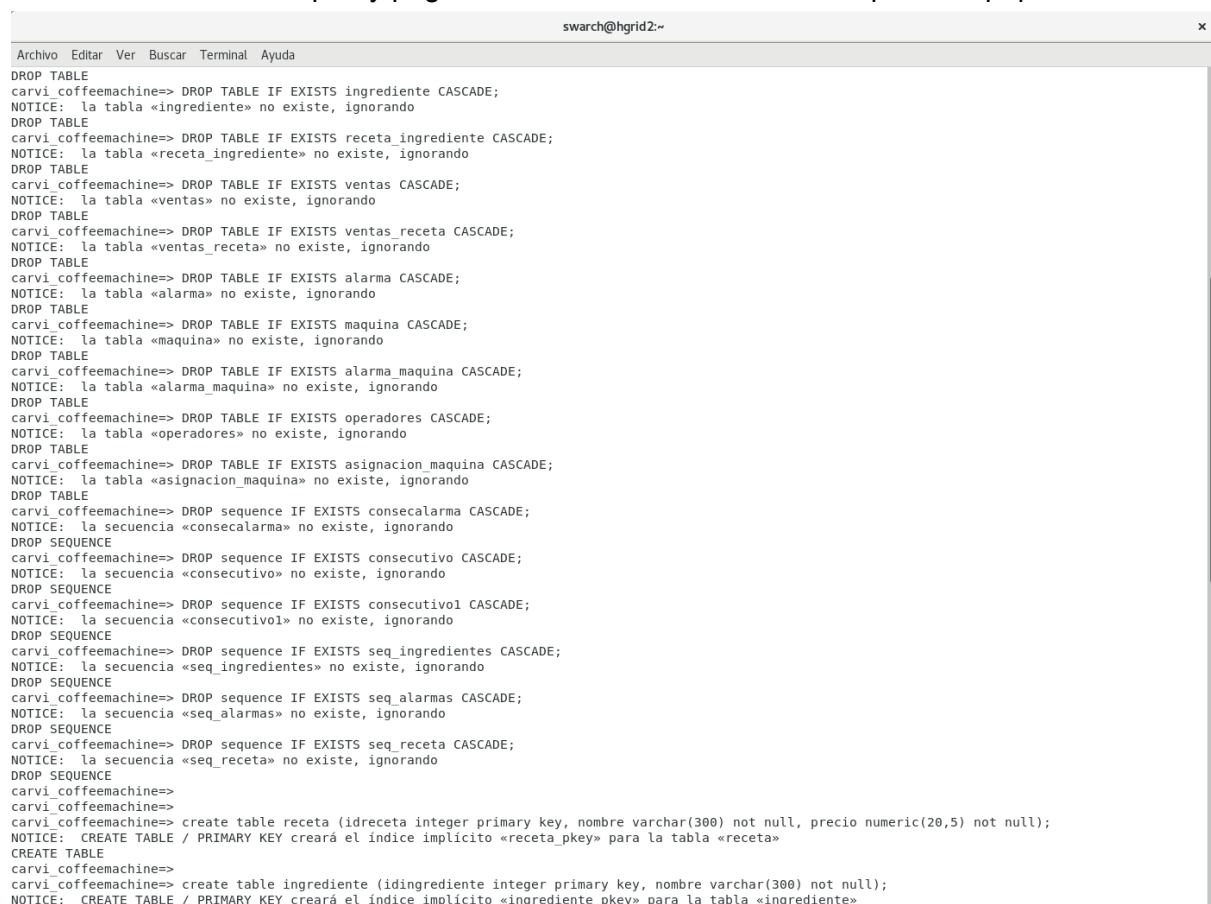
Ahora, salga del usuario postgres y pase al usuario de base de datos que acaba de crear. En nuestro caso el comando para realizar esto luce así **psql -U postgres -h localhost carvi_coffeemachine**. En la siguiente captura de pantalla puede ver este proceso:

A terminal window titled 'swarch@hgrid2:~' with a menu bar (Archivo, Editar, Ver, Buscar, Terminal, Ayuda). The terminal shows the command 'bash-4.2\$ psql -U carvicoffeemach -h localhost carvi_coffeemachine' being executed. The output indicates the user has switched to 'carvi_coffeemach' and shows the psql prompt 'psql (9.2.24)' and a message 'type "help" for help.' The prompt 'carvi_coffeemachine=>' is visible at the bottom.

```
swarch@hgrid2:~
Archivo Editar Ver Buscar Terminal Ayuda
bash-4.2$ psql -U carvicoffeemach -h localhost carvi_coffeemachine
could not change directory to "/home/swarch"
Password for user carvicoffeemach:
psql (9.2.24)
type "help" for help.

carvi_coffeemachine=>
```

Una vez accedemos con las credenciales que configuramos previamente, en la consola bash de nuestro usuario procedemos con la creación de las tablas como se muestra a continuación. Puede copiar y pegar directamente las sentencias sql del script provisto.

A terminal window titled 'swarch@hgrid2:~' showing the execution of a series of SQL commands. The commands are grouped into sections: dropping tables (ingrediente, receta_ingredient, ventas, ventas_receta, alarma, maquina, alarma_maquina, operadores, asignacion_maquina), dropping sequences (consecalarma, consecutivo, consecutivol, seq_ingredientes, seq_alarmas, seq_receta), and creating tables (receta, ingrediente). Each command is followed by a 'NOTICE' message indicating the result of the operation.

```
swarch@hgrid2:~
Archivo Editar Ver Buscar Terminal Ayuda
DROP TABLE
carvi_coffeemachine=> DROP TABLE IF EXISTS ingrediente CASCADE;
NOTICE: la tabla «ingrediente» no existe, ignorando
DROP TABLE
carvi_coffeemachine=> DROP TABLE IF EXISTS receta_ingredient CASCADE;
NOTICE: la tabla «receta_ingredient» no existe, ignorando
DROP TABLE
carvi_coffeemachine=> DROP TABLE IF EXISTS ventas CASCADE;
NOTICE: la tabla «ventas» no existe, ignorando
DROP TABLE
carvi_coffeemachine=> DROP TABLE IF EXISTS ventas_receta CASCADE;
NOTICE: la tabla «ventas_receta» no existe, ignorando
DROP TABLE
carvi_coffeemachine=> DROP TABLE IF EXISTS alarma CASCADE;
NOTICE: la tabla «alarma» no existe, ignorando
DROP TABLE
carvi_coffeemachine=> DROP TABLE IF EXISTS maquina CASCADE;
NOTICE: la tabla «maquina» no existe, ignorando
DROP TABLE
carvi_coffeemachine=> DROP TABLE IF EXISTS alarma_maquina CASCADE;
NOTICE: la tabla «alarma_maquina» no existe, ignorando
DROP TABLE
carvi_coffeemachine=> DROP TABLE IF EXISTS operadores CASCADE;
NOTICE: la tabla «operadores» no existe, ignorando
DROP TABLE
carvi_coffeemachine=> DROP TABLE IF EXISTS asignacion_maquina CASCADE;
NOTICE: la tabla «asignacion_maquina» no existe, ignorando
DROP TABLE
carvi_coffeemachine=> DROP sequence IF EXISTS consecalarma CASCADE;
NOTICE: la secuencia «consecalarma» no existe, ignorando
DROP SEQUENCE
carvi_coffeemachine=> DROP sequence IF EXISTS consecutivo CASCADE;
NOTICE: la secuencia «consecutivo» no existe, ignorando
DROP SEQUENCE
carvi_coffeemachine=> DROP sequence IF EXISTS consecutivol CASCADE;
NOTICE: la secuencia «consecutivol» no existe, ignorando
DROP SEQUENCE
carvi_coffeemachine=> DROP sequence IF EXISTS seq_ingredientes CASCADE;
NOTICE: la secuencia «seq_ingredientes» no existe, ignorando
DROP SEQUENCE
carvi_coffeemachine=> DROP sequence IF EXISTS seq_alarmas CASCADE;
NOTICE: la secuencia «seq_alarmas» no existe, ignorando
DROP SEQUENCE
carvi_coffeemachine=> DROP sequence IF EXISTS seq_receta CASCADE;
NOTICE: la secuencia «seq_receta» no existe, ignorando
DROP SEQUENCE
carvi_coffeemachine=>
carvi_coffeemachine=>
carvi_coffeemachine=> create table receta (idreceta integer primary key, nombre varchar(300) not null, precio numeric(20,5) not null);
NOTICE: CREATE TABLE / PRIMARY KEY crear  el  ndice impl cito «receta_pkey» para la tabla «receta»
CREATE TABLE
carvi_coffeemachine=>
carvi_coffeemachine=> create table ingrediente (idingrediente integer primary key, nombre varchar(300) not null);
NOTICE: CREATE TABLE / PRIMARY KEY crear  el  ndice impl cito «ingrediente_pkey» para la tabla «ingrediente»
```

Posteriormente, validamos que las tablas fueron creadas utilizando el comando `\d`, el cual nos permite listar las tablas creadas en una base de datos.

```
swarch@hgrid2:~  
Archivo Editar Ver Buscar Terminal Ayuda  
bash-4.2$ psql -U carvicoffeemach -h localhost carvi_coffeemachine  
could not change directory to "/home/swarch"  
Password for user carvicoffeemach:  
psql (9.2.24)  
Type "help" for help.  
  
carvi_coffeemachine=> \d  
List of relations  
Schema | Name | Type | Owner  
-----+-----+-----+-----  
public | alarma | table | carvicoffeemach  
public | alarma_maquina | table | carvicoffeemach  
public | asignacion_maquina | table | carvicoffeemach  
public | consecalarma | sequence | carvicoffeemach  
public | consecutivo | sequence | carvicoffeemach  
public | consecutivo1 | sequence | carvicoffeemach  
public | ingrediente | table | carvicoffeemach  
public | maquina | table | carvicoffeemach  
public | operadores | table | carvicoffeemach  
public | receta | table | carvicoffeemach  
public | receta_ingrediente | table | carvicoffeemach  
public | seq_alarmas | sequence | carvicoffeemach  
public | seq_ingredientes | sequence | carvicoffeemach  
public | seq_receta | sequence | carvicoffeemach  
public | ventas | table | carvicoffeemach  
public | ventas_receta | table | carvicoffeemach  
(16 rows)  
  
carvi_coffeemachine=> █
```

Finalmente, insertamos los datos en las tablas creadas utilizando el script, copiando y pegando en el bash las sentencias sql. Este proceso se muestra en las siguientes capturas de pantalla.

```
swarch@hgrid2:~  
Archivo Editar Ver Buscar Terminal Ayuda  
bash-4.2$ psql -U carvicoffeemach -h localhost carvi_coffeemachine  
could not change directory to "/home/swarch"  
Password for user carvicoffeemach:  
psql (9.2.24)  
Type "help" for help.  
  
carvi_coffeemachine=> select * from ingrediente;  
idingrediente | nombre  
-----+-----  
1 | Agua  
2 | Cafe  
3 | Azucar  
4 | Vaso  
(4 rows)  
  
carvi_coffeemachine=> █
```

```

carvi_coffeemachine=> insert into receta_ingrediente (idreceta,idingrediente,unidades) values (1,1,100);
INSERT 0 1
carvi_coffeemachine=> insert into receta_ingrediente (idreceta,idingrediente,unidades) values (1,2,10);
INSERT 0 1
carvi_coffeemachine=> insert into receta_ingrediente (idreceta,idingrediente,unidades) values (1,3,10);
INSERT 0 1
carvi_coffeemachine=> insert into receta_ingrediente (idreceta,idingrediente,unidades) values (1,4,1);
INSERT 0 1
carvi_coffeemachine=> insert into receta_ingrediente (idreceta,idingrediente,unidades) values (2,1,100);
INSERT 0 1
carvi_coffeemachine=> insert into receta_ingrediente (idreceta,idingrediente,unidades) values (2,2,30);
INSERT 0 1
carvi_coffeemachine=> insert into receta_ingrediente (idreceta,idingrediente,unidades) values (2,3,10);
INSERT 0 1
carvi_coffeemachine=> insert into receta_ingrediente (idreceta,idingrediente,unidades) values (2,4,1);
INSERT 0 1
carvi_coffeemachine=> select * from receta_ingrediente;
 idreceta | idingrediente | unidades
-----+-----+-----
      1 |             1 |       100
      1 |             2 |        10
      1 |             3 |        10
      1 |             4 |         1
      2 |             1 |       100
      2 |             2 |        30
      2 |             3 |        10
      2 |             4 |         1
(8 rows)

carvi_coffeemachine=> █

```

```

carvi_coffeemachine=> select * from alarma;
 idalarma | nombre
-----+-----
      1 | alarma1
      2 | alarma2
      3 | alarma3
      4 | alarma4
      5 | alarma5
      6 | alarma6
      7 | alarma7
      8 | alarma8
      9 | alarma9
     10 | alarma10
     11 | alarma11
     12 | alarma12
(12 rows)

```

```

carvi_coffeemachine=> select * from alarma_maquina;
 id_alarma | id_maquina | fecha_inicial | fecha_final | consecutivo
-----+-----+-----+-----+-----
      1 |          1 | 2023-04-25 |             |          1
      2 |          5 | 2023-04-25 |             |          2
      3 |          1 | 2023-04-25 |             |          3
      4 |          3 | 2023-04-25 |             |          4
      1 |          1 | 2023-04-25 |             |          5
(5 rows)

carvi_coffeemachine=> █

```

```
carvi_coffeemachine=> select * from operadores;
idoperador | nombre | correo | contrasena
-----+-----+-----+-----
1 | Miguel Angel | test@gmail.com | 1123
2 | Donatello | test1@gmail.com | 2123
3 | Raffaello | test2@gmail.com | 3123
4 | Leonardo | test3@gmail.com | 4123
(4 rows)

carvi_coffeemachine=> █
```

```
carvi_coffeemachine=> select * from asignacion_maquina;
id_operador | id_maquina
-----+-----
1 | 1
1 | 2
1 | 3
2 | 4
2 | 5
(5 rows)

carvi_coffeemachine=> █
```

Una vez verificada la inserción de los datos en cada una de las tablas se da por terminado el despliegue de la base de datos. Esto lo puede realizar a través de una consulta sql (recuerde terminar la consulta con un punto y coma, de lo contrario no funcionará).

2. Despliegue del servidor central

Este debe ser el segundo componente que debe desplegar pues los que se despliegan en los pasos posteriores necesitan conectarse a este. Ahora bien, para el despliegue del servidor central se decidió usar el computador **xhgrid3** con el **puerto 9096** (se asume que se hizo una petición para la apertura de este puerto).

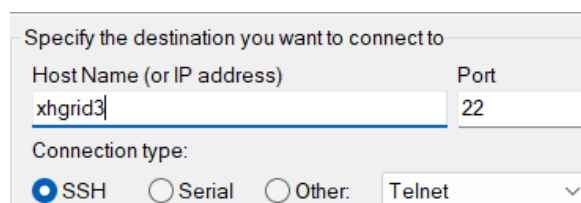
En este caso como el despliegue se continuó de manera remota, se envió al computador destino utilizando el comando **scp** (utilice **scp -h** para obtener detalles del funcionamiento del comando) el componente con el archivo de configuración modificador en el endpoint de la siguiente manera:

```
Server.Endpoints = tcp -h xhgrid3 -p 9096
```

También es importante tener en cuenta la conexión a la base de datos que se encuentra en el computador **xhgrid2**:

```
ConexionBD = jdbc:postgresql://hgrid2:5432/carvi_coffeemachine
usuarioBD = carvicoffeemach
passwordBD = carvi
```

Una vez realizadas las configuraciones, se debe usar la herramienta putty junto con xming para permitir la ejecución del servidor central, esto debido a que el servidor central cuenta con una interfaz gráfica que debe ejecutarse en el computador que se conectó a través de ssh. Para hacer esto, se ingresa a putty indicando el computador y el tipo de conexión. Además, se debe **habilitar el “x11 forwarding”** de la siguiente manera:

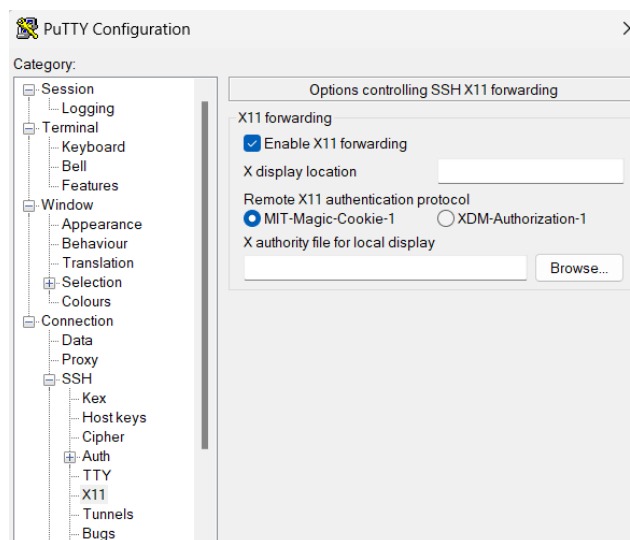


Specify the destination you want to connect to

Host Name (or IP address)	Port
xhgrid3	22

Connection type:

☒ SSH ☐ Serial ☐ Other: Telnet



PuTTY Configuration

Category:

- Session
- Logging
- Terminal
- Keyboard
- Bell
- Features
- Window
 - Appearance
 - Behaviour
 - Translation
 - Selection
 - Colours
- Connection
 - Data
 - Proxy
 - SSH
 - Kex
 - Host keys
 - Cipher
 - Auth
 - TTY
 - X11
 - Tunnels
 - Bugs

Options controlling SSH X11 forwarding

X11 forwarding

☒ Enable X11 forwarding

X display location

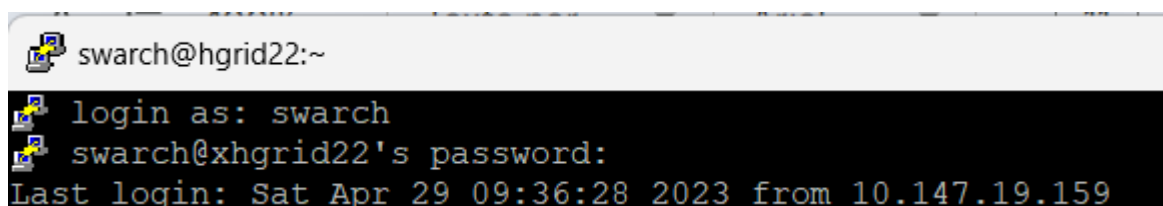
Remote X11 authentication protocol

☒ MIT-Magic-Cookie-1 ☐ XDM-Authorization-1

X authority file for local display

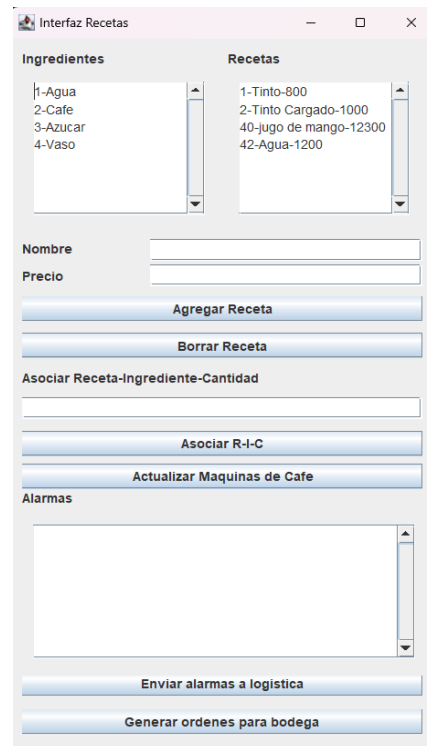
Browse...

El proceso anterior abrirá una consola en la cual debe ingresar con el usuario **swarch** y la contraseña **swarch**.



```
swarch@hgrid22:~
login as: swarch
swarch@xhgrid22's password:
Last login: Sat Apr 29 09:36:28 2023 from 10.147.19.159
```

Una vez aquí localice la carpeta donde se encuentra el proyecto, en nuestro caso en la dirección **./Documentos/GrupoCarvi**, ahora deberá ubicar el componente del servidor central, utilice el siguiente comando para acceder rápidamente **cd Coffee_Machine/coffeemach/ServidorCentral/**. Una vez en el directorio del Servidor Central utilice el comando **java -jar build/libs/ServidorCentral.jar**. Lo anterior deberá iniciar la siguiente interfaz del servidor central.



3. Despliegue del proxy-cache:

En el proyecto diríjase al directorio **./ProxyCache** y ejecute el .jar utilizando el siguiente comando **java -jar .\build\libs\ProxyCache.jar**. Este es el componente que nos ayudará a distribuir las actualizaciones de recetas a todas las máquinas que se encuentren desplegadas. Puede iniciar más instancias de proxy caché si es necesario, pero recuerde que deberá modificar las configuraciones de las máquinas de café para las que quiera gestionar utilizando el nuevo Proxy Caché. En nuestro caso utilizamos el **hgrid4** como proxy cache.

Una vez realizados los pasos anteriores en la consola verá el mensaje “Proxy Cache is running...”, esto le confirmará que no hubo errores al desplegar el proxy.

4. Despliegue del área de logística

Para realizar el despliegue del área de logística en el proyecto diríjase a **./cmLogistics**, aquí ejecute el .jar correspondiente utilizando el siguiente comando **java -jar .\build\libs\cmLogistics.jar**. En nuestro caso utilizamos **hgrid5** para el área de logística.

Una vez realizados los pasos anteriores en la consola verá el mensaje “Logistic is running..”, esto le confirmará que no hubo errores al desplegar el área de logística.

5. Despliegue de la bodega

Ahora, en el caso de la bodega diríjase en el proyecto a **./bodegaCentral**. Ahora, utilizando el siguiente comando ejecute el .jar generado **java -jar .\build\libs\bodegaCentral.jar**. En nuestro caso utilizamos **hgrid6** para el despliegue de la bodega.

Una vez realizados los pasos anteriores en la consola verá el mensaje “Warehouse is running...”, esto le confirmará que no hubo errores al desplegar la bodega.

5. Despliegue del componente reliable message

Se espera que cada máquina de café cuente con un componente de reliable message desplegado en la misma computadora. Para llevar a cabo lo anterior, en el proyecto diríjase al directorio **./reliableMessage** una vez aquí ejecute el comando **java -jar .\build\libs\reliableMessage.jar**, un mensaje se mostrará en consola confirmando que el componente se desplegó correctamente.

7. Despliegue de las máquinas de café

Para el despliegue de las máquinas de café se usaron los computadores **hgrid7** a **hgri21**. Instanciamos 2 máquinas de café por computadora con su respectivo Reliable Message para cada una.

Con respecto a los puertos, el componente reliableMessage debe tener un puerto distinto al de la máquina de café pues se encuentran desplegados en el mismo dispositivo. Por lo anterior, el hostname de la máquina de café y el reliableMessage deben ser los mismos. Un ejemplo se muestra en la siguiente imagen.

```
ProxyRM.Endpoints = tcp -h hgrid7 -p 9096  
  
relSer = ReliableMessage:tcp -h hgrid3 -p 9096
```

```
CoffeMach.Endpoints=default -h hgrid7 -p 9097  
  
alarmas = Alarmas:tcp -h hgrid3 -p 9096  
ventas = Ventas:tcp -h hgrid3 -p 9096  
recetas = Recetas:tcp -h hgrid3 -p 9096  
proxycache = ProxyCache:tcp -h hgrid4 -p 9095  
rm = RM:tcp -h hgrid7 -p 9096
```

Para realizar el despliegue remoto de una máquina de café, nuevamente debe usar putty junto con xming para poder desplegar la interfaz gráfica. Un ejemplo se muestra a continuación.

The first screenshot shows the PuTTY configuration window with the Host Name (or IP address) set to 'xhgrid17' and the Port set to '22'. The Connection type is set to 'SSH'.

The second screenshot shows the PuTTY configuration window with the Host Name (or IP address) set to 'xhgrid22' and the Port set to '22'. The Connection type is set to 'SSH'.

The third screenshot shows the PuTTY Configuration dialog box. The Category is set to 'SSH'. The X11 forwarding options are visible, including 'Enable X11 forwarding' (checked), 'X display location' (empty), 'Remote X11 authentication protocol' (MIT-Magic-Cookie-1 selected), and 'X authority file for local display' (empty).

Si todo fue realizado correctamente se observará lo siguiente:

The screenshot shows the Maquina de Cafe application interface. It has a title bar 'Maquina de Cafe' and a menu bar with 'Mantenimie...', 'Reporte Ventas', and 'Actualizar Rec'. The main window is divided into several sections:

- Monedas:** Includes buttons for 'Ingresar ...' (three times) and 'Devolver'. A text field shows '0'.
- Informacion:** Includes a section for 'Eventos' with a text field.
- Productos:** Includes a dropdown menu with 'Tinto' selected, a 'Verificar Precio' button, and an 'Ordenar' button.
- Insumos Disponibles:** Lists items and their prices: Azucar: 600.0, Cafe: 600.0, Agua: 6000.0, Vaso: 200.0, Deposito 100: 12, Deposito 200: 12, Deposito 500: 12.
- Recetas Disponibles:** Lists recipes: Tinto, Tinto Cargado.