



MeetGrid

Manual de instalación y despliegue



ÍNDICE

| | | |
|-----------|--|----------|
| 1. | Requerimientos previos a la instalación y despliegue..... | 3 |
| 2. | Instalación del sistema..... | 3 |
| 2.2. | Instalación contenedor de aplicaciones y servidor local..... | 3 |
| 2.1. | Instalación persistencia..... | 4 |
| 3. | Despliegue del sistema..... | 6 |
| 3.1. | Conexión con repositorio..... | 6 |
| 3.2. | Contenedores y servicios..... | 7 |
| 3.3. | Logs del sistema..... | 9 |
| 3.4. | Primera puesta en servicio..... | 9 |



1. Requerimientos previos a la instalación y despliegue

- Una máquina con capacidad de funcionar como servidor con una distribución de Linux o Windows instalada.
- Conexión a internet.
- Conocer los procesos de instalación de software en Windows o Linux, así como comandos básicos de la consola de éste último.

2. Instalación del sistema

Para el correcto funcionamiento de esta aplicación usaremos Tomcat y MySQL, sobre los cuales desplegaremos la aplicación compilada en formato .war e importaremos un dump con la lógica (y registros de prueba) de la base de datos en formato .sql, ambos cargados en GitHub y actualizados automáticamente por la máquina local del desarrollador.

Para este manual vamos a ver como instalarlas sobre una máquina Ubuntu.

Versiones usadas:

- Ubuntu 20.04 LTS
- Tomcat 9
- MySQL 8

2.1 Instalación contenedor de aplicaciones y servidor local

Instalaremos el paquete *Apache Tomcat 9* sobre *Ubuntu 20.04*, el cual incluye servidor local y el contenedor de aplicaciones *Catalina*.

Accedemos a la consola de comandos para poder ejecutar los siguientes pasos:

Actualizamos las listas de paquetes:

```
~$ sudo apt update
```

Instalaremos los paquetes principales, correspondientes al núcleo de Tomcat y a las aplicaciones administrativas:

```
~$ sudo apt install -y tomcat9 tomcat9-admin
```

Podemos ver el estado de nuestro servicio:

```
systemctl status tomcat
```

```
● tomcat.service - Apache Tomcat Web Application Container
   Loaded: loaded (/etc/systemd/system/tomcat.service; enabled; vendor preset: enabled)
   Active: active (running) since Thu 2020-11-26 21:23:00 CET; 42min ago
     Process: 568 ExecStart=/opt/tomcat/bin/startup.sh (code=exited, status=0/SUCCESS)
    Main PID: 611 (java)
      Tasks: 32 (limit: 2318)
     Memory: 297.7M
    CGroup: /system.slice/tomcat.service
            └─611 /usr/lib/jvm/java-1.11.0-openjdk-amd64/bin/java -Djava.util.logging.config.file=/opt/tomcat/conf/logging.properties -Djav

nov 26 21:22:59 usuario-Standard-PC-i440FX-PIIX-1996 systemd[1]: Starting Apache Tomcat Web Application Container...
nov 26 21:23:00 usuario-Standard-PC-i440FX-PIIX-1996 startup.sh[568]: Existing PID file found during start.
nov 26 21:23:00 usuario-Standard-PC-i440FX-PIIX-1996 systemd[1]: Started Apache Tomcat Web Application Container.
nov 26 21:23:00 usuario-Standard-PC-i440FX-PIIX-1996 startup.sh[568]: Removing/clearing stale PID file.
nov 26 21:23:00 usuario-Standard-PC-i440FX-PIIX-1996 startup.sh[568]: Tomcat started.
```

Para acceder a Tomcat desde la red crearemos una regla de acceso a su puerto desde la misma:

```
~$ sudo ufw allow 8080/tcp
```

Debemos crear un usuario para la aplicación Tomcat. Para ello accedemos al siguiente archivo:

```
~$ sudo nano /etc/tomcat9/tomcat-users.xml
```



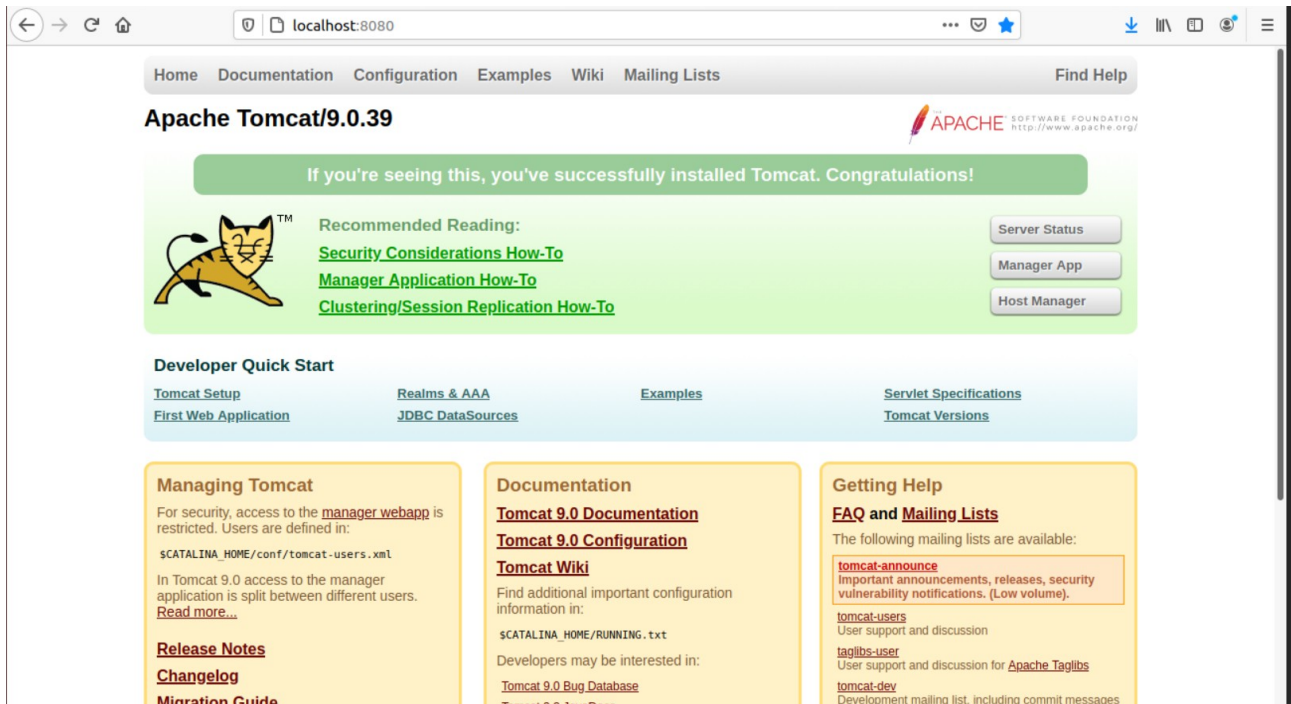
En él editaremos antes del cierre del bloque `tomcat-users`. Añadiremos una definición de usuario con contraseña y los roles necesarios:

```
<user username="usuario" password="XXXXXXXX" roles="admin-gui,manager-gui"/>
</tomcat-users>
```

Aplicamos los cambios en la aplicación reiniciándola:

```
~$ sudo systemctl restart tomcat9
```

Finalmente podemos acceder a nuestra aplicación Tomcat localmente indicando en cualquier navegador localhost:8080.



Para acceder a este manager-gui, nos pedirá el usuario y contraseña creado anteriormente.

Se recomienda además hacer que Tomcat inicie automáticamente al iniciar nuestra máquina:

```
/sbin/chkconfig tomcat on
```

2.2. Instalación persistencia

Para nuestra persistencia, haremos uso del motor de base de datos MySQL y gestor MySQL Workbench sobre Ubuntu 20.04

Actualiza el índice de paquetes en su servidor.

```
sudo apt update
```

Instala el paquete mysql-server:

```
sudo apt install mysql-server
```

A través del siguiente comando iniciaremos una secuencia guiada de instalación:

```
sudo mysql_secure_installation
```



Primero, le permitirá crear un usuario y contraseña para acceder a MySQL, según sus preferencias.

A continuación, por requisitos de la base de datos de MeetGrid **deberá indicar que la contraseña del usuario root sea también "root" (sin comillas).**

```
Please set the password for root here.
```

```
New password: root
```

```
Re-enter new password: root
```

Si tuviéramos alguna incidencia cumplimentando este paso, podemos modificarla posteriormente con los siguientes comandos:

```
sudo mysql
```

```
sql > ALTER USER 'root'@'localhost' IDENTIFIED WITH caching_sha2_password BY 'root';
```

```
sql > FLUSH PRIVILEGES;
```

Podemos verificar el estado del servicio con el siguiente comando:

```
systemctl status mysql.service
```

```
● mysql.service - MySQL Community Server
   Loaded: loaded (/lib/systemd/system/mysql.service; enabled; vendor preset: enabled)
   Active: active (running) since Thu 2020-11-26 21:23:04 CET; 1h 17min ago
     Process: 565 ExecStartPre=/usr/share/mysql/mysql-systemd-start pre (code=exited, status=0/SUCCESS)
    Main PID: 720 (mysqld)
      Status: "Server is operational"
        Tasks: 42 (limit: 2318)
       Memory: 134.7M
         CGroup: /system.slice/mysql.service
                └─720 /usr/sbin/mysqld

nov 26 21:22:59 usuario-Standard-PC-i440FX-PIIX-1996 systemd[1]: Starting MySQL Community Server...
nov 26 21:23:04 usuario-Standard-PC-i440FX-PIIX-1996 systemd[1]: Started MySQL Community Server.
```

A partir de aquí podemos arrancar el servicio con:

```
sudo systemctl start mysql
```

Ahora vamos a instalar MySQL Workbench para controlar y facilitar la gestión de nuestro dump de la base de datos de MeetGrid.

```
sudo apt install mysql-workbench-community
```

Ya la tendremos disponible para lanzar entre nuestras aplicaciones:





3. Despliegue del sistema

Sobre nuestro servidor, contenedor de aplicaciones y sistema de persistencia vamos a desplegar el código de la aplicación.

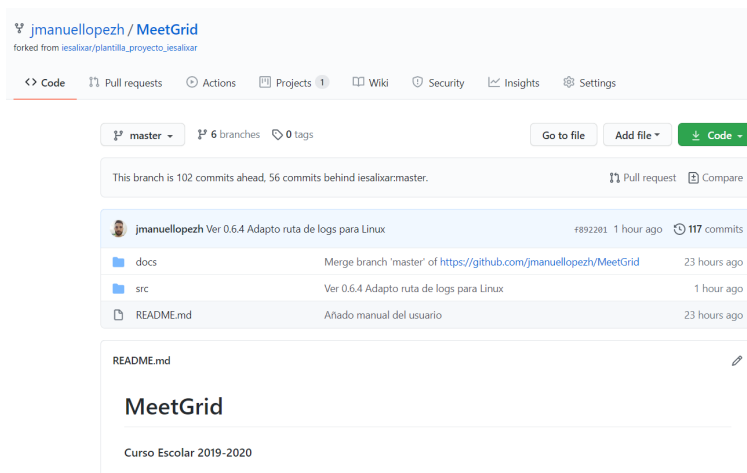
Para ello, primeramente debemos hacernos con los archivos que contienen este código.

3.1. Conexión con repositorio.

Necesitaremos sólo dos archivos del repositorio GitHub de la aplicación para hacerla funcionar.

Podemos acceder a la raíz del proyecto, para ver su contenido y documentación en:

<https://github.com/jmanuellopezh/MeetGrid>



Primero, descargaremos el archivo Java compilado en formato .war. Éste incluye la lógica y vistas de la aplicación:

<https://github.com/jmanuellopezh/MeetGrid/blob/master/src/Meetgrid.war>



Segundo, descargamos el archivo dump de la base de datos en formato .sql. Éste incluye las tablas, reglas y registros de prueba necesarios para que la aplicación funcione:

<https://github.com/jmanuellopezh/MeetGrid/blob/master/src/meetgrid-db.sql>



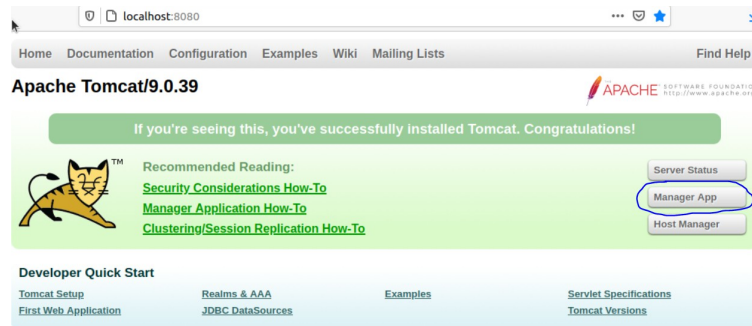


3.2. Contenedores y servicios

Una vez tengamos instalados nuestros servicios y descargados los archivos necesarios, solo hace falta integrarlos en ellos.

Primero, el código de la aplicación en sí: Meetgrid.war.

Entramos en localhost:8080 con nuestro usuario y clave, pulsamos sobre Manager App:



Buscamos la opción “Archivo war a desplegar”, pulsamos sobre “examinar...”, seleccionamos Meetgrid.war de nuestro directorio de descarga, y posteriormente “Desplegar”.

MeetGrid aparecerá entre las aplicaciones desplegadas:

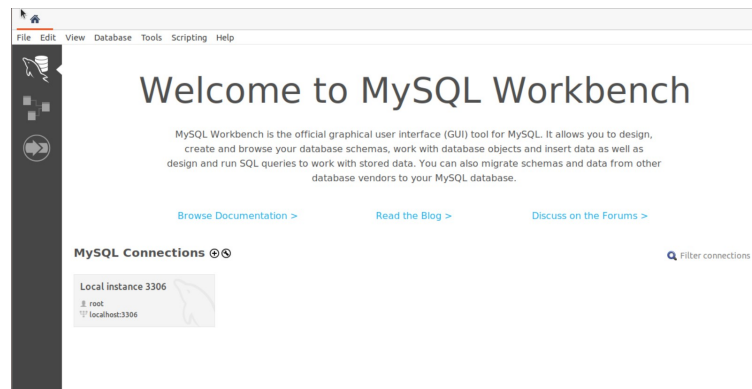
| Aplicaciones | | | | | |
|---------------|----------------------|---------------------------------|--------------|----------|--|
| Ruta | Versión | Nombre a Mostrar | Ejecutándose | Sesiones | Comandos |
| / | Ninguno especificado | Welcome to Tomcat | true | 0 | Arrancar Parar Recargar Replegar Expirar sesiones sin trabajar ≥ 30 minutos |
| /Meetgrid | Ninguno especificado | MeetGrid | true | 0 | Arrancar Parar Recargar Replegar Expirar sesiones sin trabajar ≥ 30 minutos |
| /docs | Ninguno especificado | Tomcat Documentation | true | 0 | Arrancar Parar Recargar Replegar Expirar sesiones sin trabajar ≥ 30 minutos |
| /examples | Ninguno especificado | Servlet and JSP Examples | true | 0 | Arrancar Parar Recargar Replegar Expirar sesiones sin trabajar ≥ 30 minutos |
| /host-manager | Ninguno especificado | Tomcat Host Manager Application | true | 0 | Arrancar Parar Recargar Replegar Expirar sesiones sin trabajar ≥ 30 minutos |
| /manager | Ninguno especificado | Tomcat Manager Application | true | 1 | Arrancar Parar Recargar Replegar Expirar sesiones sin trabajar ≥ 30 minutos |

Podemos pulsar sobre ella o escribir en nuestro navegador localhost:8080/Meetgrid para acceder a ella.

Pero la aplicación aún no funcionará, necesita de las tablas y registros de nuestra base de datos para ello.

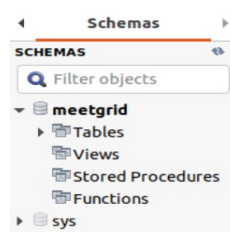


Tenemos que entrar en MySQL Workbench, entrar en la instancia local 3306 y poner nuestras credenciales ("root", "root").

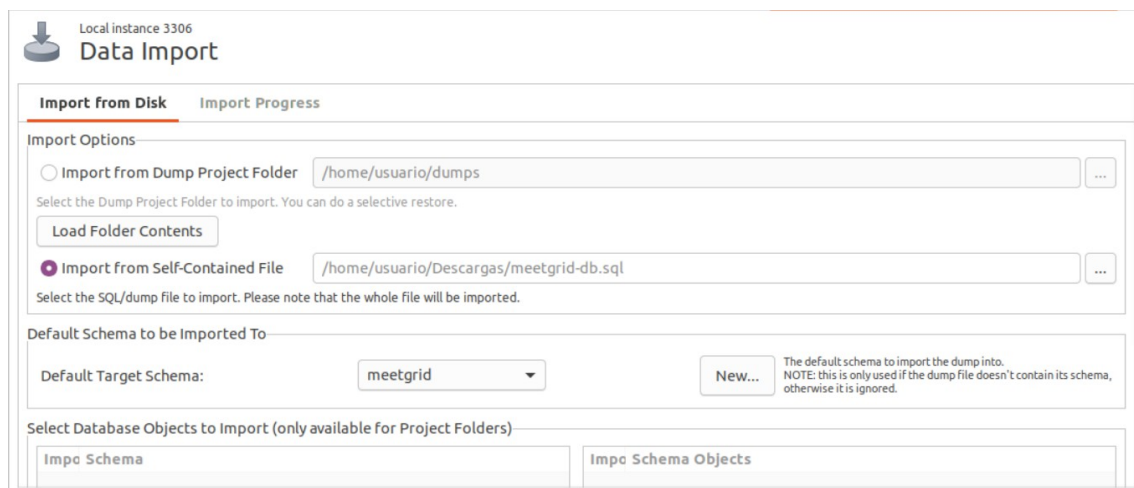


Ya dentro de la instancia, vamos a importar nuestra base de datos.

Creamos la base de datos que contendrá los registros, se ha de llamar "meetgrid" (sin comillas).



Pulsamos en Server > data import. Nos llevará a la pantalla de importación de datos en la que haremos que nuestro código SQL se vuelque en esta base de datos creando las tablas y registros necesarios.



En "Import from Self-Contained File" seleccionaremos nuestro archivo sql descargado del repositorio y en "Default Target Schema" la base de datos que lo contendrá ("meetgrid").



3.3. Logs del sistema

La aplicación incluye una completa lógica de creación, actualización y registro de actividades en la aplicación, todo ello de manera automática.

Para ello usa log4j (Log for Java).

Crearé un archivo en la ruta /home/usuario/logger.log y en él registrará todas las acciones pertinentes.

Importante: Para la correcta creación de este archivo, el usuario tomcat debe tener permisos de escritura en este directorio, así como una vez creado, el usuario “usuario” debe tener permisos de lectura para poder leer los registros que en él se almacenan.

Un extracto para mostrar el formato y tipo de información que estos registros muestran:

```
2020-11-25 22:34:08 INFO Login:53 - A user has logged in
2020-11-25 22:34:10 INFO Register:41 - 'Favorite' has been invoked!
2020-11-25 22:34:21 INFO Login:37 - 'ReadUsersByFilter' has been invoked!
2020-11-25 22:34:22 INFO Login:55 - Un usuario ha accedido al grid
2020-11-25 22:34:24 INFO Login:58 - Un usuario ha accedido al perfil del usuario prueba@meetgrid.com
2020-11-25 22:34:28 INFO Register:73 - El usuario id 4 hizo favorito al usuario 3
2020-11-25 22:34:28 INFO Register:41 - 'Favorite' has been invoked!
2020-11-25 22:34:29 INFO Login:58 - Un usuario ha accedido al perfil del usuario prueba@meetgrid.com
2020-11-25 22:34:30 INFO Login:42 - 'Messages' has been invoked!
```

3.4. Primera puesta en servicio

Al haber cumplimentado correctamente los pasos anteriores tendremos:

- La lógica que crea automáticamente el archivo de logs y sus registros.
- Una serie de registros necesarios para el correcto funcionamiento y pruebas añadidos automáticamente mediante la importación del dump de la base de datos.

Por lo tanto, no hay ningún paso necesario más a dar para la primera puesta en servicio, simplemente la instalación y despliegue indicado a lo largo de este documento.

Solo hay que entrar en localhost:8080/Meetgrid, será 100% operativa.