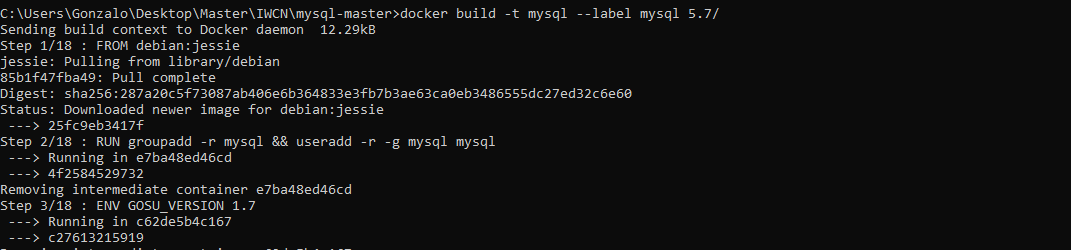
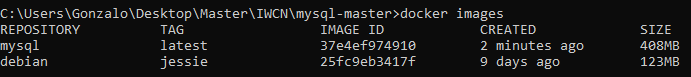
Descargamos el proyecto de docker a través de github en:

<https://github.com/docker-library/mysql.git>

Introducimos el proyecto en cualquier carpeta y accedemos a la terminal en modo administrador e instalamos mysql en la versión que queramos.



Podemos ver la imagen docker de la base de datos mysql



Creamos un contenedor a partir de la imagen.

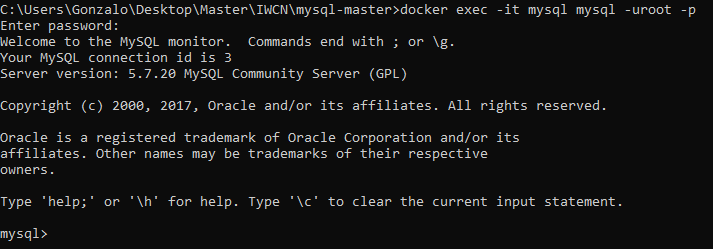
Aunque no se ve muy bien en la imagen se está especificando que la contraseña del root es aguilar, la base de datos se ha llamado prueba, un usuario de mysql tendrá como identidad gonzalo, y su contraseña será aguilar. Además, debemos especificar el puerto del docker y el puerto host que será el 3306.



Los contenedores que están funcionando actualmente los podemos visualizar a través del siguiente comando:

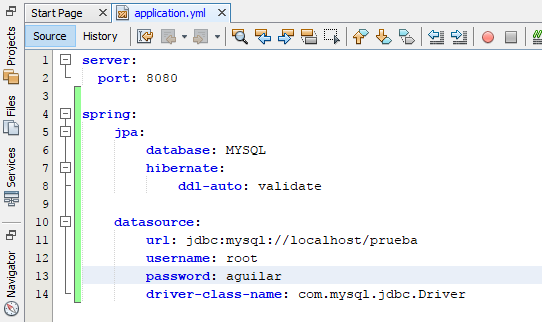


A continuación, se crea un cliente mysql dentro de docker. Nos pedirá la contraseña que hemos especificado anteriormente.



A continuación, vamos a modificar la base de datos de nuestro proyecto.

Accedemos tanto en el proyecto cliente como en el servidor en el fichero application.yml y deberá quedar de la siguiente forma:



La primera vez en el apartado ddl-auto debemos cambiar validate por create, consiguiendo de esta forma crear la tabla productos.

Una vez creada lo cambiaremos por validate.

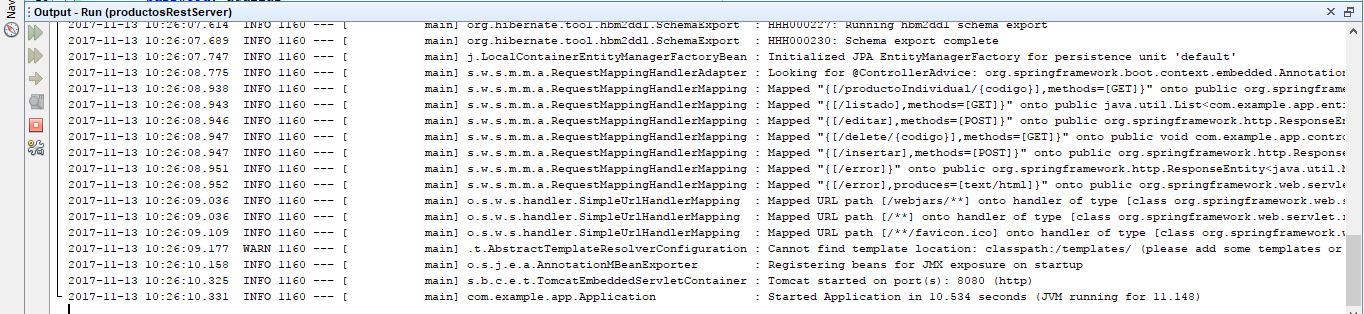
Podemos observar que se ha creado un icono similar al que representa una base de datos, en este caso especifica que tenemos ya la base de datos mysql.



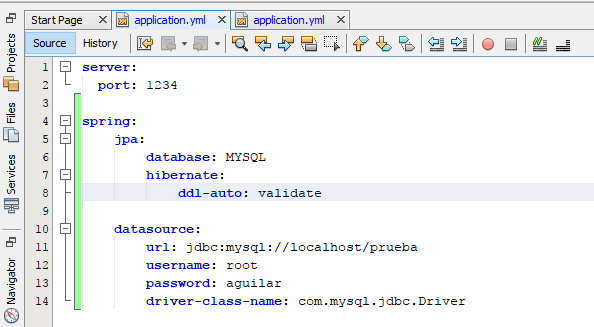
Modificamos las dependencias en el pom.xml, eliminando la dependencia de h2 e introduciendo la de mysql.

Pulsamos en el proyecto y le ordenamos realizar un clean y posteriormente Build with dependencies.

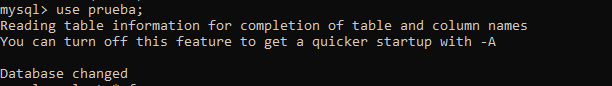
Lanzamos el proyecto del lado servidor para que se ejecute ddl-auto: create



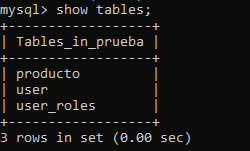
En el lado cliente, tendremos que dejar el puerto que teníamos en anteriores versiones (cualquiera diferente al puerto 8080) de lo contrario, colapsaría dicho puerto.



Seguidamente comprobamos que se hayan creado las tablas, para ello seleccionamos la base de datos.



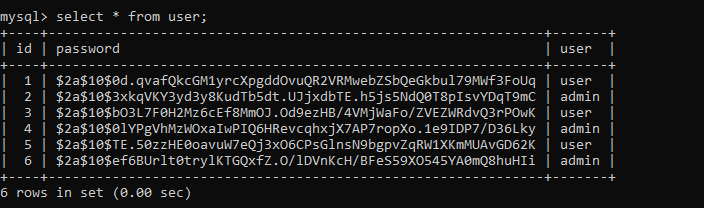
Mostramos las tablas:



Visualizamos la tabla producto y podemos comprobar que está vacia:

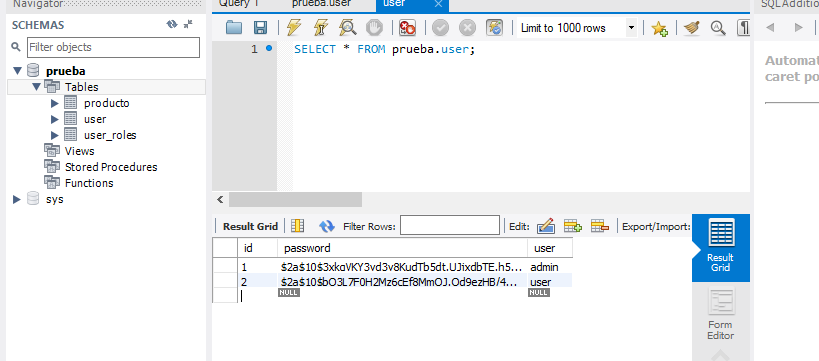


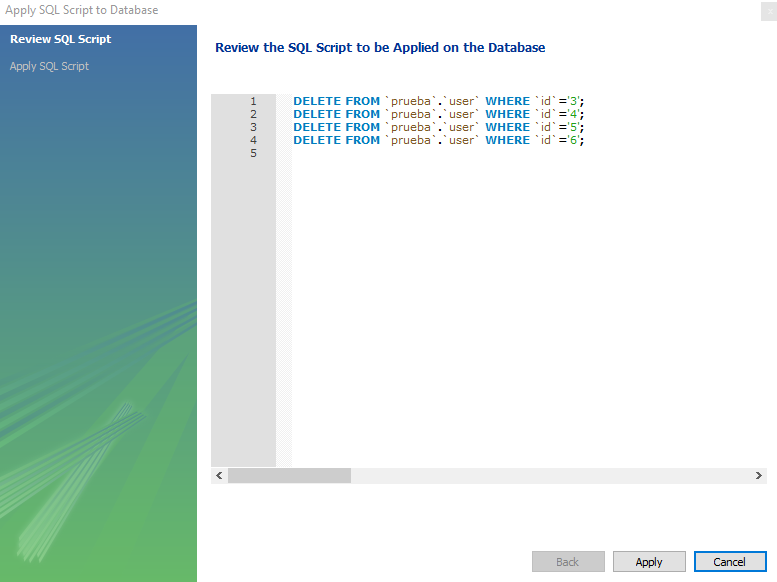
Visualizamos la tabla usuarios:



Según puedo ver, me ha creado 3 veces los mismos usuarios y contraseñas, debemos borrarlos, porque al realizar la consulta con la base de datos si busca por la columna user, coinciden varios resultados.

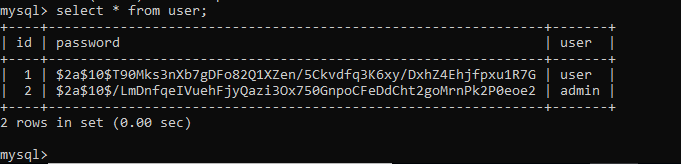
Para no tener que introducir comandos, existe una interfaz gráfica llamada Mysql Workbench con la que podremos realizar las modificaciones pertinentes en una base de datos.



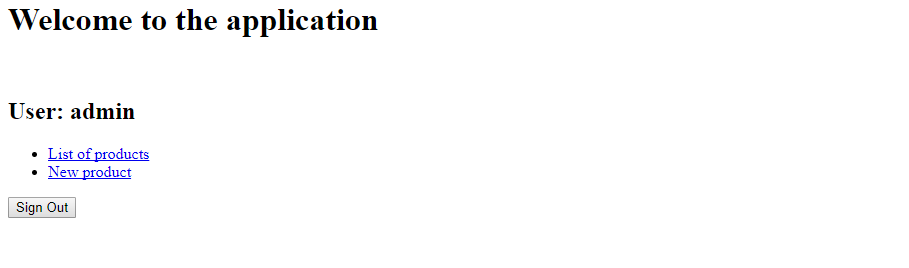
Automáticamente genera las sentencias necesarias de las modificaciones que hemos indicado:

Muestra errores de clave foránea (supongo que será por el id).

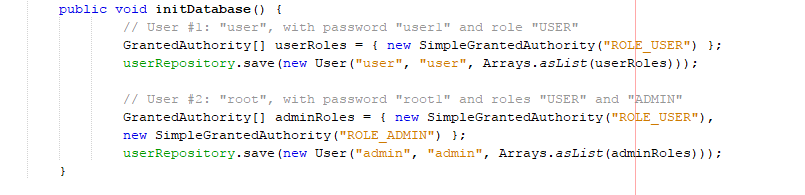
Solución, creo que he ejecutado tres veces la aplicación con el ddl-auto: create, por ello que se hayan triplicado las inserciones, por lo tanto, he eliminado las tablas y lo vuelvo a inicializar desde cero:



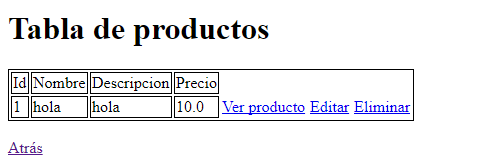
Equivocarse es de sabios… ;)



Una pequeña anotación y no por ello menos importante, según tengo mi código aunque este especificado el ddl-auto: validate, tenemos una función llamada initDatabase, que en este caso es la que me está creado siempre las mismas inserciones de los usuarios, SOLUCIÓN: comentar el código interno de esta función una vez realizado el create.



Insertamos un producto a través de la aplicación. Comprobamos en nuestra aplicación que se ha insertado correctamente.



También, para asegurarnos, lo comprobamos en la terminal de mysql.

