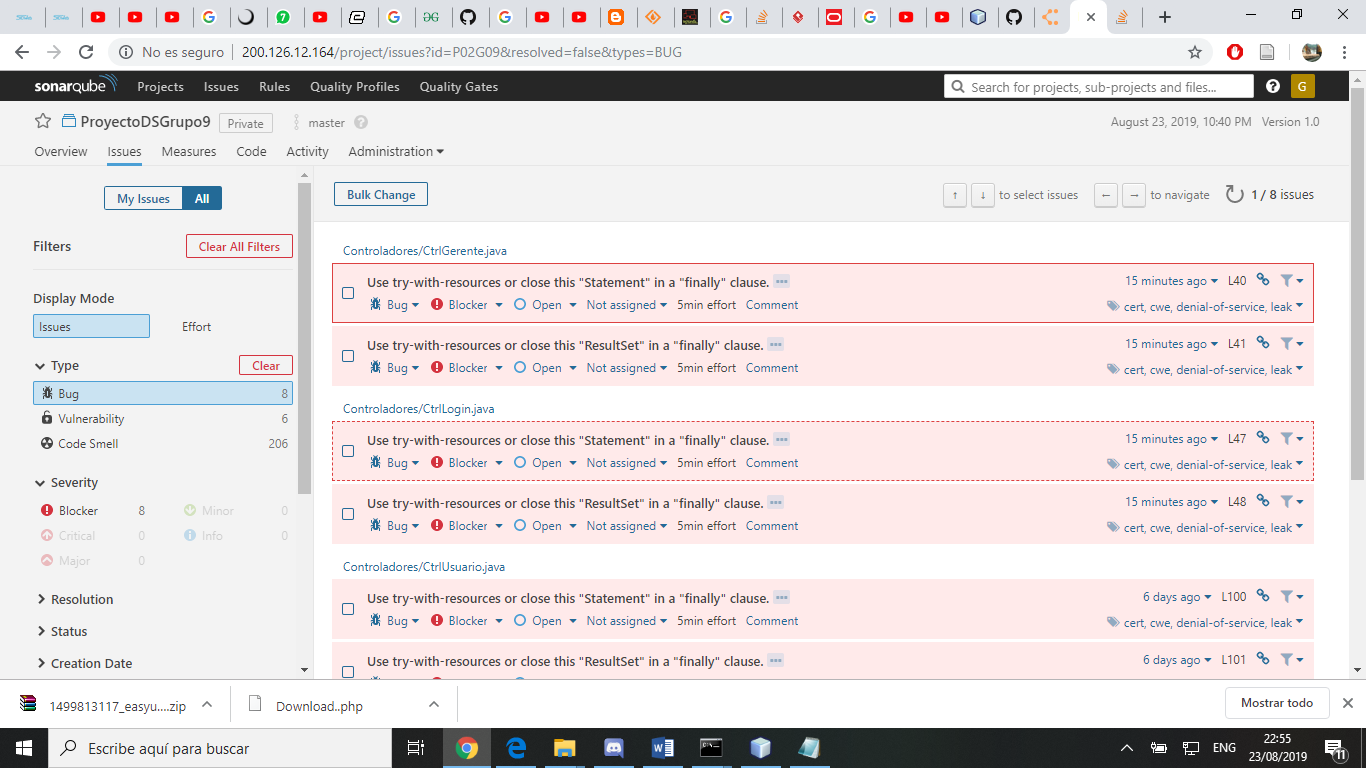
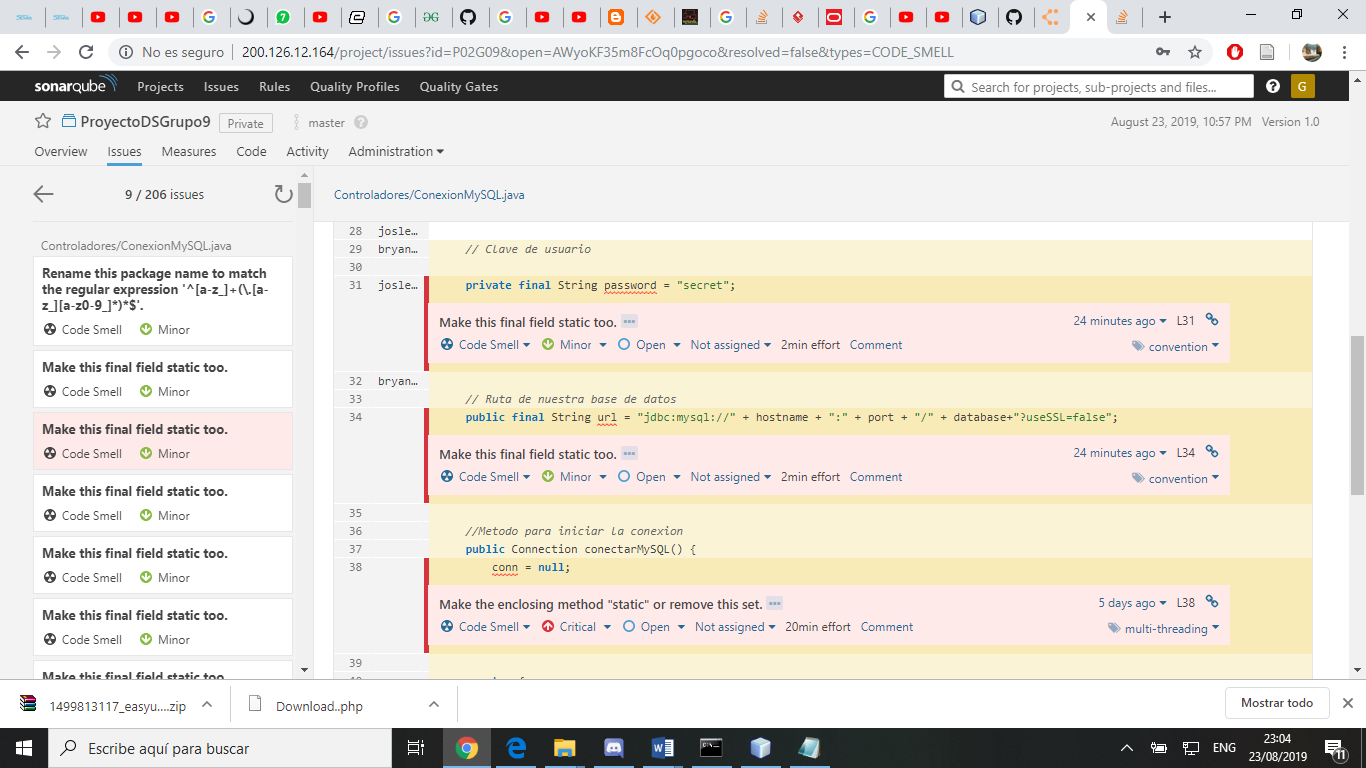
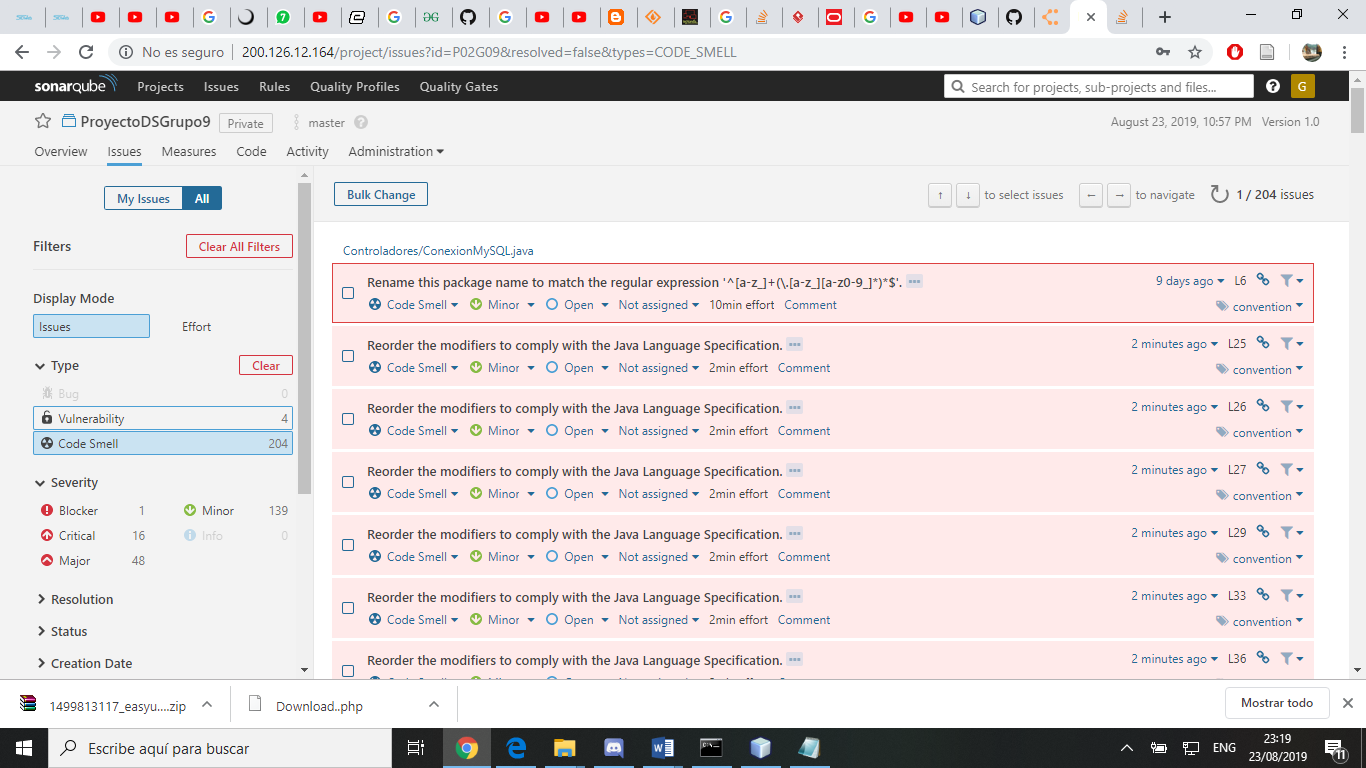
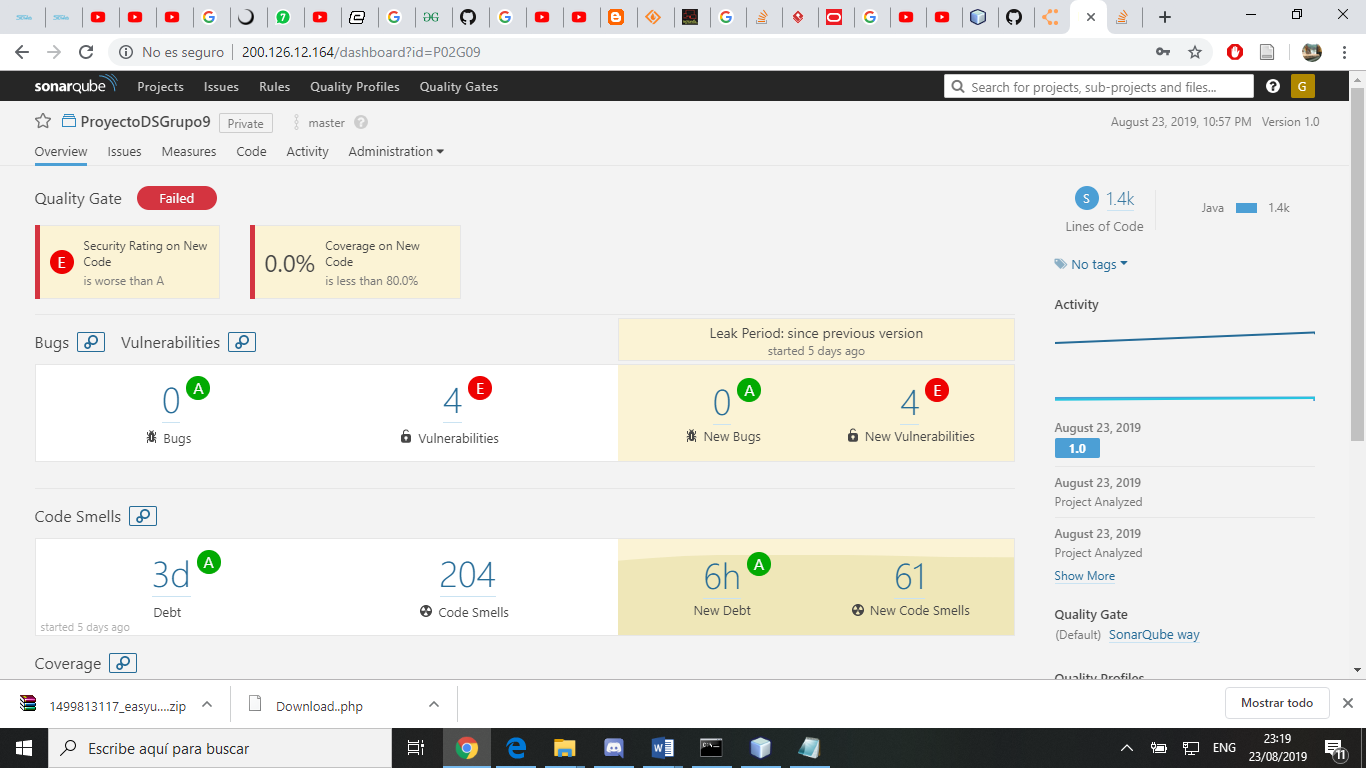
**REPORTE DISEÑO DE SOFTWARE**

**Refactoring**









**Uso de docker**

Primero nos bajamos la imagen de MySQL con el comando.

**docker pull mysql**

Después creamos un volumen para mantener los cambios hechos en la base de datos que se almacenara en Docker.

**docker volume create mysql-db-data**

En este comando mysql-db-data será el nombre del volumen creado

Después es necesario levantar el contenedor para ya poder empezar a usarlo

**docker run -d -p 33060:3306 --name mysql-db -e MYSQL\_ROOT\_PASSWORD=secret --mount src=mysql-db-data,dst=/var/lib/mysql mysql**

En este comando estamos seteando el Puerto que vamos a usar para la conexión, que por defecto mysql maneja el Puerto 3306 y lo cambiamos por 33060.

Además, con MYSQL\_ROOT\_PASSWORD=secret seteamos la contraseña para poder usar la base de datos

Además con --mount src=mysql-db-data,dst=/var/lib/mysql mysql, hacemos que los cambios realizados se guarden en el volumen que creamos, asi tendremos una consistencia de datos.

De esta manera ya tendremos configurado el contenedor de docker y listo para usarse

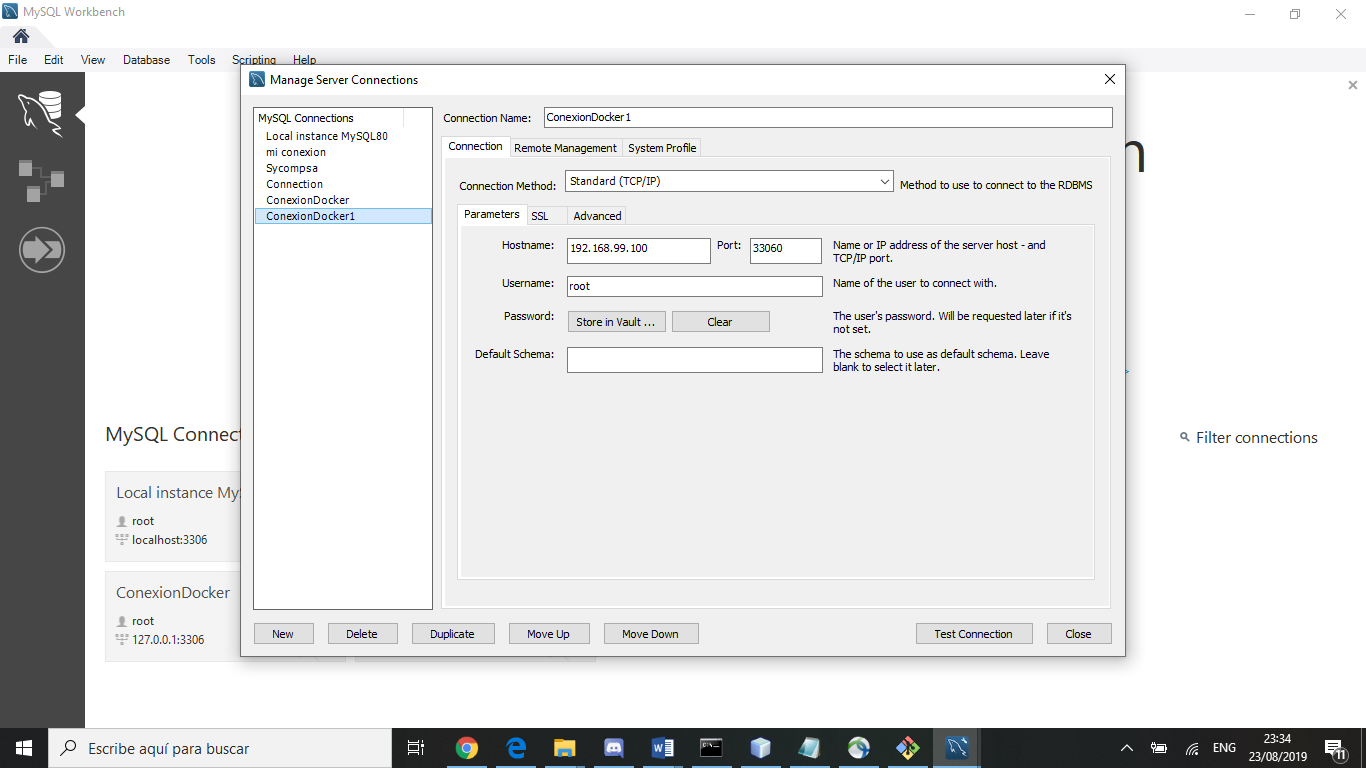
Si queremos usar la base de datos de manera interactiva desde la terminal de Docker, podemos usar el siguiente comando

**docker exec -it mysql-db mysql -p**

El comando exec nos permite ejecutar las diferentes funcionalidades de Docker.

En general para montar en si la base de datos, se creó la conexión con el contenedor directamente desde MySQL Workbench.

Para esto primero levantamos el contenedor de Docker, y en el workbench al momento de crear la conexión, le damos la ip por defecto de Docker y el puerto que le asignamos al contenedor al momento de crearlo.



Esto es lo que se hizo para poder realizar una conexión, donde el host es 192.168.99.100 que es el que Docker inicia por defecto.

Una vez hecho esto ya es solo cuestión de correr los scripts con la conexión activa.