



UNIVERSIDAD DE LAS AMÉRICAS

FACULTAD DE INGENIERÍA Y CIENCIAS APLICADAS

INGENIERÍA EN CIBERSEGURIDAD E INGENIERÍA EN TECNOLOGÍAS DE LA  
INFORMACIÓN

ITIZ3201 ADMINISTRACIÓN DE BASE DE DATOS

PROYECTO INTEGRADOR

Profesor:

Luis Patricio Moreno Buitrón

Integrantes:

Alejandra Caicedo

Mateo Cueva

Jean Moyano

04 de julio del 2023

## Tabla de contenido

I.	Descripción de la Solución Implementada .....	3
1.	Creación de la Máquina Virtual (VM) con iniciales que identifiquen al grupo .....	3
2.	Setear el nombre del host .....	3
3.	Instalación de Docker .....	4
4.	MySQL .....	7
5.	Persistencia .....	7
6.	Arranque del contenedor con configuración nombre, contraseña, puertos y persistencia de archivos.....	8
7.	Verificación de persistencia de archivos y configuración .....	8
8.	Mapeo de puerto al puerto modificado previamente .....	11
9.	Instalación del Cliente en Host .....	12
10.	Creación de roles y usuarios.....	15
11.	Diagrama lógico .....	16
12.	Diccionario de datos.....	16
13.	Faker.....	18
II.	Instrucciones para acceder al repositorio en GitHub.....	23
III.	Informe de cumplimiento de los siguientes criterios de desempeño .....	24
IV.	Anexos .....	25

## I. Descripción de la Solución Implementada

### 1. Creación de la Máquina Virtual (VM) con iniciales que identifiquen al grupo

Al momento de importar la máquina virtual hemos de modificar el nombre de la VM, de modo que cumpla con las indicaciones del docente.

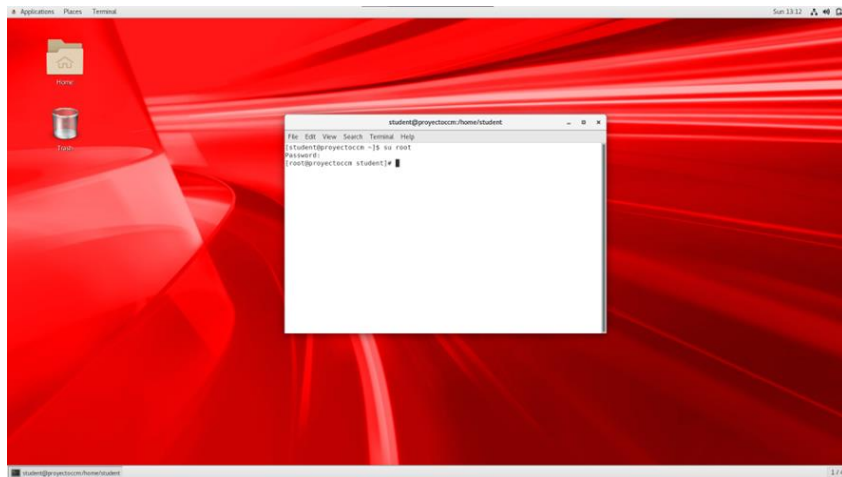
#### Preferencias de servicio

Estas son las máquinas virtuales contenidas en el servicio y las preferencias sugeridas de las máquinas virtuales importadas de VirtualBox. Puede cambiar varias de las propiedades mostradas haciendo doble clic en los elementos y deshabilitar otros usando las casillas de verificación de abajo.

Sistema virtual 1	
Nombre	ProyectoIntegradorCCm-SeedOL7
Producto	Máquina virtual para instrucción universitaria y profesional
Vendedor	Magister Luis Patricio Moreno Buitrón - GETI ec
Versión	23 Dic 2020 - 1.1
Descripción	Swap 8GB...
Tipo de SO invitado	Oracle (64-bit)
CPU	2
RAM	4096 MB
DVD	<input checked="" type="checkbox"/>
Controlador USB	<input checked="" type="checkbox"/>
Adaptador de red	<input checked="" type="checkbox"/> Intel PRO/1000 MT Desktop (82540EM)
Adaptador de red	<input checked="" type="checkbox"/> Intel PRO/1000 MT Desktop (82540EM)
Controlador de almacenamiento (IDE)	PIIX4
Controlador de almacenamiento (IDE)	PIIX4
Controlador de almacenamiento (SATA)	AHCI
Imagen de disco virtual	seedOL7-disk001.vdi
Carpeta base	C:\Users\jeanp\VirtualBox VMs
Grupo primario	/

### 2. Setear el nombre del host

Para cambiar el nombre hemos de dirigirnos al archivo “hostname”, contenido en el directorio “/etc/”, para ello necesitaremos ser usuario root.



student@proyectoccm:/home/student

File Edit View Search Terminal Help

[student@proyectoccm ~]\$ su root

Password:

[root@proyectoccm student]#

[root@proyectoccm student]# cd /etc/

[root@proyectoccm etc]# nano hostname

```
student@proyectoccm:/etc
File Edit View Search Terminal Help
GNU nano 2.3.1 File: hostname Modified
proyectoccm.lpm.com
```

Del mismo modo, asociaremos el hostname que asignamos al host modificando el archivo “hosts”, contenido en el mismo directorio.

```
student@proyectoccm:/etc
File Edit View Search Terminal Help
GNU nano 2.3.1 File: hosts
127.0.0.1 localhost.localdomain localhost
10.10.10.2 proyectoccm.lpm.com proyectoccm

[ Read 2 lines ]
^G Get Help ^O WriteOut ^R Read File ^Y Prev Page ^K Cut Text ^C Cur Pos
^X Exit ^J Justify ^W Where Is ^V Next Page ^U UnCut Text ^T To Spell
```

Tras guardar los archivos, reiniciaremos la máquina, de modo que los ajustes que realizamos se vean reflejados.

```
student@proyectoccm:/etc
File Edit View Search Terminal Help
[root@proyectoccm etc]#
```

### 3. Instalación de Docker

Para proceder, hemos de habilitar los repositorios para descargar y utilizar los paquetes necesarios para la instalación, ello accediendo al directorio “/etc/yum.repos.d/” y modificando el archivo “public-yum-ol7.repo”

```
ProjectoIntegradorCCm-SeedOL7 [Corriendo] - Oracle VM VirtualBox
Archivo Máquina Ver Entrada Dispositivos Ayuda
Applications Places Terminal
student@proyectoccm:/etc/yum.repos.d
File Edit View Search Terminal Help
GNU nano 2.3.1 File: public-yum-ol7.repo
[ol7 latest]
name=Oracle Linux $releasever Latest ($basearch)
baseurl=https://yum.oracle.com/repo/OracleLinux/OL7/latest/$basearch/
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-oracle
gpgcheck=1
enabled=1
```

```
[ol7_latest]
name=Oracle Linux $releasever Latest ($basearch)
baseurl=https://yum.oracle.com/repo/OracleLinux/OL7/latest/$basearch/
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-oracle
gpgcheck=1
enabled=1
```

```
[ol7_u6_base]
name=Oracle Linux $releasever Update 6 installation media copy ($basearch)
baseurl=https://yum.oracle.com/repo/OracleLinux/OL7/6/base/$basearch/
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-oracle
gpgcheck=1
enabled=1
```

```
[ol7_addons]
name=Oracle Linux $releasever Add ons ($basearch)
baseurl=https://yum.oracle.com/repo/OracleLinux/OL7/addons/$basearch/
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-oracle
gpgcheck=1
enabled=1
```

```
[ol7_developer]
name=Oracle Linux $releasever Development Packages ($basearch)
baseurl=https://yum.oracle.com/repo/OracleLinux/OL7/developer/$basearch/
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-oracle
gpgcheck=1
enabled=1
```

```
[root@proyectoccm yum.repos.d]# yum repoinfo
Loaded plugins: langpacks, ulninfo
Repo-id      : docker-ce-stable/7Server/x86_64
Repo-name    : Docker CE Stable - x86_64
Repo-revision: 1685087479
Repo-updated : Fri May 26 02:51:19 2023
Repo-pkgs    : 245
Repo-size    : 5.2 G
Repo-baseurl : https://download.docker.com/linux/centos/7Server/x86_64/stable
Repo-expire  : 21,600 second(s) (last: Wed Jun 28 15:22:18 2023)
  Filter     : read-only:present
Repo-filename: /etc/yum.repos.d/docker-ce.repo
```

```
Repo-id      : ol7_addons/x86_64
Repo-name    : Oracle Linux 7Server Add ons (x86_64)
Repo-revision: 1687917020
Repo-updated : Tue Jun 27 20:50:21 2023
Repo-pkgs    : 451
Repo-size    : 3.5 G
Repo-baseurl : https://yum.oracle.com/repo/OracleLinux/OL7/addons/x86_64/
Repo-expire  : 21,600 second(s) (last: Wed Jun 28 15:20:18 2023)
  Filter     : read-only:present
Repo-excluded: 303
Repo-filename: /etc/yum.repos.d/public-yum-ol7.repo
```

```
Repo-id      : ol7_developer/x86_64
Repo-name    : Oracle Linux 7Server Development Packages (x86_64)
Repo-revision: 1687917256
Repo-updated : Tue Jun 27 20:54:22 2023
Repo-pkgs    : 2,309
Repo-size    : 95 G
Repo-baseurl : https://yum.oracle.com/repo/OracleLinux/OL7/developer/x86_64/
Repo-expire  : 21,600 second(s) (last: Wed Jun 28 15:20:18 2023)
  Filter     : read-only:present
Repo-excluded: 1,991
Repo-filename: /etc/yum.repos.d/public-yum-ol7.repo
```

```

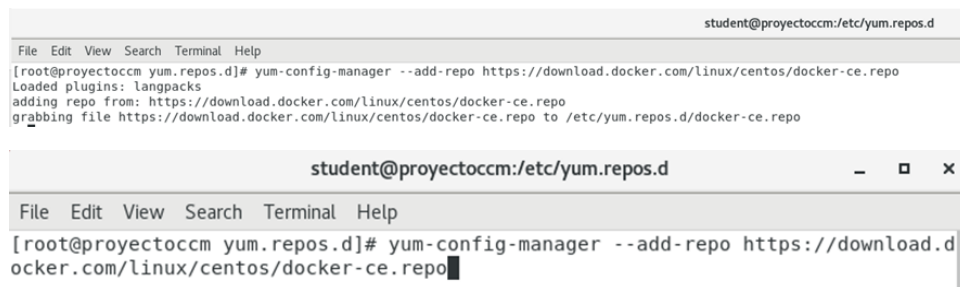
Repo-id      : ol7_latest/x86_64
Repo-name    : Oracle Linux 7Server Latest (x86_64)
Repo-revision: 1687567363
Repo-updated : Fri Jun 23 19:42:47 2023
Repo-pkgs    : 20,629
Repo-size    : 40 G
Repo-baseurl : https://yum.oracle.com/repo/OracleLinux/OL7/latest/x86_64/
Repo-expire  : 21,600 second(s) (last: Wed Jun 28 15:20:18 2023)
  Filter     : read-only:present
Repo-excluded: 5,144
Repo-filename: /etc/yum.repos.d/public-yum-ol7.repo

Repo-id      : ol7_u6_base/x86_64
Repo-name    : Oracle Linux 7Server Update 6 installation media copy (x86_64)
Repo-updated : Sun Nov  4 13:22:43 2018
Repo-pkgs    : 5,180
Repo-size    : 3.5 G
Repo-baseurl : https://yum.oracle.com/repo/OracleLinux/OL7/6/base/x86_64/
Repo-expire  : 21,600 second(s) (last: Wed Jun 28 15:20:18 2023)
  Filter     : read-only:present
Repo-excluded: 2,164
Repo-filename: /etc/yum.repos.d/public-yum-ol7.repo

repolist: 28,814

```

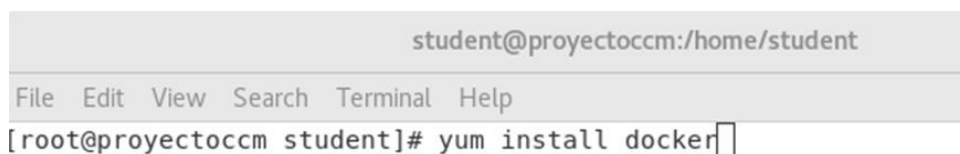
Tras ello, hemos de ejecutar el siguiente comando: `yum-config-manager --add-repo https://download.docker.com/linux/centos/docker-ce.repo` que permitirá agregar la configuración del repositorio Docker CE a la lista de repositorios del sistema.



The first screenshot shows a terminal window titled 'student@proyectoccm:/etc/yum.repos.d'. The command `yum-config-manager --add-repo https://download.docker.com/linux/centos/docker-ce.repo` is entered. The output shows 'Loaded plugins: langpacks', 'adding repo from: https://download.docker.com/linux/centos/docker-ce.repo', and 'grabbing file https://download.docker.com/linux/centos/docker-ce.repo to /etc/yum.repos.d/docker-ce.repo'.

The second screenshot shows the same terminal window after the command has been executed. The prompt is now `[root@proyectoccm yum.repos.d]#`, indicating the command was successful.

Al hacerlo, podremos instalar Docker haciendo uso de los siguientes comandos:



The screenshot shows a terminal window titled 'student@proyectoccm:/home/student'. The command `yum install docker` is entered at the prompt `[root@proyectoccm student]#`.

Tras correr el comando, hemos de levantar el servicio con los comandos `systemctl enable docker` y `systemctl start docker`.

```

[root@proyectoccm student]# systemctl enable docker
[root@proyectoccm student]# systemctl start docker

```

Podremos revisar si el servicio se encuentra activo haciendo uso del comando `systemctl status docker`.

```
ProjectoIntegradorCCm-SeedOL7 [Corriendo] - Oracle VM VirtualBox
Archivo Máquina Ver Entrada Dispositivos Ayuda
Applications Places Terminal

student@proyectoccm:/home/student

File Edit View Search Terminal Help

[root@proyectoccm student]# systemctl status docker
● docker.service - Docker Application Container Engine
   Loaded: loaded (/usr/lib/systemd/system/docker.service; enabled; vendor preset: disabled)
   Active: active (running) since Sun 2023-07-02 13:08:17 -05; 3h 22min ago
     Docs: https://docs.docker.com
   Main PID: 3333 (dockerd)
    CGroup: /system.slice/docker.service
            └─3333 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock

Jul 02 13:08:16 proyectoccm.lpm.com dockerd[3333]: time="2023-07-02T13:08:16.894699503-05:00" level=info msg="Default bri
Jul 02 13:08:16 proyectoccm.lpm.com dockerd[3333]: time="2023-07-02T13:08:16.894946383-05:00" level=info msg="failed to r
Jul 02 13:08:16 proyectoccm.lpm.com dockerd[3333]: time="2023-07-02T13:08:16.969558658-05:00" level=info msg="Loading con
Jul 02 13:08:17 proyectoccm.lpm.com dockerd[3333]: time="2023-07-02T13:08:17.103502312-05:00" level=warning msg="Not usin
Jul 02 13:08:17 proyectoccm.lpm.com dockerd[3333]: time="2023-07-02T13:08:17.104211812-05:00" level=warning msg="WARNING:
Jul 02 13:08:17 proyectoccm.lpm.com dockerd[3333]: time="2023-07-02T13:08:17.104234120-05:00" level=warning msg="WARNING:
Jul 02 13:08:17 proyectoccm.lpm.com dockerd[3333]: time="2023-07-02T13:08:17.104274353-05:00" level=info msg="Docker daem
Jul 02 13:08:17 proyectoccm.lpm.com dockerd[3333]: time="2023-07-02T13:08:17.107081474-05:00" level=info msg="Daemon has
Jul 02 13:08:17 proyectoccm.lpm.com dockerd[3333]: time="2023-07-02T13:08:17.174094665-05:00" level=info msg="API listen
Jul 02 13:08:17 proyectoccm.lpm.com systemd[1]: Started Docker Application Container Engine.
Hint: Some lines were ellipsized, use -l to show in full.
[root@proyectoccm student]#
```

```
ProjectoIntegradorCCm-SeedOL7 [Corriendo] - Oracle VM VirtualBox
Archivo Máquina Ver Entrada Dispositivos Ayuda
Applications Places Terminal

st

File Edit View Search Terminal Help

[root@proyectoccm student]# systemctl status docker
● docker.service - Docker Application Container Engine
   Loaded: loaded (/usr/lib/systemd/system/docker.service; enabled; vendor preset: disabled)
   Active: active (running) since Sun 2023-07-02 13:08:17 -05; 3h 22min ago
     Docs: https://docs.docker.com
   Main PID: 3333 (dockerd)
    CGroup: /system.slice/docker.service
            └─3333 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock
```

## 4. MySQL

Para utilizar la versión más reciente de MySQL hemos de utilizar el siguiente comando:

```
student@proyectoccm:/home/student

File Edit View Search Terminal Help

[root@proyectoccm student]# docker pull mysql:latest
latest: Pulling from library/mysql
Digest: sha256:15f069202c46cf861ce429423ae3f8dfa6423306fbf399eaf36094ce30dd75c
Status: Image is up to date for mysql:latest
docker.io/library/mysql:latest
[root@proyectoccm student]#
```

Y para posteriormente interactuar con el, utilizaremos *sudo yum install mysql*

```
[root@proyectoccm student]# sudo yum install mysql
Loaded plugins: langpacks, ulninfo
docker-ce-stable
ol7_addons
ol7_addons/x86_64/updateinfo
ol7_latest/x86_64/primary_db
Resolving Dependencies
--> Running transaction check
--> Package mariadb.x86_64 1:5.5.68-1.el7 will be installed
--> Processing Dependency: mariadb-libs(x86-64) = 1:5.5.68-1.el7 for package: 1:mariadb-5.5.68-1.el7.x86_64
--> Running transaction check
--> Package mariadb-libs.x86_64 1:5.5.60-1.el7_5 will be updated
--> Package mariadb-libs.x86_64 1:5.5.68-1.el7 will be an update
--> Finished Dependency Resolution

Dependencies Resolved
```

## 5. Persistencia

Para que pueda haber persistencia hemos de crear volúmenes para archivos y configuración.



```
student@proyectoccm:/home/student/mysql

File Edit View Search Terminal Help

[root@proyectoccm mysql]# mkdir -p data
[root@proyectoccm mysql]# ls
data
[root@proyectoccm mysql]# mkdir -p conf
[root@proyectoccm mysql]# ls
conf data
[root@proyectoccm mysql]#
```

## 6. Arranque del contenedor con configuración nombre, contraseña, puertos y persistencia de archivos.

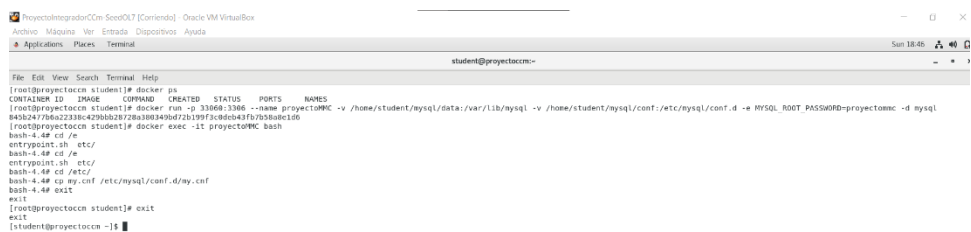
Utilizaremos los comandos:

```
docker run -p 33060:3306 --name proyectoMMC -v /home/student/mysql/data:/var/lib/mysql -v /home/student/mysql/conf:/etc/mysql/conf.d -e MYSQL_ROOT_PASSWORD=proyectommc -d mysql
```

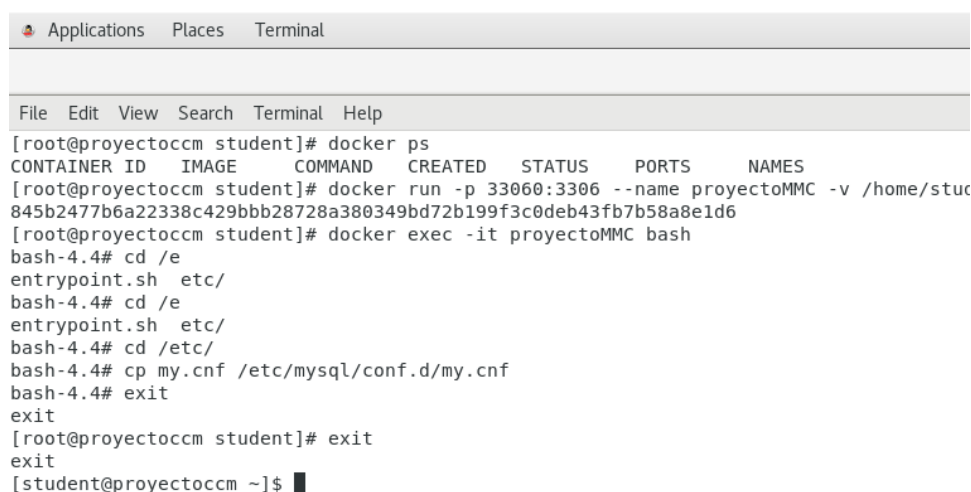
```
docker exec -it proyectoMMC bash
```

```
cd /etc/
```

```
cp my.cnf /etc/mysql/conf.d/my.cnf
```



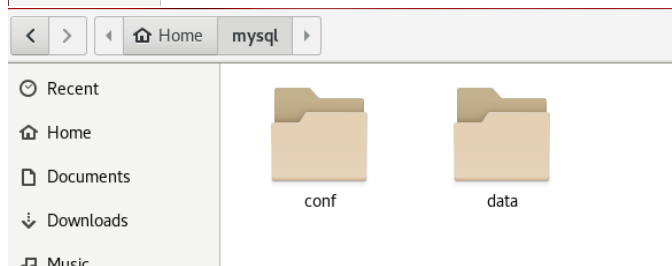
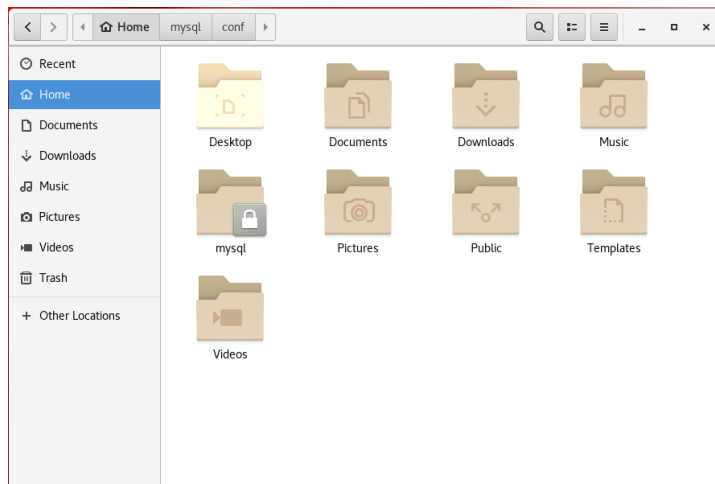
```
ProjectoIntegradorCm Seed017 [Corriendo] - Oracle VM VirtualBox
Archivo Máquina Ver Entrada Dispositivos Ayuda
Applications Places Terminal
student@proyectoccm:~
File Edit View Search Terminal Help
[root@proyectoccm student]# docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS              PORTS              NAMES
845b2477b6a22338c429bbb28728a380349bd72b199f3c0deb43fb7b58a8e1d6
[root@proyectoccm student]# docker run -p 33060:3306 --name proyectoMMC -v /home/student/mysql/data:/var/lib/mysql -v /home/student/mysql/conf:/etc/mysql/conf.d -e MYSQL_ROOT_PASSWORD=proyectommc -d mysql
bash-4.4# cd /
entrypoint.sh etc/
bash-4.4# cd /
entrypoint.sh etc/
bash-4.4# cd /etc/
bash-4.4# cp my.cnf /etc/mysql/conf.d/my.cnf
bash-4.4# exit
exit
[root@proyectoccm student]# exit
exit
[student@proyectoccm ~]$
```



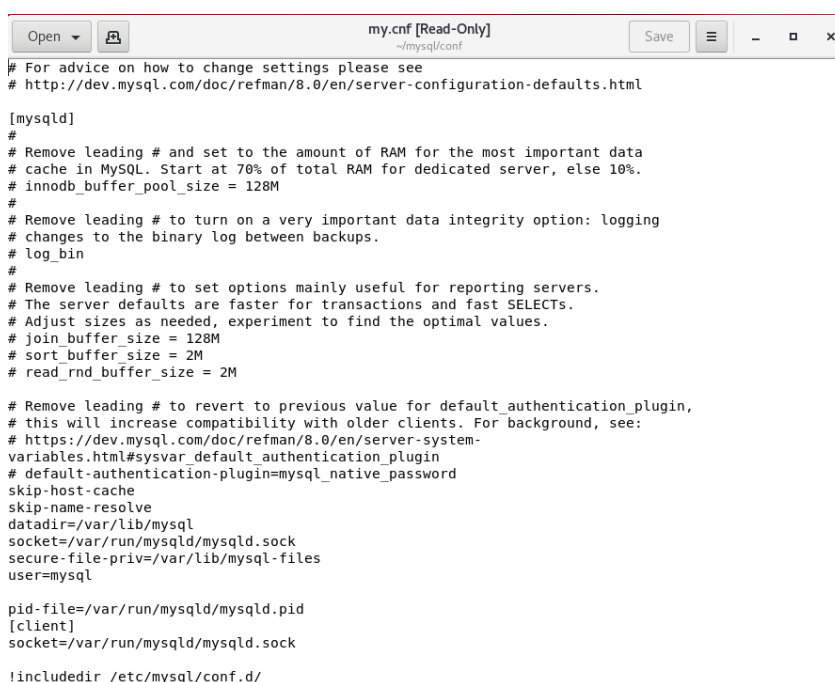
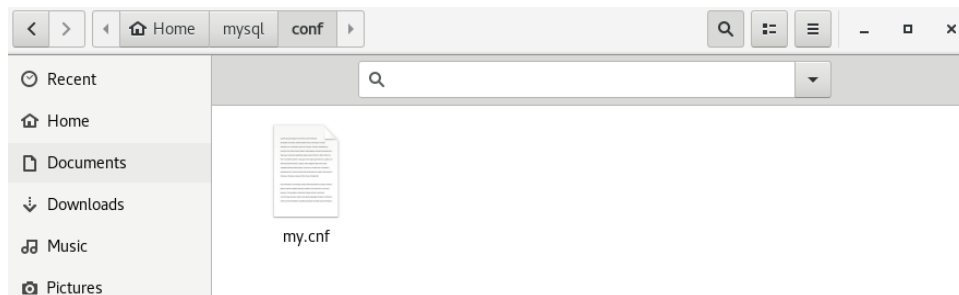
```
Applications Places Terminal
File Edit View Search Terminal Help
[root@proyectoccm student]# docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS              PORTS              NAMES
845b2477b6a22338c429bbb28728a380349bd72b199f3c0deb43fb7b58a8e1d6
[root@proyectoccm student]# docker run -p 33060:3306 --name proyectoMMC -v /home/student/mysql/data:/var/lib/mysql -v /home/student/mysql/conf:/etc/mysql/conf.d -e MYSQL_ROOT_PASSWORD=proyectommc -d mysql
bash-4.4# cd /
entrypoint.sh etc/
bash-4.4# cd /
entrypoint.sh etc/
bash-4.4# cd /etc/
bash-4.4# cp my.cnf /etc/mysql/conf.d/my.cnf
bash-4.4# exit
exit
[root@proyectoccm student]# exit
exit
[student@proyectoccm ~]$
```

## 7. Verificación de persistencia de archivos y configuración

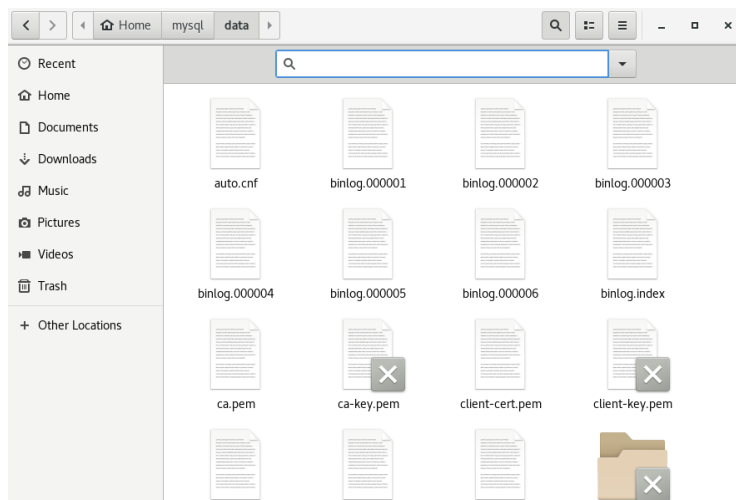




## Configuraciones:



## Archivos:



```
root@proyectoccm:/home/student/mysql/data
File Edit View Search Terminal Help
[root@proyectoccm ~]# cd /home/student/mysql/conf/
[root@proyectoccm conf]# ls
my.cnf
[root@proyectoccm conf]# cd /home/student/mysql/data/
[root@proyectoccm data]# ls
auto.cnf      ca-key.pem      ib_buffer_pool  mysql.ibd      server-key.pem
binlog.000001 ca.pem          ibdata1         mysql.sock      sys
binlog.000002 client-cert.pem ibtmp1          performance_schema undo_001
binlog.000003 client-key.pem  #innodb_redo   private_key.pem undo_002
binlog.000004 #ib_16384_0.dblwr #innodb_temp   public_key.pem  server-cert.pem
binlog.index  #ib_16384_1.dblwr mysql          server-cert.pem
[root@proyectoccm data]#
```

## Conexión con el servidor de MySQL (desde CentOS):

```
Applications Places Terminal
student@proyectoccm:/home/student
File Edit View Search Terminal Help
[root@proyectoccm student]# docker exec -it proyectoccm mysql -u root -p
mysql: [Warning] Skipping 'includedir /etc/mysql/conf.d/' directive as maximum include recursion level was reached in file /etc/mysql/conf.d/my.cnf at line 36.
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 8
Server version: 8.0.33 MySQL Community Server - GPL

Copyright (c) 2000, 2023, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
```

## Procederemos a verificar el puerto de conexión por defecto:

Oracle is a registered trademark of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

```
mysql> show variables like 'port';
+-----+
| Variable_name | Value |
+-----+
| port          | 3306  |
+-----+
1 row in set (0.01 sec)

mysql>
```

## Modificación de archivo de configuración para cambio de puerto (no default)

Para modificar el puerto que se utiliza por defecto hemos de añadir *port=7777* (o el puerto que deseemos) en el archivo de configuración.

```
student@proyectoccm:/home/student/mysql/conf
File Edit View Search Terminal Help
GNU nano 2.3.1 File: my.cnf

# Remove leading # to revert to previous value for default authentication plugin,
# this will increase compatibility with older clients. For background, see:
# https://dev.mysql.com/doc/refman/8.0/en/server-system-variables.html#sysvar_default_authentication_plugin
# default-authentication-plugin=mysql_native_password
skip-host-cache
skip-name-resolve
datadir=/var/lib/mysql
socket=/var/run/mysql/mysql.sock
secure-file-priv=/var/lib/mysql-files
user=mysql
port=7777

pid-file=/var/run/mysql/mysql.pid
[client]
socket=/var/run/mysql/mysql.sock

!includedir /etc/mysql/conf.d/
```

## 8. Mapeo de puerto al puerto modificado previamente

Para ello es necesario modificar el siguiente comando:

```
docker run -p 33060:3306 --name proyectoMMC -v /home/student/mysql/data:/var/lib/mysql -
v /home/student/mysql/conf:/etc/mysql/conf.d -e MYSQL_ROOT_PASSWORD=proyectommc -
d mysql
```

```
docker exec -it proyectoMMC bash
```

```
cd /etc/
```

```
cp my.cnf /etc/mysql/conf.d/my.cnf
```

Estableciendo el nuevo puerto a utilizar en `-p xxxx:xxxx`

De modo que tenemos:

```
docker run -p 33060:7777 --name proyectoMMC -v /home/student/mysql/data:/var/lib/mysql -
v /home/student/mysql/conf:/etc/mysql/conf.d -e MYSQL_ROOT_PASSWORD=proyectommc -
d mysql
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
71828bfb13a7	mysql	"docker-entrypoint.s..."	4 seconds ago	Up 3 seconds	3306/tcp, 33060/tcp, 0.0.0.0:33060->7777/tcp	proyectoMMC

```
[root@proyectoccm student]#
```

## Validación:

```
Applications Places Terminal
student@proyectoccm:/home/student
File Edit View Search Terminal Help
[root@proyectoccm student]# docker exec -it proyectoMMC mysql -u root -p
mysql: [Warning] Skipping 'includedir /etc/mysql/conf.d/' directive as maximum include recursion level was reached in file /etc/mysql/conf.d/my.cnf at line 30
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 9
Server version: 8.0.33 MySQL Community Server - GPL

Copyright (c) 2000, 2023, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> show variables like 'port';
+-----+-----+
| Variable_name | Value |
+-----+-----+
| port          | 7777  |
+-----+-----+
1 row in set (0.01 sec)

mysql>
```

Copyright (c) 2000, 2023, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

```
mysql> show variables like 'port';
```

```
+-----+-----+
| Variable_name | Value |
+-----+-----+
| port          | 7777  |
+-----+-----+
1 row in set (0.01 sec)
```

```
mysql> █
```

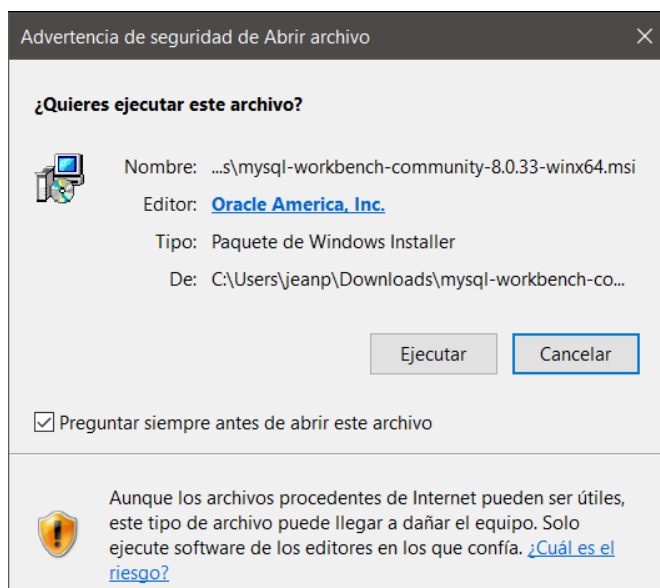
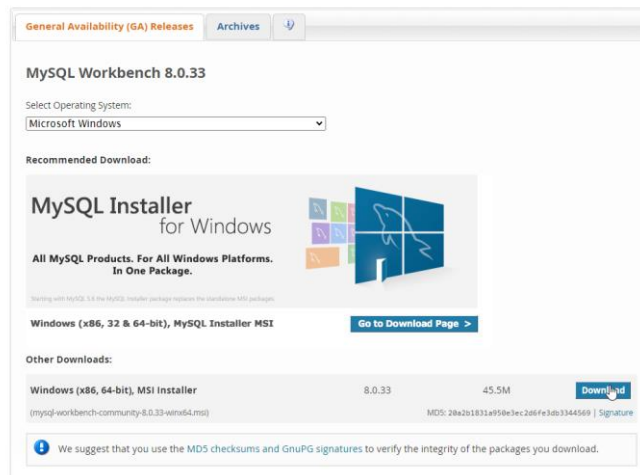
## 9. Instalación del Cliente en Host

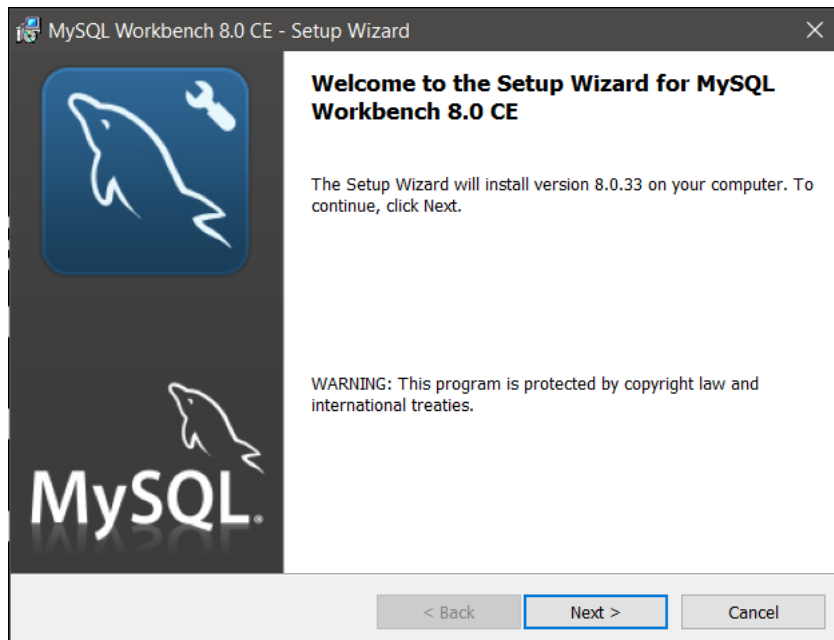
Para este ejercicio, es necesario interactuar con nuestra base de datos desde el host, para ello haremos uso de MySQL Workbench.

Para instalarla nos dirigiremos al sitio oficial.

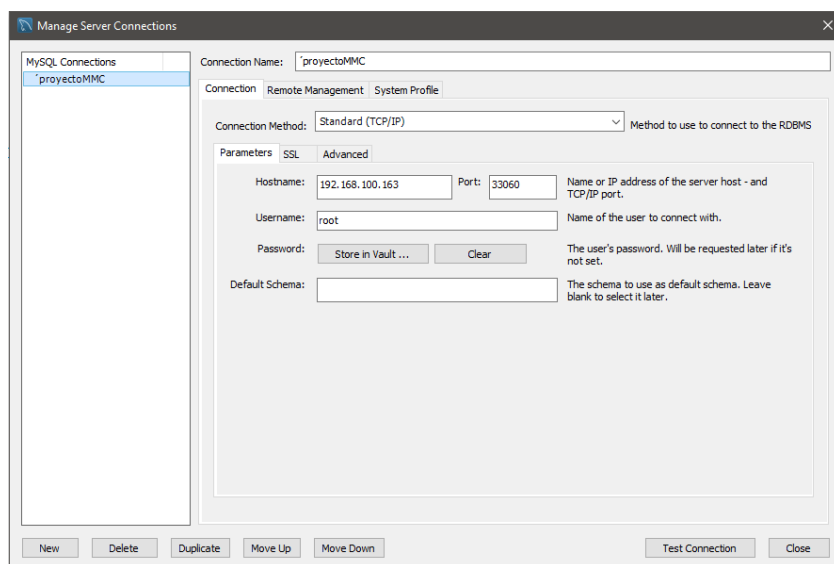
🔗 [MySQL Community Downloads](#)

← MySQL Workbench





Conexión desde host:



Creación de la base de datos:

```

1  -- Create the database if it doesn't exist
2  CREATE DATABASE IF NOT EXISTS ProyectoCCM;
3  USE ProyectoCCM;
4
5  -- Creación de la tabla "Sucursal"
6  CREATE TABLE Sucursal (
7      ID_Sucursal INT PRIMARY KEY,
8      Nombre VARCHAR(100),
9      Direccion VARCHAR(200),
10     Ciudad VARCHAR(100),
11     Telefono VARCHAR(50)
12 );
13
14 -- Creación de la tabla "Empleado"
15 CREATE TABLE Empleado (
16     ID_Empleado INT PRIMARY KEY,
17     Nombre VARCHAR(100),
18     Apellido VARCHAR(100),
19     Direccion VARCHAR(200),
20     Telefono VARCHAR(50),
21     ID_Sucursal INT,
22     FOREIGN KEY (ID_Sucursal) REFERENCES Sucursal(ID_Sucursal)
23 );
24
25 -- Creación de la tabla "Producto"
26 CREATE TABLE Producto (
27     ID_Producto INT PRIMARY KEY,
28     Nombre VARCHAR(100),
29     Descripcion VARCHAR(200),
30     Precio DECIMAL(10, 2)
31 );
32
33 -- Creación de la tabla "Inventario"
34 CREATE TABLE Inventario (
35     ID_Inventario INT PRIMARY KEY,
36     ID_Sucursal INT,
37     ID_Producto INT,
38     Cantidad INT,
39     FOREIGN KEY (ID_Sucursal) REFERENCES Sucursal(ID_Sucursal),
40     FOREIGN KEY (ID_Producto) REFERENCES Producto(ID_Producto)
41 );
42
43 -- Creación de la tabla "Venta"
44 CREATE TABLE Venta (
45     ID_Venta INT PRIMARY KEY,
46     ID_Sucursal INT,
47     ID_Empleado INT,
48     Fecha DATE,
49     Total DECIMAL(10, 2),
50     FOREIGN KEY (ID_Sucursal) REFERENCES Sucursal(ID_Sucursal),
51     FOREIGN KEY (ID_Empleado) REFERENCES Empleado(ID_Empleado)
52 );
--

```

## 10. Creación de roles y usuarios

```
54 -- Create the roles only if they don't exist
55 CREATE ROLE IF NOT EXISTS 'usuario_leer';
56 CREATE ROLE IF NOT EXISTS 'usuario_editar';
57
58 -- Create the users only if they don't exist
59 CREATE USER IF NOT EXISTS 'UsuarioLeer'@'localhost' IDENTIFIED BY '1234';
60 CREATE USER IF NOT EXISTS 'UsuarioEditor'@'localhost' IDENTIFIED BY '1234';
61
62 -- Grant privileges to the roles
63 GRANT SELECT ON ProyectoCCM.* TO 'usuario_leer';
64 GRANT INSERT, UPDATE, DELETE ON ProyectoCCM.* TO 'usuario_editar';
65
66 -- Grant roles to the users
67 GRANT usuario_leer TO 'UsuarioLeer'@'localhost';
68 GRANT usuario_editar TO 'UsuarioEditor'@'localhost';
69
70 -- Flush privileges to apply the changes
71 FLUSH PRIVILEGES;
```

Comprobación de creación de usuarios

```
mysql> SELECT User, Host From mysql.user;
```

User	Host
root	%
usuario_editar	%
usuario_leer	%
UsuarioEditor	localhost
UsuarioLeer	localhost
mysql.infoschema	localhost
mysql.session	localhost
mysql.sys	localhost
root	localhost

```
9 rows in set (0.00 sec)
```

Comprobación de roles de usuario

```
1 • USE FarmaciasCCM;
2 • SELECT USER, HOST FROM mysql.user WHERE USER = 'UsuarioLeer' AND HOST = 'localhost';
3 • SHOW GRANTS FOR 'UsuarioLeer'@'localhost';
4
```

Result Grid		Filter Rows:	Export:	Wrap Cell Content:
Grants for UsuarioLeer@localhost				
GRANT USAGE ON *.* TO 'UsuarioLeer'@'loc...				
GRANT 'usuario_leer'@'%' TO 'UsuarioLeer'...				



```

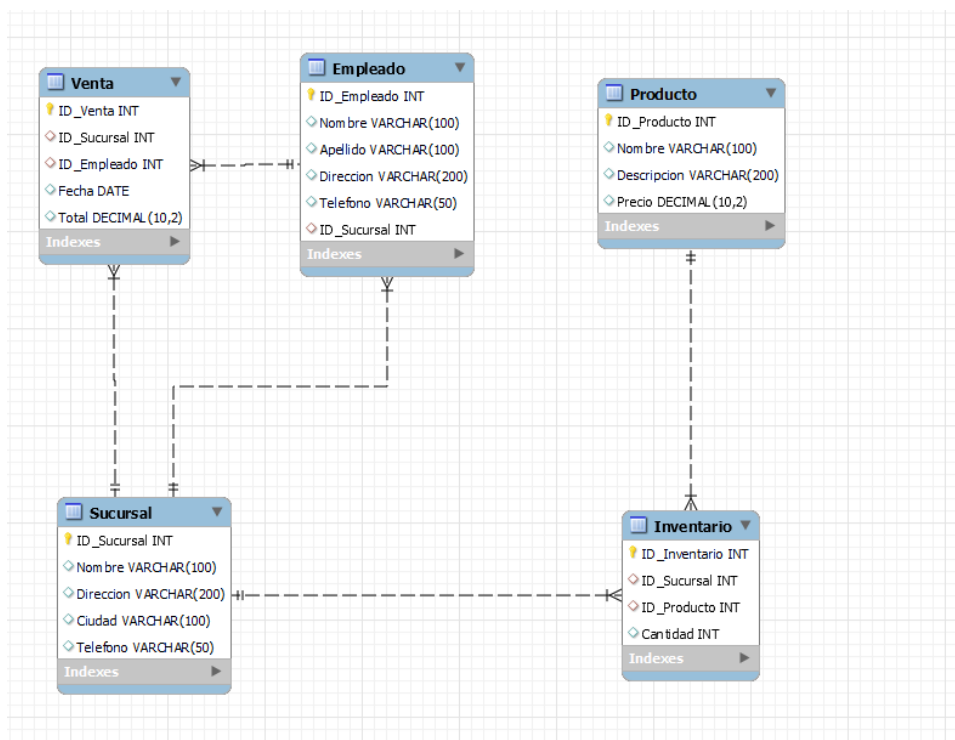
1 • USE FarmaciasCCM;
2 • SELECT USER, HOST FROM mysql.user WHERE USER = 'UsuarioEditor' AND HOST = 'localhost';
3 • SHOW GRANTS FOR 'UsuarioEditor'@'localhost';
4 |

```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: |

Grants for UsuarioEditor@localhost	
▶	GRANT USAGE ON *.* TO 'UsuarioEditor'@'lo...
	GRANT 'usuario_editor'@'%' TO 'UsuarioEdi...

## 11. Diagrama lógico



## 12. Diccionario de datos

Tabla "Sucursales":

ID\_Sucursal (Clave primaria): Identificador único de la sucursal.

Nombre: Nombre de la sucursal.

Dirección: Dirección de la sucursal.

Ciudad: Ciudad donde se encuentra la sucursal.

Teléfono: Número de teléfono de la sucursal.

Tabla "Empleados":

ID\_Empleado (Clave primaria): Identificador único del empleado.

Nombre: Nombre del empleado.

Apellido: Apellido del empleado.

Dirección: Dirección del empleado.

Teléfono: Número de teléfono del empleado.

ID\_Sucursal (Clave foránea referenciando a la tabla "Sucursales"): ID de la sucursal a la que pertenece el empleado.

Tabla "Productos":

ID\_Producto (Clave primaria): Identificador único del producto.

Nombre: Nombre del producto.

Descripción: Descripción del producto.

Precio: Precio del producto.

Tabla "Inventario":

ID\_Inventario (Clave primaria): Identificador único del registro de inventario.

ID\_Sucursal (Clave foránea referenciando a la tabla "Sucursales"): ID de la sucursal a la que pertenece el inventario.

ID\_Producto (Clave foránea referenciando a la tabla "Productos"): ID del producto en el inventario.

Cantidad: Cantidad disponible del producto en la sucursal.

Tabla "Ventas":

ID\_Venta (Clave primaria): Identificador único de la venta.

ID\_Sucursal (Clave foránea referenciando a la tabla "Sucursales"): ID de la sucursal donde se realizó la venta.

ID\_Empleado (Clave foránea referenciando a la tabla "Empleados"): ID del empleado que realizó la venta.

Fecha: Fecha en que se realizó la venta.

Total: Monto total de la venta.

### 13. Faker

Para poblar las tablas con información generada de forma aleatoria se utilizó el siguiente código de python, utilizando la librería Faker.

```
from faker import Faker
from datetime import datetime, date
import mysql.connector

fake = Faker(locale='es_ES')

# Connection setting
config = {
    'user': 'root',
    'password': 'proyectoccm',
    'host': '192.168.100.163',
    'port': 33060,
    'database': 'FarmaciasCCM',
}

def create_product(num_product, cursor):
    for i in range(1, num_product + 1):
        product_name = fake.random_element(elements=("Loratadina", "Claritin", "Alavert",
        "Wal-itin",
        "Hidroxicloroquina", "Plaquenil", "Quineprox", "Dolquine",
        "Ibuprofeno", "Advil", "Motrin", "Nuprin", "Naproxeno",
        "Aleve", "Naprosyn", "Anaprox", "Aspirina", "Bayer", "Ecotrin",
        "Bufferin", "Simvastatina", "Zocor", "Simcor", "Lipex",
        "Atorvastatina", "Lipitor", "Torvast", "Sortis", "Lisinopril",
        "Prinivil", "Zestril", "Metformina", "Glucophage", "Fortamet",
        "Riomet", "Omeprazol", "Prilosec", "Losec", "Zegerid",
        "Pantoprazol", "Protonix", "Pantoloc", "Panto IV",
        "Ranitidina", "Zantac", "Tritec", "Pepcid", "Citalopram",
        "Celexa", "Seropram", "Cipramil", "Escitalopram",
        "Lexapro", "Cipralex"))
        product_description = fake.random_element(elements=("Analgésico y antifebril",
        "Medicamento para aliviar alergias",
        "Antidepresivo para trastornos del estado de ánimo",
        "Antiinflamatorio para dolores articulares",
        "Antihistamínico para síntomas de alergia",
        "Antiácido para acidez estomacal e indigestión",
        "Antibiótico para infecciones bacterianas",
        "Antipsicótico para trastornos de salud mental",
        "Broncodilatador para afecciones respiratorias",
        "Anticoagulante para prevenir coágulos sanguíneos",
        "Estatina para reducir el colesterol",
```

```
"Diurético para controlar la retención de líquidos",
"Antiviral para infecciones virales",
"Antifúngico para infecciones por hongos",
"Anticoagulante para prevenir coágulos",
"Antihipertensivo para reducir la presión arterial",
"Medicamento antidiabético para el control de la diabetes",
"Ansiolítico para trastornos de ansiedad",
"Antiemético para prevenir náuseas y vómitos",
"Suplemento vitamínico y mineral",
"Antiinflamatorio para tratar el dolor",
"Antipirético para reducir la fiebre"))
```

```
query = "INSERT INTO Producto (ID_Producto, Nombre, Descripcion, Precio)
VALUES (%s, %s, %s, %s)"
values = (i, product_name, product_description, fake.random_int(min=10,
max=100))
cursor.execute(query, values)
```

```
def create_sale(num_sales, cursor):
    start_date = date(2021, 1, 1)
    end_date = date(2023, 12, 31)
```

```
# Get existing store_ids from the Sucursal table
cursor.execute("SELECT ID_Sucursal FROM Sucursal")
store_ids = [row[0] for row in cursor.fetchall()]
```

```
for i in range(1, num_sales + 1):
    store_id = fake.random_element(elements=store_ids)
    employee_id = fake.random_int(min=1, max=100)
    date_quantity = fake.date_between(start_date=start_date, end_date=end_date)
    total = fake.random_int(min=10, max=100)
```

```
query = "INSERT INTO Venta (ID_Venta, ID_Sucursal, ID_Empleado, Fecha,
Total) VALUES (%s, %s, %s, %s, %s)"
values = (i, store_id, employee_id, date_quantity, total)
cursor.execute(query, values)
```

```
def create_store(num_store, cursor):
    for i in range(1, num_store + 1):
        store_name = fake.random_element(elements=("Farmacia San Martín", "Farmacia
Esperanza",
"Farmacia del Sol", "Farmacia Santa Cruz",
"Farmacia Nueva Vida", "Farmacia San Rafael",
"Farmacia El Faro", "Farmacia Santa Clara",
"Farmacia La Paz", "Farmacia San José"))
```

```

store_address = fake.street_address()
store_city = fake.city()
store_phone = fake.phone_number()

query = "INSERT INTO Sucursal (ID_Sucursal, Nombre, Direccion, Ciudad,
Telefono) VALUES (%s, %s, %s, %s, %s)"
values = (i, store_name, store_address, store_city, store_phone)
cursor.execute(query, values)

def create_inv(num_inv, cursor):
    for i in range(1, num_inv + 1):
        store_id = fake.random_int(min=1, max=4)
        product_id = fake.random_int(min=1, max=100)
        inv_quant = fake.random_int(min=1, max=100)

        query = "INSERT INTO Inventario (ID_Inventario, ID_Sucursal, ID_Producto,
Cantidad) VALUES (%s, %s, %s, %s)"
        values = (i, store_id, product_id, inv_quant)
        cursor.execute(query, values)

def create_employee(num_employee, cursor):
    for i in range(1, num_employee + 1):
        employee_name = fake.first_name()
        employee_surname = fake.last_name()
        employee_address = fake.address()
        employee_city = fake.city()
        store_id = fake.random_int(min=1, max=4)

        query = "INSERT INTO Empleado (ID_Empleado, Nombre, Apellido, Direccion,
Telefono, ID_Sucursal) VALUES (%s, %s, %s, %s, %s, %s)"
        values = (i, employee_name, employee_surname, employee_address, employee_city,
store_id)
        cursor.execute(query, values)

def import_data_to_mysql():
    cnx = mysql.connector.connect(**config)
    cursor = cnx.cursor()

    # Create and import stores
    create_store(num_store=10, cursor=cursor)

    # Create and import products
    create_product(num_product=100, cursor=cursor)

```

```
# Create and import employees
create_employee(num_employee=100, cursor=cursor)
```

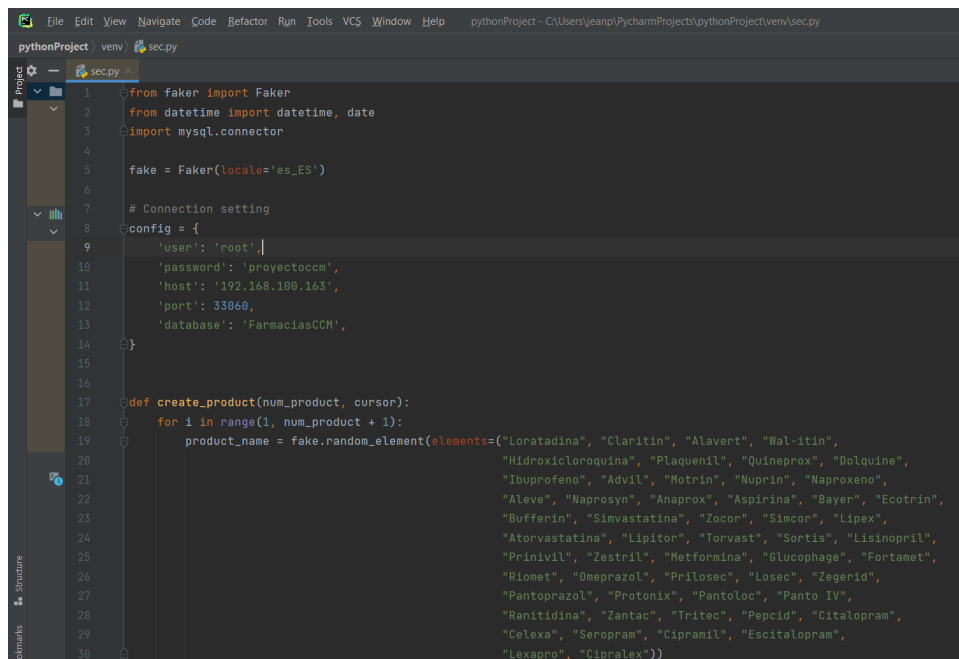
```
# Create and import inventory
create_inv(num_inv=100, cursor=cursor)
```

```
# Create and import sales
create_sale(num_sales=100, cursor=cursor)
```

```
cnx.commit()
cursor.close()
cnx.close()
```

```
if __name__ == '__main__':
    import_data_to_mysql()
```

Evidencia en IDE (PyCharm):

A screenshot of the PyCharm IDE interface. The main editor window displays a Python script named 'sec.py'. The script imports 'Faker' from the 'faker' library, 'datetime' from the 'datetime' library, and 'mysql.connector'. It sets a locale for 'fake' to 'es\_ES'. A configuration dictionary is defined with database connection details: user 'root', password 'proyectoccm', host '192.168.100.163', port '33060', and database 'FarmaciasCCM'. A function 'create\_product' is defined, which takes 'num\_product' and 'cursor' as arguments. It loops through a range of 1 to 'num\_product + 1', generating random product names from a list of 40 various drug names. The IDE interface includes a top menu bar, a left sidebar with 'Project' and 'Structure' views, and a bottom status bar. The file path in the title bar is 'pythonProject - C:\Users\jeanp\PycharmProjects\pythonProject\venv\sec.py'.

```
pythonProject \ venv \ secpy
Project
secpy
31 product_description = fake.random_element(elements=("Analgesico y antifebril",
32 "Medicamento para aliviar alergias",
33 "Antidepresivo para trastornos del estado de ánimo",
34 "Antiinflamatorio para dolores articulares",
35 "Antihistaminico para sintomas de alergia",
36 "Antiácido para acidez estomacal e indigestión",
37 "Antibiótico para infecciones bacterianas",
38 "Antipsicótico para trastornos de salud mental",
39 "Broncodilatador para afecciones respiratorias",
40 "Anticoagulante para prevenir coágulos sanguíneos",
41 "Estatina para reducir el colesterol",
42 "Diurético para controlar la retención de líquidos",
43 "Antiviral para infecciones virales",
44 "Antifúngico para infecciones por hongos",
45 "Anticoagulante para prevenir coágulos",
46 "Antihipertensivo para reducir la presión arterial",
47 "Medicamento antidiabético para el control de la diabetes",
48 "Ansiolítico para trastornos de ansiedad",
49 "Antiemético para prevenir náuseas y vómitos",
50 "Suplemento vitamínico y mineral",
51 "Antiinflamatorio para tratar el dolor",
52 "Antipirético para reducir la fiebre"))
53
54 query = "INSERT INTO Producto (ID_Producto, Nombre, Descripcion, Precio) VALUES (%s, %s, %s, %s)"
55 values = (i, product_name, product_description, fake.random_int(min=10, max=100))
56 cursor.execute(query, values)
57
58
```

```
pythonProject \ venv \ secpy
Project
secpy
58
59 def create_sale(num_sales, cursor):
60     start_date = date(2021, 1, 1)
61     end_date = date(2023, 12, 31)
62
63     # Get existing store_ids from the Sucursal table
64     cursor.execute("SELECT ID_Sucursal FROM Sucursal")
65     store_ids = [row[0] for row in cursor.fetchall()]
66
67     for i in range(1, num_sales + 1):
68         store_id = fake.random_element(elements=store_ids)
69         employee_id = fake.random_int(min=1, max=100)
70         date_quantity = fake.date_between(start_date=start_date, end_date=end_date)
71         total = fake.random_int(min=10, max=100)
72
73         query = "INSERT INTO Venta (ID_Venta, ID_Sucursal, ID_Empleado, Fecha, Total) VALUES (%s, %s, %s, %s, %s)"
74         values = (i, store_id, employee_id, date_quantity, total)
75         cursor.execute(query, values)
76
77
```

```
pythonProject \ venv \ secpy
Project
secpy
77
78 def create_store(num_store, cursor):
79     for i in range(1, num_store + 1):
80         store_name = fake.random_element(elements=("Farmacia San Martín", "Farmacia Esperanza",
81 "Farmacia del Sol", "Farmacia Santa Cruz",
82 "Farmacia Nueva Vida", "Farmacia San Rafael",
83 "Farmacia El Faro", "Farmacia Santa Clara",
84 "Farmacia La Paz", "Farmacia San José"))
85
86         store_address = fake.street_address()
87         store_city = fake.city()
88         store_phone = fake.phone_number()
89
90         query = "INSERT INTO Sucursal (ID_Sucursal, Nombre, Direccion, Ciudad, Telefono) VALUES (%s, %s, %s, %s, %s)"
91         values = (i, store_name, store_address, store_city, store_phone)
92         cursor.execute(query, values)
93
94 def create_inv(num_inv, cursor):
95     for i in range(1, num_inv + 1):
96         store_id = fake.random_int(min=1, max=4)
97         product_id = fake.random_int(min=1, max=100)
98         inv_quant = fake.random_int(min=1, max=100)
99
100         query = "INSERT INTO Inventario (ID_Inventario, ID_Sucursal, ID_Producto, Cantidad) VALUES (%s, %s, %s, %s)"
101         values = (i, store_id, product_id, inv_quant)
102         cursor.execute(query, values)
103
104
```

```
pythonProject \ venv \ secpy
Project
secpy
104
105 def create_employee(num_employee, cursor):
106     for i in range(1, num_employee + 1):
107         employee_name = fake.first_name()
108         employee_surname = fake.last_name()
109         employee_address = fake.address()
110         employee_city = fake.city()
111         store_id = fake.random_int(min=1, max=4)
112
113         query = "INSERT INTO Empleado (ID_Empleado, Nombre, Apellido, Direccion, Telefono, ID_Sucursal) VALUES (%s, %s, %s, %s, %s, %s)"
114         values = (i, employee_name, employee_surname, employee_address, employee_city, store_id)
115         cursor.execute(query, values)
116
117
```



```

pythonProject > venv > sec.py
116
117
118 def import_data_to_mysql():
119     cnx = mysql.connector.connect(**config)
120     cursor = cnx.cursor()
121
122     # Create and import stores
123     create_store(num_store=10, cursor=cursor)
124
125     # Create and import products
126     create_product(num_product=100, cursor=cursor)
127
128     # Create and import employees
129     create_employee(num_employee=100, cursor=cursor)
130
131     # Create and import inventory
132     create_inv(num_inv=100, cursor=cursor)
133
134     # Create and import sales
135     create_sale(num_sales=100, cursor=cursor)
136
137     cnx.commit()
138     cursor.close()
139     cnx.close()
140
141
142 if __name__ == '__main__':
143     import_data_to_mysql()
144

```

Ejecución:

```

C:\Users\jeanp\PycharmProjects\pythonProject\venv\Scripts\python.exe C:\Users\jeanp\PycharmProjects\pythonProject\venv\sec.py

Process finished with exit code 0

```

Validación:

SCHEMAS

Filter objects

FarmaciasCCM

Tables

Empleado

Columns

Indexes

Foreign Keys

Triggers

Inventario

Producto

Sucursal

Venta

Views

Stored Procedures

Functions

sys

1 • SELECT \* FROM FarmaciasCCM.Empleado;

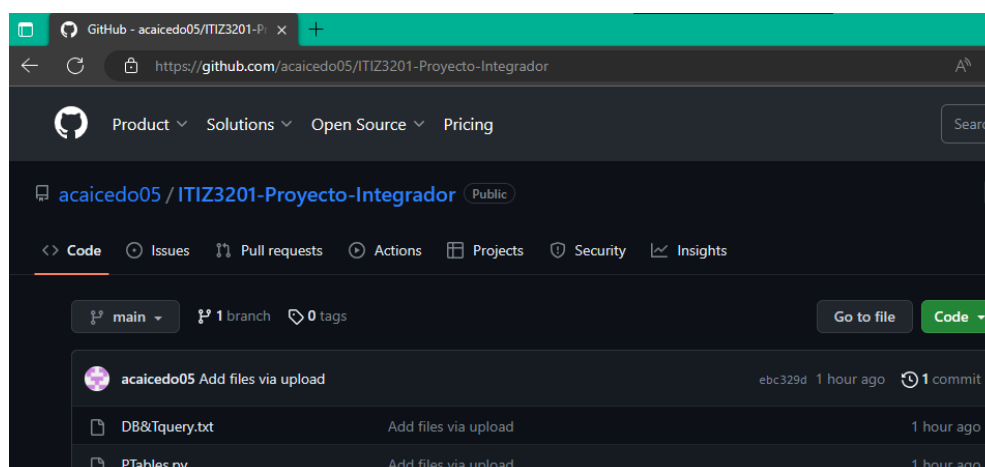
Result Grid

ID_Empleado	Nombre	Apellido	Direccion	Telefono	ID_Sucursal
1	Reyna	Briones	C. Esperanza Rojas 49 Lleida, 51461	Ciudad	1
2	Isidro	Barrena	Pasaje Roberto Martorell 94 Piso 1 Bar...	Mellila	3
3	Aarón	Roma	Vial de Vidal Pons 75 Puerta 4 Zaragoz...	Ourense	1
4	Alfonso	Marco	Glorieta Leonardo Cánovas 61 Puerta 2...	Teruel	2
5	Ramón	Franco	Paseo de Altana Olivares 47 León, 21315	Barcelona	3
6	Luisa	Arana	Rambla de José Calderón 58 Puerta 3 ...	Badajoz	4
7	Sosimo	Navas	Pasadizo Demetrio Pineda 14 Piso 2 Có...	Cáceres	2
8	Tomás	Guzman	Avenida de Mariana Pérez 93 Apt. 78 ...	Ciudad	2
9	Encarnita	Román	Cuesta Daniela Lobo 63 Mellila, 41859	Murcia	2
10	René	Sobrinio	Ronda Eliseo Jerez 1 Santa Cruz de Te...	Jaén	3
11	Ángel	Aparicio	Urbanización de Jenny Antúñez 39 Ála...	Teruel	3
12	Piedad	Marquez	Cuesta de Octavia Seguí 24 Piso 4 Bal...	Sevilla	4
13	Ovidio	Fuster	Alameda Sofia Olivera 754 Valladolid, 1...	Tarragona	1
14	Anita	Iglesia	Vial Merche Sevilla 44 Apt. 05 Guadalaj...	Castellón	4
15	Porfirio	Sosa	Camino de Marino Benavente 4 Cuenca...	Badajoz	2
16	Jesusa	Hoz	Calle Edelmiro Navarrete 85 Murcia, 32...	Segovia	1

## II. Instrucciones para acceder al repositorio en GitHub

Para ingresar al repositorio de GitHub lo que se debe hacer es abrir un navegador, e ingresar el siguiente enlace: <https://github.com/acaicedo05/ITIZ3201-Proyecto-Integrador.git>

Donde se podrá visualizar todos los archivos cargados en el repositorio.

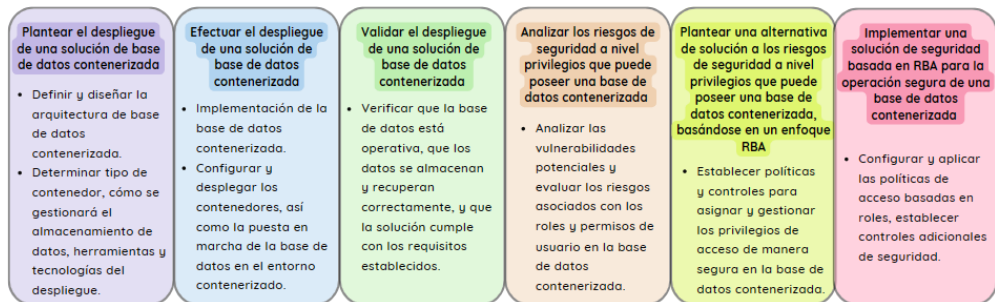


### III. Informe de cumplimiento de los siguientes criterios de desempeño

Criterio	Porcentaje de Cumplimiento	Evidencia(s)
Definición de los Objetivos de Trabajo	100%	Anexo 1: Objetivos
Definición de Cronograma	100%	Anexo 2: Cronograma
Definición de Roles	100%	Anexo 3: Definición y asignación de roles y responsabilidades
Asignación de Roles	100%	
Asignación de Responsabilidades	100%	
Cronograma de reuniones de trabajo	100%	Anexo 4: Cronograma de reuniones
Ideas aportadas para la implementación de la solución por cada participante	100%	Anexo 5: Ideas para Implementación
Aporte individual para la consecución de los Objetivos de Trabajo	100%	Anexo 6: Consecución de objetivos
Aporte individual para la resolución de posibles conflictos	100%	Anexo 7: Resolución de conflictos

## IV. Anexos

### 1. Objetivos



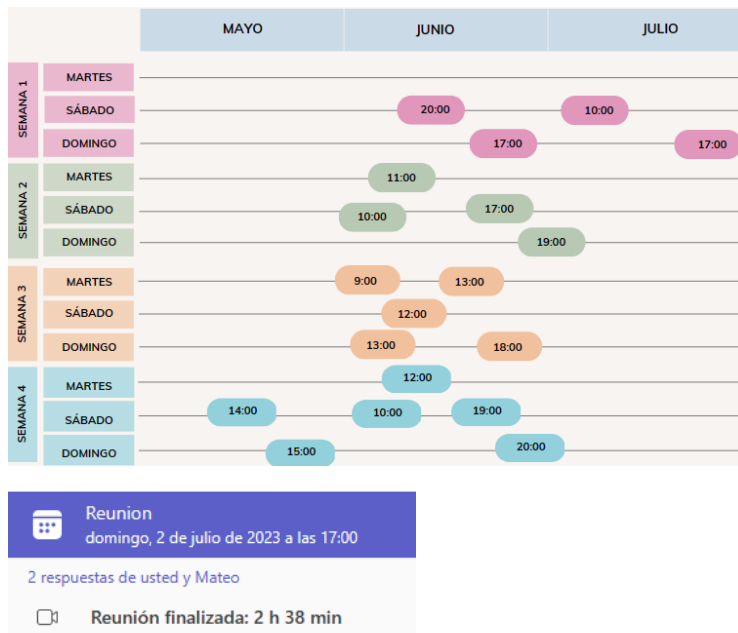
### 2. Cronograma

TAREAS	MAYO	JUNIO	JULIO
Asignación de tema, roles, responsabilidades	●		
Diseño de la base de datos contenerizada	●		
Validación de datos a ingresar	●	○	
Implementación de base de datos contenerizada		○	●
Asignación de seguridad y privilegios		○	●
Finalización del informe			●

### 3. Definición y asignación de roles y responsabilidades

Roles	Responsabilidades	Miembros
Arquitecto de soluciones	Diseñar y definir la estructura de contenedores, seleccionar las tecnologías y establecer mejores prácticas para el despliegue de la base de datos en un entorno contenerizado.	Alejandra Caicedo
Ingeniero de despliegue	Configurar y lanzar los contenedores, asegurarse de que estén correctamente conectados y funcionando adecuadamente, establecer los mecanismos de almacenamiento.	Mateo Cueva Líder del Proyecto
Especialista en seguridad	Evaluar vulnerabilidades potenciales, proponer una estrategia de seguridad adecuada, como la implementación de un enfoque basado en RBA, implementar controles de seguridad para proteger la base de datos contenerizada y garantizar su operación segura.	Jean Moyano

### 4. Cronograma de reuniones



## 5. Ideas para Implementación

Miembros	Ideas para implementación
Mateo Cueva	Asegurar que los archivos de configuración y los de la base de datos residan en un almacenamiento persistente. Se puede usar volúmenes de Docker para montar directorios del host en los contenedores correspondientes. Esto garantizará que los datos y la configuración se conserven incluso si los contenedores se detienen o se reinician.
Alejandra Caicedo	Configurar la base de datos contenerizada que pueda admitir roles de base de datos. Crear un rol de base de datos con privilegios de solo lectura y otro rol con privilegios para manipular los datos. Luego, se procede a crear dos usuarios de base de datos y asigna a cada uno el rol correspondiente. Esto ayudará a garantizar que solo los usuarios autorizados puedan acceder y manipular los datos de la base de datos.
Jean Moyano	Implementar la solución utilizando contenedores Docker para poder asegurar la portabilidad y replicabilidad de la base de datos, ya que cada componente como el servidor de base de datos y el cliente gráfico SQL, pueden ejecutarse en su propio contenedor, lo que facilita la configuración y la gestión.

## 6. Consecución de objetivos

Miembros	Consecución de objetivos
Jean Moyano	Realizar una investigación exhaustiva sobre las buenas prácticas en el despliegue y operación de una base de datos contenerizada. Esto incluye estudiar casos de uso similares, documentación oficial de proveedores de contenedores y bases de datos, así como consultas a expertos en el campo. Esta investigación ayudará a comprender los requisitos técnicos y las mejores prácticas para implementar una solución de base de datos contenerizada de manera efectiva y segura.
Mateo Cueva	Diseñar e implementar un modelo de datos eficiente que cumpla con las necesidades de almacenamiento planteadas por el docente. Esto implica comprender los requisitos funcionales y no funcionales de la farmacia y diseñar una estructura de base de datos adecuada que permita un almacenamiento eficiente y un acceso rápido a los datos. El modelo de datos debe ser flexible y escalable para adaptarse a posibles cambios o expansiones futuras.
Alejandra Caicedo	Realizar una evaluación para identificar posibles vulnerabilidades, amenazas y posibles escenarios de riesgo de seguridad asociados con el acceso a los datos en la base de datos contenerizada. Con base a está, se debe diseñar e implementar una solución de seguridad basada en un enfoque de administración de privilegios basado en roles (RBA). Esta solución garantizará que los usuarios solo tengan los privilegios necesarios para realizar sus tareas asignadas y reducir los riesgos de seguridad asociados con el acceso no autorizado o incorrecto a los datos.

## 7. Resolución de conflictos

Miembros	Resolución de conflictos
Alejandra Caicedo	Las amenazas y las mejores prácticas de seguridad evolucionan constantemente, por lo que es importante mantenerse actualizado y estar al tanto de las últimas recomendaciones y medidas de seguridad. También es importante capacitar al personal involucrado en el manejo de la base de datos en las mejores prácticas de seguridad y concienciarlos sobre las políticas y procedimientos establecidos.
Jean Moyano	Llevar a cabo auditorías regulares de seguridad para identificar posibles vulnerabilidades y detectar actividades sospechosas o no autorizadas. Estas auditorías pueden incluir revisiones de registros de acceso, monitoreo de eventos y análisis de registros de actividad. Al implementar una solución de base de datos contenerizada, se deben utilizar herramientas y técnicas de auditoría adecuadas para evaluar la integridad y la seguridad de la base de datos.
Matteo Cueva	Definir políticas de control de acceso a la base de datos contenerizada. Esto implica establecer roles y privilegios adecuados para cada usuario o rol, limitando el acceso solo a las funciones y datos necesarios para llevar a cabo sus tareas. Además, se deben implementar mecanismos de autenticación y autorización sólidos, como contraseñas seguras, para garantizar que solo los usuarios autorizados puedan acceder a la base de datos.