

INFORME DE LABORATORIO 07 INSTALACIÓN DE UNA INSTANCIA DE MICROSOFT SQL SERVER

Huichi Contreras, Franklin Carlos (2016054948)

Escuela Profesional de Ingeniería de Sistemas

Universidad Privada de Tacna

Tacna, Perú

1. INFORMACIÓN GENERAL

1.1. *Objetivos*

- Instalación de una instancia SQL Server en Docker y conectarse al mismo mediante SQL Server Management Studio

1.2. *Equipos, materiales, programas y recursos utilizados*

- Computadora con sistema operativo Windows XP, Vista, Windows 7, Windows 8 y/o Windows 8.1.
- CPU SLAT-capable feature al menos 4GB de RAM
- Docker Desktop (Para lo cual se debe primero crear una cuenta en Docker Hub)
- Microsoft SQL Server Management Studio en su última versión

2. Marco Teórico

2.1. *Docker*

Docker se define como un proyecto de código abierto que proporciona una capa de abstracción y virtualización a nivel de sistema operativo, a través de la instalación de contenedores de software.

2.1.1. *SQL Server Management Studio*

SQL Server Management Studio (SSMS) es la interfaz de usuario cliente preferida y oficial con la cual se puede manejar, configurar, desplegar, actualizar y administrar una instancia SQL Server.

3. PROCEDIMIENTO

3.0.1. *Paso 1: Gestionar Docker Setup*

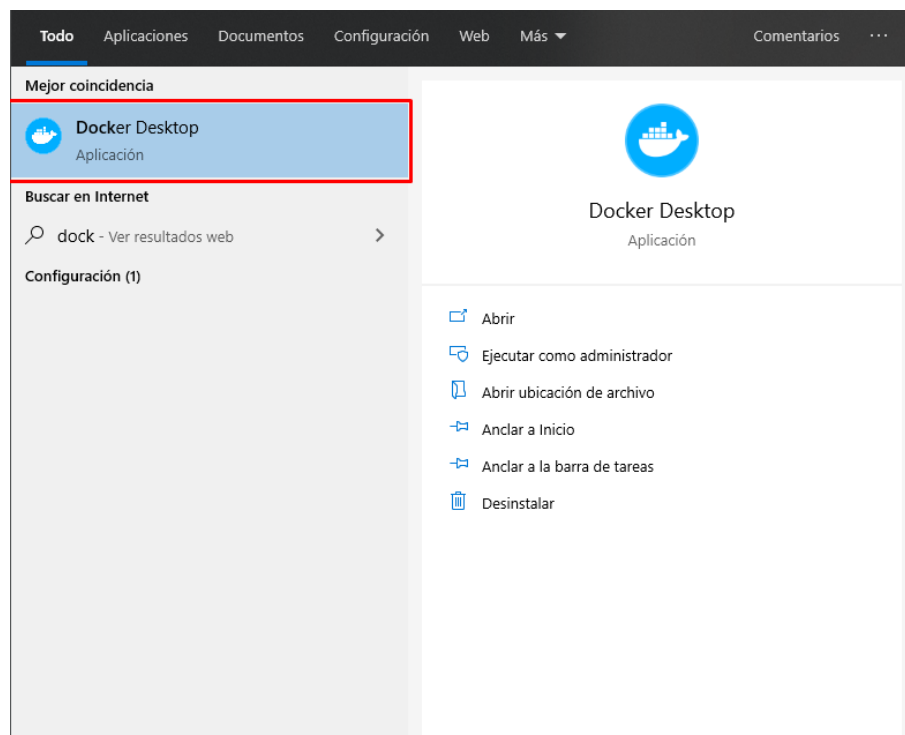


Figura 1: Ingresar a Docker Setup

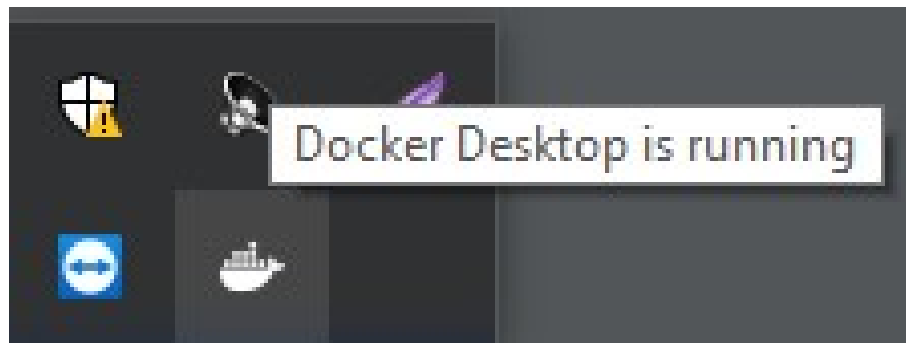


Figura 2: Comprobamos que esta arrancando el Docker

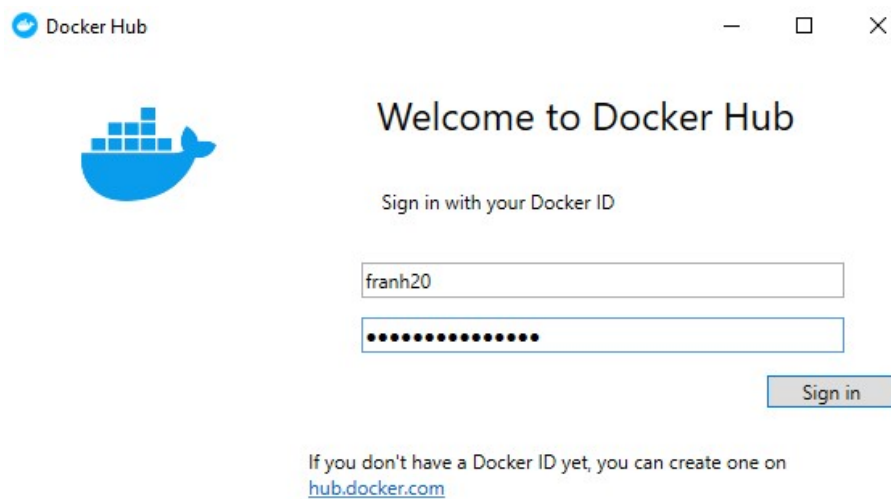


Figura 3: Ingresamos nuestra cuenta de Docker

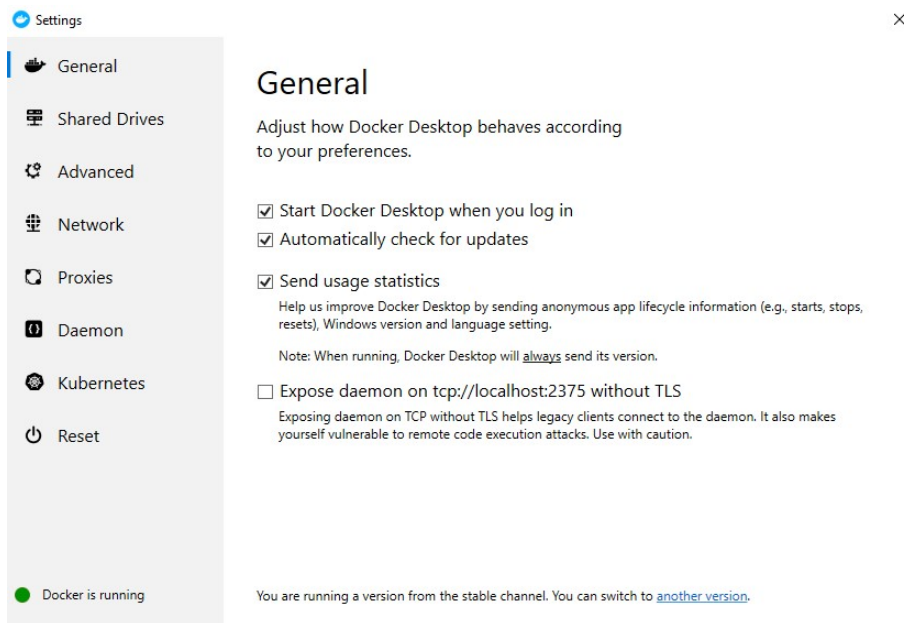


Figura 4: Como se ve podemos ver los ajustes del Docker.

3.0.2. Paso 2: Gestionar los contenedores mediante PowerShell

```
Windows PowerShell
Copyright (C) Microsoft Corporation. Todos los derechos reservados.

Prueba la nueva tecnología PowerShell multiplataforma https://aka.ms/pscore6

PS C:\Users\Usuario> docker version
Client: Docker Engine - Community
Version:          19.03.4
API version:      1.40
Go version:       go1.12.10
Git commit:       9013bf5
Built:           Thu Oct 17 23:44:48 2019
OS/Arch:         windows/amd64
Experimental:     false

Server: Docker Engine - Community
Engine:
Version:          19.03.4
API version:      1.40 (minimum version 1.12)
Go version:       go1.12.10
Git commit:       9013bf5
Built:           Thu Oct 17 23:50:38 2019
OS/Arch:         linux/amd64
Experimental:     false
containerd:
Version:          v1.2.10
GitCommit:       b34a5c8af56e510852c35414db4c1f4fa6172339
runc:
Version:          1.0.0-rc8+dev
GitCommit:       3e425f80a8c931f88e6d94a8c831b9d5aa481657
docker-init:
Version:          0.18.0
GitCommit:       fec3683
PS C:\Users\Usuario>
```

Figura 5: Ingresamos “Docker versión” para ver si tenemos Docker

```
PS C:\Users\Usuario> docker search mssql
```

NAME	OFFICIAL	AUTOMATED	DESCRIPTION	STARS
microsoft/mssql-server-linux			Deprecated SQL Server on Linux Container Rep...	1157
microsoft/mssql-server-windows-developer			Official Microsoft SQL Server Developer Edit...	365
microsoft/mssql-server-windows-express			Official Microsoft SQL Server Express Editio...	329
microsoft/mssql-tools			Official images for Microsoft SQL Server Com...	51
rsmoorthy/mssql			MSSQL Database (version SQL2000)	12
datagrip/mssql-server-linux		[OK]	SQL Server and SQL Server tools on Linux(201...	9
gantrior/mssql-server-2014-express-windows-with-iis		[OK]	mssql 2014 + IIS	4
microsoft/mssql-monitoring-influxdb			Sample Image for Influxdb, This image is des...	3
mcroe/mssqldocker			Builds on microsoft/mssql-server-linux and a...	3
tsgkadot/mssql-tools		[OK]	SQL Server tools on Linux (sqlcmd)	3
ansibleplaybookbundle/mssql-apb		[OK]	MS SQL Server on Linux (APB)	2
bitwarden/mssql		[OK]	The Bitwarden database.	2
mlitellovinx/mssql-server-linux		[OK]	microsoft/mssql-server-linux with mssql-tool...	2
jboesl/mssql-server-linux		[OK]	mssql-server-linux with mssql-tools installe...	2
microsoft/mssql-monitoring-collectd			This Sample image is designed to work with t...	1
ansibleplaybookbundle/mssql-remote-apb		[OK]	An APB that deploys Microsoft SQL Server	1

Figura 6: Buscamos la Iso para descargar

```
Windows PowerShell
PS C:\Users\Usuario> docker pull microsoft/mssql-server-linux
Using default tag: latest
latest: Pulling from microsoft/mssql-server-linux
59ab41dd721a: Downloading [=====] 11.21MB/42.22MB
57da90bec92c: Download complete
06fe57530625: Download complete
5a6315c8a1ff: Waiting
739f58768b3f: Waiting
0b751601bca3: Waiting
bcf04a22644a: Waiting
6b5009e4f470: Waiting
a9dca2f6722a: Waiting
```

Figura 7: Descargamos la iso definida

```
Windows PowerShell
PS C:\Users\Usuario> docker pull microsoft/mssql-server-linux
Using default tag: latest
latest: Pulling from microsoft/mssql-server-linux
59ab41dd721a: Pull complete
57da90bec92c: Pull complete
06fe57530625: Pull complete
5a6315c8a1ff: Pull complete
739f58768b3f: Pull complete
0b751601bca3: Pull complete
bcf04a22644a: Pull complete
6b5009e4f470: Pull complete
a9dca2f6722a: Pull complete
Digest: sha256:9b700672670bb3db4b212e8aef841ca79eb2fce7d5975a5ce35b7129a9b90ec0
Status: Downloaded newer image for microsoft/mssql-server-linux:latest
docker.io/microsoft/mssql-server-linux:latest
PS C:\Users\Usuario>
```

Figura 8: Descargamos la iso definida

```
PS C:\Users\Usuario> docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
microsoft/mssql-server-linux  latest            314918ddaedf       10 months ago      1.35GB
PS C:\Users\Usuario>
```

Figura 9: Revisamos si tenemos descargado la ISO

```
PS C:\Users\Usuario> docker run -e "ACCEPT_EULA=Y" -e "SA_PASSWORD=Epis.2019" `
>> -p 1433:1433 --name SQLLN01 `
>> -d microsoft/mssql-server-linux
```

Figura 10: Instalamos nuestro primer contenedor con MSSQL-Server

```
PS C:\Users\Usuario> docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS              PORTS               NAMES
279452ab7716       microsoft/mssql-server-linux  "/opt/mssql/bin/sqls..."  6 seconds ago      Up 5 seconds       0.0.0.0:1433->1433/tcp  SQLLN01
```

Figura 11: Verificamos que tenemos instalado

3.0.3. Paso 3: Conectarnos a la base de datos mediante Microsoft SQL Server Management Studio

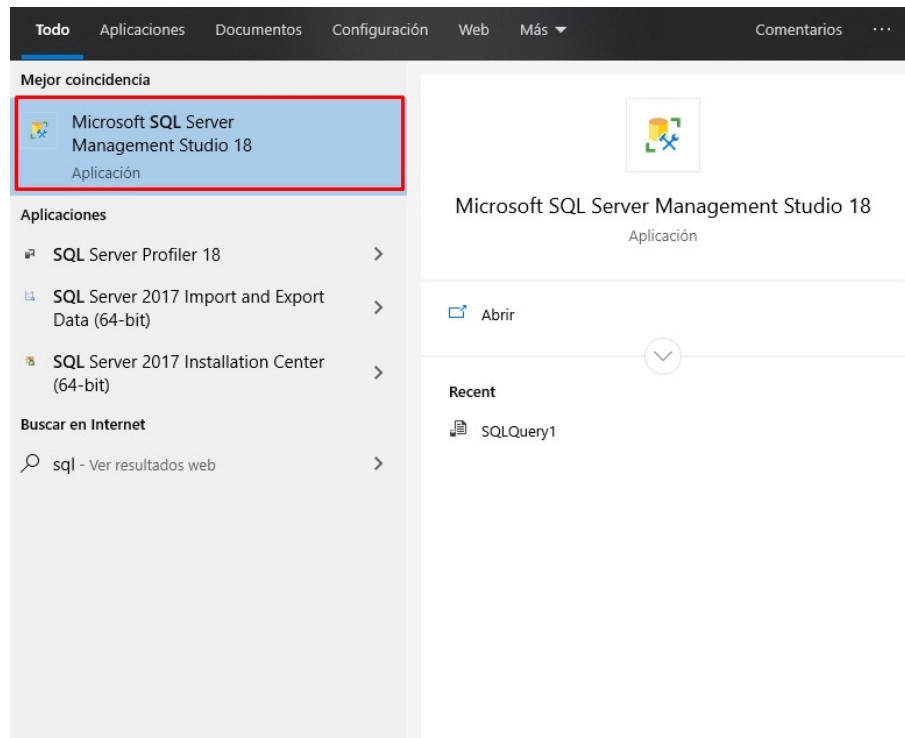


Figura 12: Ingresamos a Management Studio

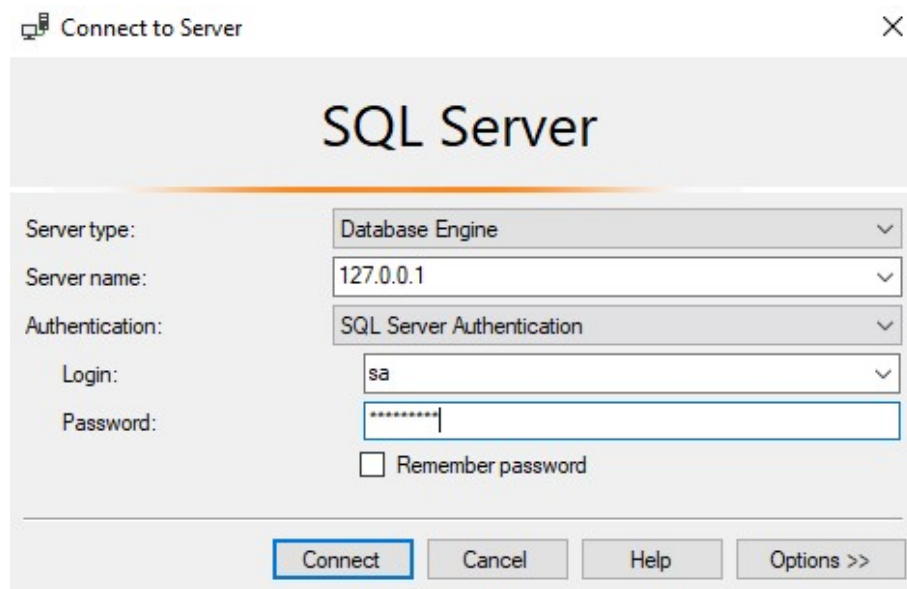


Figura 13: Ingresamos la ip local nuestro cuenta y contraseña establecida al momento de instalación

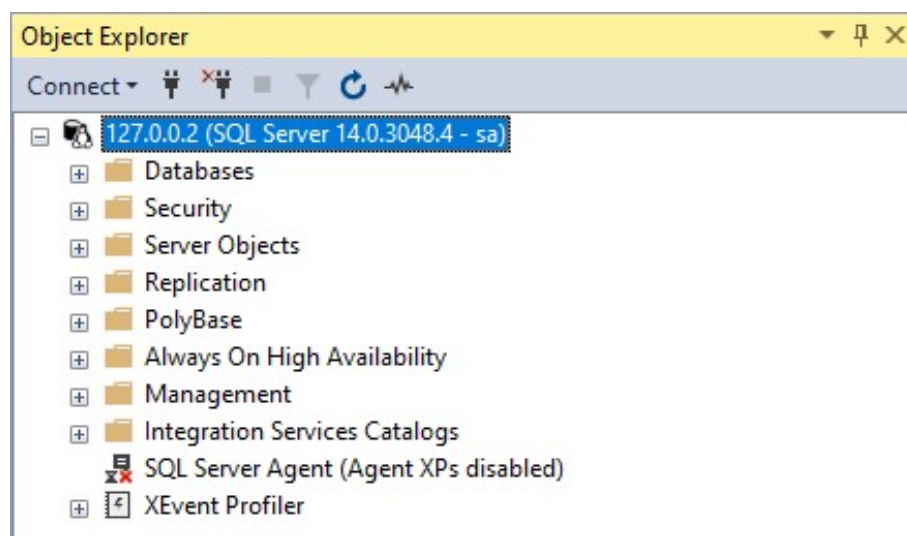


Figura 14: Como podemos visualizar ya nos podemos conectar a nuestra base de datos y visualizar que sale la ID del contenedor

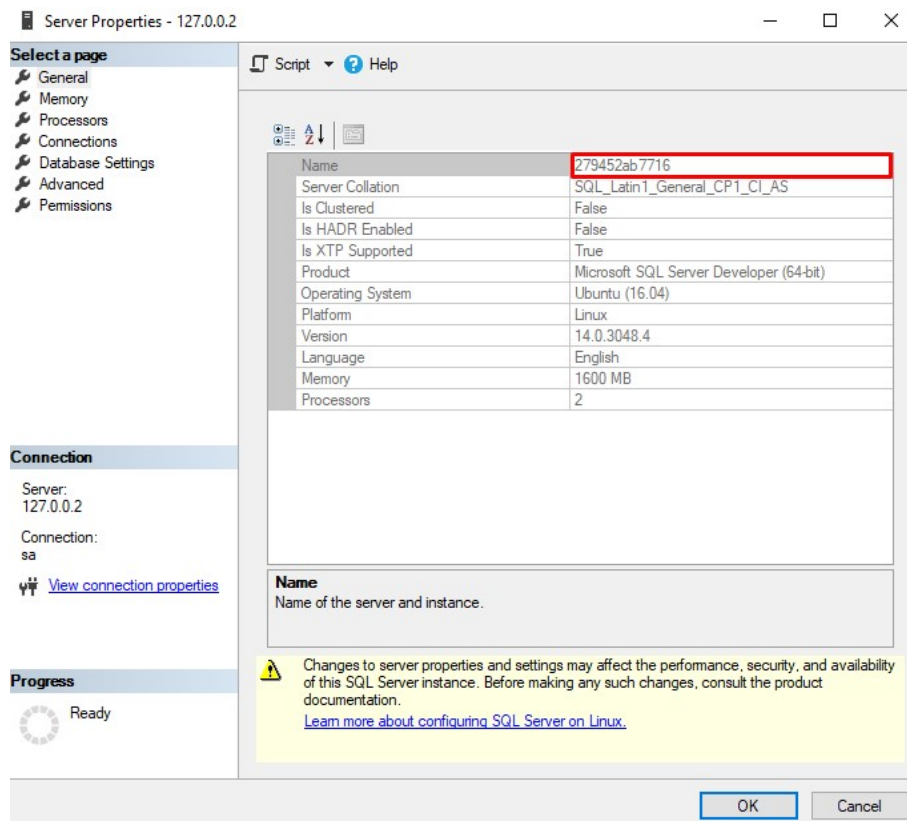


Figura 15: Como podemos visualizar ya nos podemos conectar a nuestra base de datos y visualizar que sale la ID del contenedor

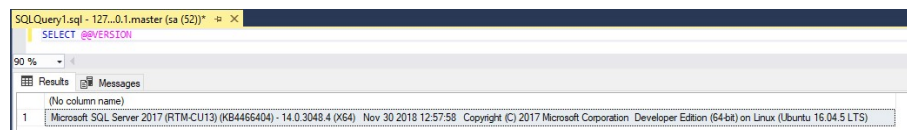


Figura 16: También podemos visualizar nuestra versión

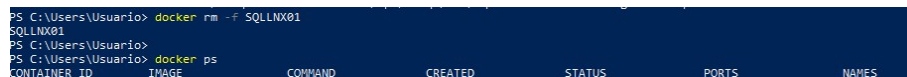


Figura 17: Por último eliminamos nuestra contenedor.

4. ANALISIS E INTERPRETACION DE RESULTADOS

- ¿Qué indican los resultados?
Pudimos realizar exitosamente la conexión de nuestro contenedor a la base de datos
- ¿Que se ha encontrado?
Encontré una manera más rápida de poder tener una base de datos SQL Server sin necesidad de estar haciendo toda la instalación necesaria del MSSQL en mi computadora.

5. CUESTIONARIO

- ¿Con qué comando(s) exportaría la imagen de Docker de Microsoft SQL Server a otra PC o servidor?
Con el comando: `docker push Tunombre/my-first-repo`
- ¿Con qué comando(s) podría generar dos volúmenes para un contenedor para distribuir en un volumen el Archivo de Datos (.mdf) y en otro el Archivo Log (.ldf)?
`docker volume create ArchivodeDatos.mdf`
`docker volume create ArchivodeLog.ldf`
- Genere un nuevo contenedor y cree la base de datos con las siguientes características.
Nombre : FINANCIERA
Archivos :
DATOS (mdf) : Tamaño Inicial : 50MB, Incremento: 10MB, Ilimitado
INDICES (ndf) Tamaño Inicial : 100MB, Incremento: 20MB, Maximo: 1GB
HISTORICO (ndf) Tamaño Inicial : 100MB, Incremento: 50MB, Ilimitado
LOG (ldf) Tamaño Inicial : 10MB, Incremento: 10MB, Ilimitado
¿Cuál sería el script SQL que generaría esta base de datos?

```

CREATE DATABASE FINANCIERA ON
PRIMARY (
    NAME='Datos',
    FILENAME='C:\Program Files\Microsoft SQL Server\MSSQL14.MSSQLSERVER\MSSQL\DATA\Datos.mdf',
    SIZE=50MB,
    FILEGROWTH = 10MB
),
FILEGROUP Indices(
    NAME = 'Indices',
    FILENAME = 'C:\Program Files\Microsoft SQL Server\MSSQL14.MSSQLSERVER\MSSQL\DATA\Indices.ndf',
    SIZE = 100MB,
    FILEGROWTH=20MB,
    MAXSIZE=1GB
),
FILEGROUP Indices(
    NAME = 'Historico',
    FILENAME = 'C:\Program Files\Microsoft SQL Server\MSSQL14.MSSQLSERVER\MSSQL\DATA\Historico.ndf',
    SIZE = 100MB,
    FILEGROWTH=50MB
) LOG ON (
    NAME = 'LOGFinanciera',
    FILENAME = 'C:\Program Files\Microsoft SQL Server\MSSQL14.MSSQLSERVER\MSSQL\Log\LOGFinanciera.ldf',
    SIZE = 10MB ,
    FILEGROWTH = 10MB
);
GO

```

Figura 18: Script planteado

6. CONCLUSION

En conclusión, los contenedores nos ayudan a montar nuestra base de datos de forma mas rápida para poder manejar nuestros diversos sistemas a implementarlos y conectarlos, a su vez también aprendí que los ISO nos vienen a permitir con Docker subirlas para poder usarlas en otras ocasiones.