

UNIVERSIDAD PRIVADA-DE-TACNA



INGENIERIA DE SISTEMAS

TITULO:

**INFORME DE LABORATORIO No 05**

**CURSO:**

BASE DE DATOS II

**DOCENTE(ING):**

Patrick Cuadros Quiroga

Integrantes:

Arlyn Cotrado Coaquira	(2016054466)
Yaneth Virginia Aquino Huallpa	(2017059286)
Sharon Sosa Bedoya	(2016054460)
Marlon Villegas Arando	(2015053890)

# Índice

<b>1. INFORMACIÓN GENERAL</b>	<b>1</b>
1.1. Objetivos: . . . . .	1
1.2. Equipos, materiales, programas y recursos utilizados: . . . . .	1
<b>2. MARCO TEORICO</b>	<b>2</b>
2.1. Docker: . . . . .	2
2.2. Oracle Database en Docker: . . . . .	2
2.3. Referencias de cómo usar Oracle con Docker en Linux Y en Windows: . . . . .	2
2.4. Construir la imagen: . . . . .	2
<b>3. PROCEDIMIENTO</b>	<b>4</b>
3.1. Parte 1: Iniciando Docker . . . . .	4
3.2. Parte 2: Creando un contenedor con Oracle Database para Linux . . . . .	4
3.3. Parte 3: Adicionando persistencia . . . . .	8
<b>4. ANALISIS E INTERPRETACION DE RESULTADOS</b>	<b>10</b>
4.1. Parte 1: Actividades Encargadas . . . . .	10
<b>5. CONCLUSIONES</b>	<b>11</b>
<b>6. REFERENCIAS</b>	<b>12</b>

# **1. INFORMACIÓN GENERAL**

## **1.1. Objetivos:**

- Conocer los fundamentos sobre contenedores y Docker.
- Poder instalar correctamente una instancia.

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

- Virtualización activada en el BIOS.
- Windows 10 64bit: Pro, Enterprise o Education, con al menos 4GB de RAM.
- Docker Desktop
- Microsoft SQL Server 2017 o superior

## **2. MARCO TEORICO**

### **2.1. Docker:**

- Tener un docker que provea el gestor de base de datos es muy útil porque se reducen tiempos de instalación y configuración y en caso de tener un error muy grave en la configuración es tan sencillo resolverlo como borrar el contenedor y crear uno nuevo.
- Los contenedores funcionan bien para desarrollo y tal vez algunos ambientes de evaluación para el cliente, pero para ambientes productivos para nada se recomiendan, en estos casos siempre será lo mejor que se cuente con una base de datos instalada en el servidor.
- Sirven para desplegar aplicaciones en un entorno virtual aislado, pero sin el overhead de tener un Sistema Operativo (SO) nuevo como se tiene en una Virtual Machine (VM).

### **2.2. Oracle Database en Docker:**

- Los productos de Oracle son compatibles con Docker si el sistema operativo del host es Oracle Linux 7, pero no necesita usar un host OL7 para que esto funcione. Puedes ver cómo instalar Docker en OL7 .
- Usar imágenes de Oracle Container Registry o de Docker Store tiene la ventaja que los binarios de instalación vienen incluidos, lo que no es permitido por licencia en el resto de las distribuciones.

### **2.3. Referencias de cómo usar Oracle con Docker en Linux Y en Windows:**

- Docker en Windows 10:
- Para usar la versión completa es necesario habilitar Microsoft Hyper-V, lo que implica deshabilitar la virtualización por hardware de nuestro PC. Si estamos usando VirtualBox en el mismo host, con este cambio deja de funcionar. Docker Toolbox no tiene esta restricción, aunque se mantiene como una versión antigua (Legacy), y Docker recomienda usar la versión completa. Otra diferencia de Docker Toolbox es que necesita una VM VirtualBox para ejecutar. Esta VM se crea de forma automática al usar Toolbox, de nombre default, y se usa como host para los containers que creemos.

### **2.4. Construir la imagen:**

- La compilación espera el siguiente sistema de archivos. Tendrá que descargar la base de datos Oracle 19c y el software APEX usted mismo y colocarlo en el directorio "software".

```
$ arbol
.
├─ Dockerfile
├─ README.md
├─ guiones
│  └─ healthcheck.sh
├─ ┌─ start.sh
└─ software
   ├── apex_19.1_en.zip
   ├── LINUX.X64_193000_db_home.zip
   └─ put_software_here.txt

PS
```

## 3. PROCEDIMIENTO

### 3.1. Parte 1: Iniciando Docker

- Abrir el menu inicio y buscar la aplicación Docker for Windows.
- Ubicar la aplicación PowerShell, ejecutarla como Administrador. En la ventana de comandos de PowerShell escribir lo siguiente.

```
PS C:\> docker version
Client: Docker Engine - Community
 Version:      18.09.2
 API version:  1.39
 Go version:   go1.10.8
 Git commit:   6247962
 Built:        Sun Feb 10 04:12:31 2019
 OS/Arch:      windows/amd64
 Experimental: false

Server: Docker Engine - Community
 Engine:
  Version:      18.09.2
  API version:  1.39 (minimum version 1.24)
  Go version:   go1.10.6
  Git commit:   6247962
  Built:        Sun Feb 10 04:28:48 2019
  OS/Arch:      windows/amd64
  Experimental: false
```

### 3.2. Parte 2: Creando un contenedor con Oracle Database para Linux

- En un navegador de internet acceder a la dirección <https://hub.docker.com/>. Iniciar sesión o crear una cuenta nueva
- Buscar el repositorio para Oracle Database. Ingresar y proceder con el CheckOut, completar los datos y aceptar las condiciones obligatorias para obtener el acceso al contenido.



- En la ventana de PowerShell, escribir el siguiente comando:

```
PS C:\> docker login
Authenticating with existing credentials...
Login Succeeded
```

- Ejecutar el siguiente comando en Powershell, lo cual descargará la imagen del contenedor de Oracle Database en un servidor Linux

```
PS C:\> docker pull store/oracle/database-enterprise:12.2.0.1
12.2.0.1: Pulling from store/oracle/database-enterprise
4ce27fe12c04: Pull complete
9d3556e8e792: Pull complete
fc60a1a28025: Pull complete
0c32e4ed872e: Pull complete
b465d9b6e399: Pull complete
Digest: sha256:40760ac70dba2c4c70d0c542e42e082e8b04d9040d91688d63f728af764a2f5d
Status: Downloaded newer image for store/oracle/database-enterprise:12.2.0.1
```

- Seguidamente ejecutar el comando, como respuesta se visualizará un ID que corresponde al contenedor.

```
PS C:\> docker run -d -it --name ORACLEDB01 -p 1521:1521 -p 5500:5500 store/oracle/database-enterprise:12.2.0.1
79c978c3eda27e6ee1f4506313fd66c970827f60993dc54e0f9c4bfebe47ce02
```

- Verificar que el contenedor se esté ejecutando correctamente mediante el comando:

```
PS C:\> docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATE
79c978c3eda2	store/oracle/database-enterprise:12.2.0.1	"/bin/sh -c '/bin/ba..."	7 minu

```
tes ago Up 7 minutes (healthy) 0.0.0.0:1521->1521/tcp, 0.0.0.0:5500->5500/tcp ORACLEDB01
```

- Cuando el estado del contenedor sea “healthy”, en la consola de Powershell, ejecutar el siguiente comando:

```
PS C:\> docker exec -it ORACLEDB01 bash -c "source /home/oracle/.bashrc; sqlplus / as sysdba"
SQL*Plus: Release 12.2.0.1.0 Production on Fri May 31 21:40:55 2019
Copyright (c) 1982, 2016, Oracle. All rights reserved.
Connected to:
Oracle Database 12c Enterprise Edition Release 12.2.0.1.0 - 64bit Production
SQL>
```

- En la línea de comentarios de SQL\*Plus, escribir lo siguiente

```

Administrador: Windows PowerShell

OWNER
-----
TABLE_NAME
-----
TABLESPACE_NAME
-----
CLUSTER_NAME
-----
IOT_NAME
-----
STATUS      PCT_FREE    PCT_USED    INI_TRANS    MAX_TRANS    INITIAL_EXTENT    NEXT_EXTENT
-----
IN_EXTENTS  MAX_EXTENTS PCT_INCREASE FREELISTS    FREELIST_GROUPS LOG B    NUM_ROWS
-----
BLOCKS      EMPTY_BLOCKS    AVG_SPACE    CHAIN_CNT    AVG_ROW_LEN
-----
AVG_SPACE_FREELIST_BLOCKS    NUM_FREELIST_BLOCKS
-----
DEGREE
-----
INSTANCES
-----
CACHE      TABLE_LO
-----
SAMPLE_SIZE LAST_ANAL PAR IOT_TYPE T S NES BUFFER_FLASH_C CELL_FL ROW_MOVE
-----
CLO USE DURATION SKIP_COR MON
-----
CLUSTER_OWNER
-----
DEPENDEN COMPRESS COMPRESS_FOR DRO REA SEG RESULT_ CLU
-----
CTIVITY_TRACKING DDL_TIMESTAMP HAS CON INMEMORY INMEMORY
-----
INMEMORY_DISTRI INMEMORY_COMPRESS INMEMORY_DUPL
-----
DEFAULT_COLLATION
-----
S EXT CELLMEMORY CON CON EXT EXT INMEMORY_SER
-----
INMEMORY_SERVICE_NAME
-----
CON
---

572 rows selected.
                                est of line ignored.
SQL> SQL> SQL> SP2-0042: unknown command "sadr" - rest of line ignored.
SQL> SQL>

```

- Escribir el comando quit para cerrar la sesión de SQL\*Plus

```

SQL> exit
Disconnected from Oracle Database 12c Enterprise Edition Release 12.2.0.1.0 - 64bit Production
PS C:\>

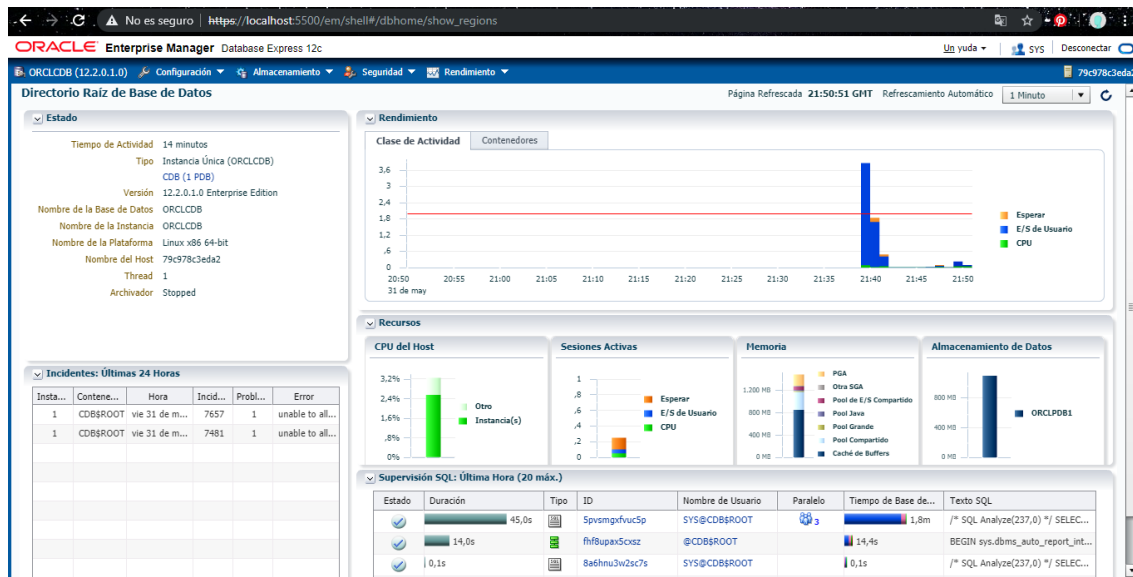
```

- En una pestaña nueva del navegador de internet acceder a la siguiente dirección: <https://localhost:5500/em>  
Iniciar sesión con los siguientes datos:

Usuario: sys  
 Contraseña: Oradoc\_db1  
 Marcar check como SYSDBA

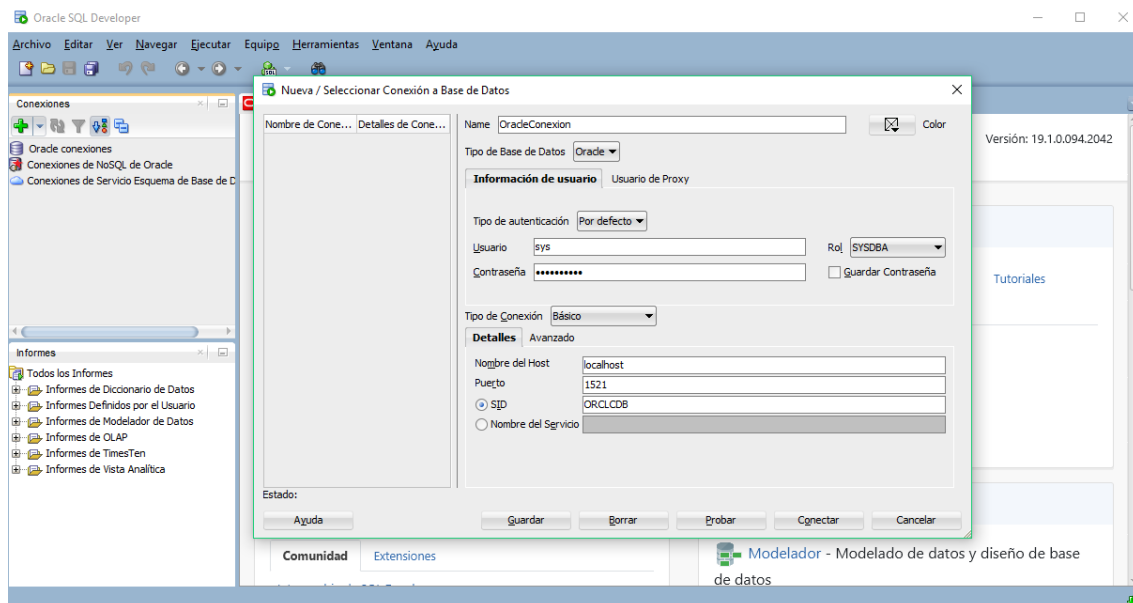


- Luego se visualizará la siguiente ventana. Cerrar sesión y la pestaña del navegador de internet.



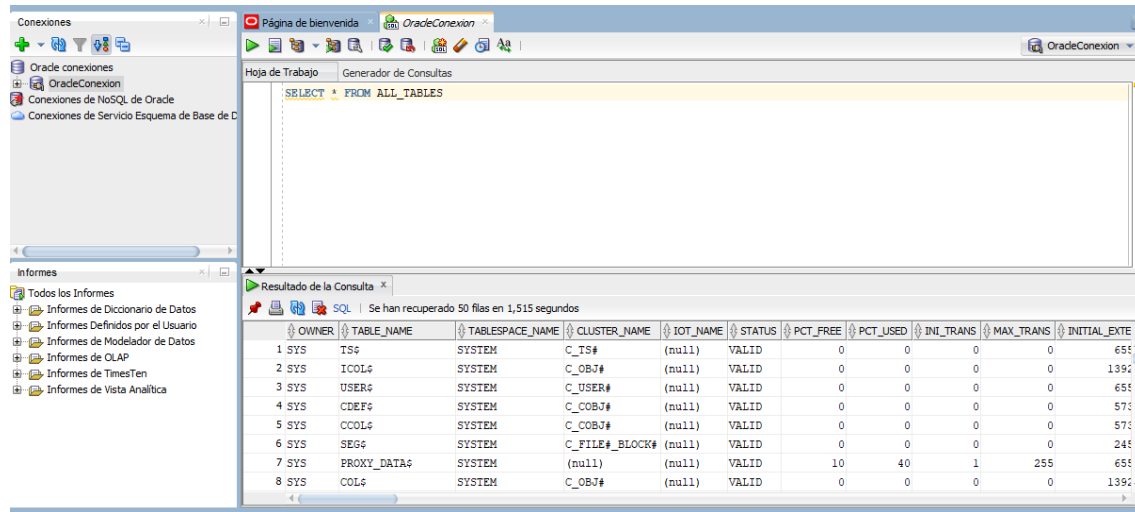
- Iniciar el aplicativo Oracle SQL Developer, crear una nueva conexión con los siguientes parámetros:

**Name : OracleConexion**  
**Usuario : sys**  
**Contraseña : Oradoc\_db1**  
**Rol : SYSDBA**  
**Nombre del Host : localhost**  
**Puerto : 1521**  
**SID : ORCLCDB**



- Iniciar una nueva consulta, escribir y ejecutar lo siguiente; deberá retornar varios registros

que representan las tablas de las base de datos



- Cerrar la aplicación Oracle SQL Developer
- En PowerShell ejecutar el siguiente comando. Y verificar la eliminación del contenedor con ejecutando

```
PS C:\> docker rm -f ORACLEDB01
ORACLEDB01
PS C:\> docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS
PORTS
NAMES
```

### 3.3. Parte 3: Adicionando persistencia

- En PowerShell ejecutar el siguiente comando, lo cual dara como respuesta se visualizará un ID que corresponde al contenedor

```
PS C:\> docker run -d -it --name ORACLEDB01 -p 1521:1521 -p 5500:5500 -v D:\ORCL:/ORCL store/oracle/database-enterprise:12.2.0.1
8f80d3a62c7497bcd62d9216d7312551557611ec52e5f95f429b926af99e5312
```

- Repetir el paso 13 y modificar la contraseña del usuario SYS

```
PS C:\> docker exec -it ORACLEDB01 bash -c "source /home/oracle/.bashrc; sqlplus / as sysdba"

SQL*Plus: Release 12.2.0.1.0 Production on Fri Jun 7 20:00:26 2019

Copyright (c) 1982, 2016, Oracle. All rights reserved.

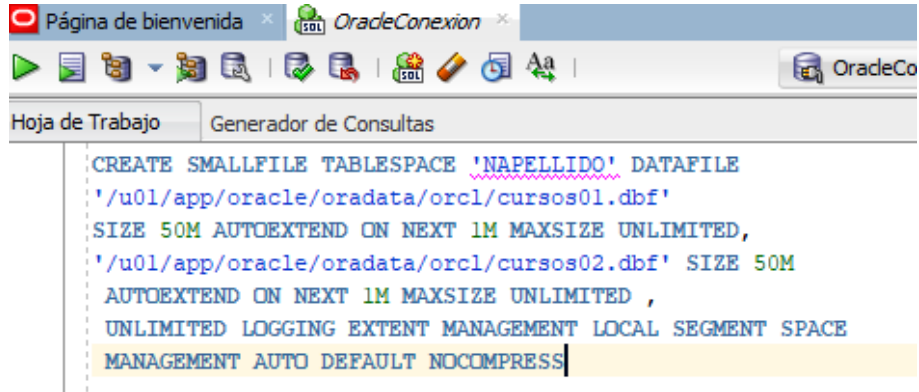
Connected to:
Oracle Database 12c Enterprise Edition Release 12.2.0.1.0 - 64bit Production

SQL> alter user sys identified by 123;

User altered.

SQL> quit
Disconnected from Oracle Database 12c Enterprise Edition Release 12.2.0.1.0 - 64bit Production
PS C:\>
```

- Iniciar el aplicativo Oracle SQL Developer, conectarse como el usuario SYS y ejecutar el siguiente comando



The screenshot shows the Oracle SQL Developer interface. The 'Generador de Consultas' (Query Generator) tab is active, displaying a SQL script to create a smallfile tablespace named 'NAPELLIDO'. The script is as follows:

```
CREATE SMALLFILE TABLESPACE 'NAPELLIDO' DATAFILE
'/u01/app/oracle/oradata/orcl/cursos01.dbf'
SIZE 50M AUTOEXTEND ON NEXT 1M MAXSIZE UNLIMITED,
'/u01/app/oracle/oradata/orcl/cursos02.dbf' SIZE 50M
AUTOEXTEND ON NEXT 1M MAXSIZE UNLIMITED ,
UNLIMITED LOGGING EXTENT MANAGEMENT LOCAL SEGMENT SPACE
MANAGEMENT AUTO DEFAULT NOCOMPRESS
```

- Verificar el contenido de la carpeta ORCL
- En PowerShell ejecutar el siguiente comando. Verificar la eliminación del contenedor con ejecutando



The screenshot shows a PowerShell terminal window. The user has executed the command `docker rm -f ORACLEDB01` to remove the container. Subsequently, the user runs `docker ps` to verify the removal. The output of `docker ps` is shown as a table with columns: CONTAINER ID, IMAGE, COMMAND, CREATED, STATUS, and PORTS.

```
PS C:\> docker rm -f ORACLEDB01
ORACLEDB01
PS C:\> docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS        PORTS
NAME
PS C:\>
```

- Cerrar la aplicación Oracle SQL Developer.

## 4. ANALISIS E INTERPRETACION DE RESULTADOS

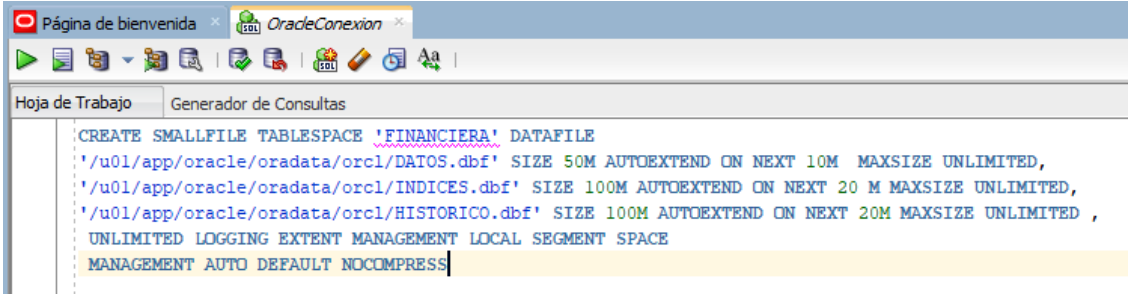
### 4.1. Parte 1: Actividades Encargadas

- ¿Con qué comando(s) puedo iniciar y detener una instancia de Oracle, detalle cada uno de los pasos y opciones, utilizando Docker?
- ¿Con qué comando(s) puedo iniciar y detener el Listener y el Enterprise manager, detalle cada uno de los pasos y opciones, utilizando Docker?
- Genere un nuevo contenedor y cree un espacio de tablas con las siguientes características.

Nombre : FINANCIERA:

- DATOS (dbf) : Tamaño Inicial : 50MB, Incremento: 10MB, Ilimitado
- INDICES (dbf) Tamaño Inicial : 100MB, Incremento: 20MB, Maximo: 1GB
- HISTORICO (dbf) Tamaño Inicial : 100MB, Incremento: 50MB, Ilimitado

¿Cuál sería el script SQL que generaría esta base de datos?



```
CREATE SMALLFILE TABLESPACE 'FINANCIERA' DATAFILE
'/u01/app/oracle/oradata/orcl/DATOS.dbf' SIZE 50M AUTOEXTEND ON NEXT 10M MAXSIZE UNLIMITED,
'/u01/app/oracle/oradata/orcl/INDICES.dbf' SIZE 100M AUTOEXTEND ON NEXT 20 M MAXSIZE UNLIMITED,
'/u01/app/oracle/oradata/orcl/HISTORICO.dbf' SIZE 100M AUTOEXTEND ON NEXT 20M MAXSIZE UNLIMITED ,
UNLIMITED LOGGING EXTENT MANAGEMENT LOCAL SEGMENT SPACE
MANAGEMENT AUTO DEFAULT NOCOMPRESS
```

## 5. CONCLUSIONES

- En conclusión hemos observado y experimentado con docker, y nos resulta que es muy útil al momento de instalar multiples bases de datos y que no existe la necesidad de armar o instalar múltipler ordenadores físicos o virtuales.
- Es por eso que resulta factible en muchos aspectos como migrar de version, tener varias bases de datos disponibles o además que existieran y comparen diferentes versiones de bases de datos a la vez

## 6. REFERENCIAS

- [ 1 ] Hat, R. (2017). ¿Qué es Docker?. Recuperado de <https://www.redhat.com/es/topics/containers/what-is-docker>
- [ 2 ] código chido. (2019). Docker Oracle. Recuperado de <https://codigochido.com/post/2019-01-21-docker-oracle/>
- [ 3 ] Nelson, C. (2018). Usando Oracle 12c en Docker sobre Windows 10. Recuperado de <https://www.docker-win10-4485487-esa.html>
- [ 4 ] The ORACLE-BASE Blog. (2018). Oracle Database en Docker. Recuperado de <https://oracle-base.com/articles/linux/docker-oracle-database-on-docker>