

Universidad Pontificia Bolivariana

Tópicos Avanzados en Bases de Datos

Luisa María Flórez Múnera ID 000449529 – Samuel Pérez Hurtado ID 000459067

Guía de Abastecimiento de la Infraestructura

Examen 1 - Servicios de Domicilios en el Valle de Aburrá

En el desarrollo del proyecto, se ha tomado la decisión de utilizar Docker como infraestructura principal para implementar y gestionar los motores de bases de datos Oracle 21 XE y PostgreSQL. Esta elección de estructura permitirá una administración más eficiente de los entornos de bases de datos, así como la flexibilidad para desplegar y gestionar múltiples sistemas de bases de datos de manera independiente y reproducible.

Docker se ha convertido en una herramienta fundamental para el desarrollo y despliegue de aplicaciones debido a su capacidad para crear y gestionar contenedores. En este proyecto, aprovecharemos esta tecnología para encapsular tanto el motor de base de datos Oracle como Postgres en entornos aislados. Esto no solo facilitará la gestión de las bases de datos, sino que también asegurará la portabilidad entre diferentes sistemas y ambientes.

La elección de utilizar Oracle en contenedores Docker permitirá tener un entorno de base de datos Oracle ligero y fácilmente escalable. Docker nos brinda la posibilidad de definir las configuraciones necesarias para el motor de base de datos en archivos de configuración, lo que asegura una consistencia en la configuración y evita problemas de configuraciones incompatibles entre distintos ambientes. Además, Docker facilita la creación de instantáneas de la base de datos para respaldos y recuperación.

De manera similar, Postgres también será implementado en contenedores Docker. Esto ofrecerá ventajas como la rápida creación de entornos de desarrollo, pruebas y producción consistentes. Los contenedores de Postgres pueden ser fácilmente replicados en diferentes etapas del ciclo de vida del proyecto, garantizando una coherencia en la configuración y los datos en cada etapa.

Nuestra guía de abastecimiento se centrará en la instalación de los motores de bases de datos en MacOS, sin embargo es importante precisar que para Windows se deben seguir unos pasos adicionales:

1. Activar la virtualización en la BIOS y en las características de Windows
2. Instalar el componente WSL y una distribución Linux
3. Es recomendado instalar Powershell en lugar de usar CMD

PostgreSQL – Realizado por Luisa

En MacOS es necesario únicamente instalar Docker Desktop, para la creación del contenedor de Postgres se debe poner el comando

docker pull postgres:latest

Esto con el fin de descargar la imagen del contenedor

```
(base) luisaflorezm@luisaflorezm ~ % docker pull postgres:latest
latest: Pulling from library/postgres
90524f7dc01b: Pull complete
ef7240014a47: Pull complete
e4d2c856d784: Pull complete
885c9ba4e584: Pull complete
7f003c5d3c77: Pull complete
9c86f7fa2874: Pull complete
fa86bc459ff8: Pull complete
5bd6da0c1c54: Pull complete
0bd6bd462f7c: Pull complete
470986c63b92: Pull complete
703ecfe64e45: Pull complete
d1e8339d5d0b: Pull complete
412759686ffd: Pull complete
Digest: sha256:8775adb39f0db45cf4cdb3601380312ee5e9c4f53af0f89b7dc5cd4c9a78e4e8
Status: Downloaded newer image for postgres:latest
docker.io/library/postgres:latest
```

Una vez descargada la imagen, procedemos a crear el contenedor con el comando

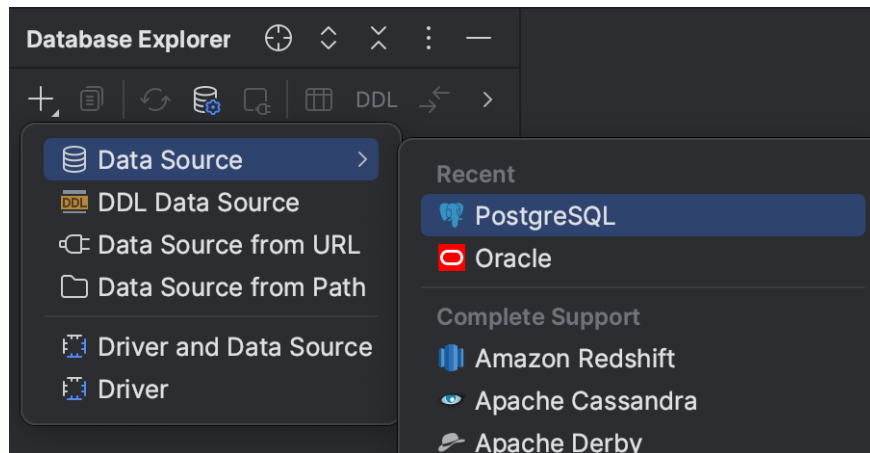
docker run --name servicios-domicilios -e

POSTGRES_PASSWORD=domicilios -d -p 5432:5432 postgres:latest

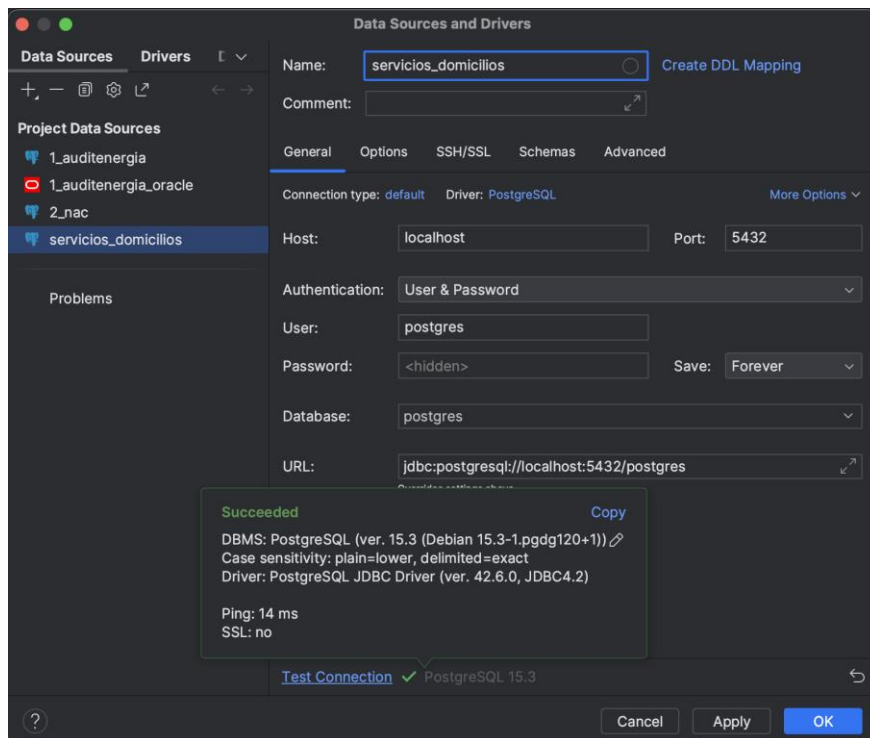
Esto significa que nuestro contenedor se llamará servicios-domicilios, la contraseña será domicilios, el puerto de acceso local es el 5432 y la imagen es postgres

```
(base) luisaflorezm@luisaflorezm ~ % docker run --name servicios-domicilios -e
POSTGRES_PASSWORD=domicilios -d -p 5432:5432 postgres:latest
0ac1d0b32b7d2d8b6ced7fcf566665406011596d7476151ec452bf36e511805f
```

Una vez creado el contenedor, podemos acceder a este con permisos administrativos directamente desde un IDE como DataGrip, debemos hacer la conexión con la fuente de datos, que en este caso es PostgreSQL



Luego, se debe ingresar el host (en este caso localhost), el puerto local especificado al momento de crear el contenedor, el usuario administrativo postgres y la contraseña domicilios, también hay que especificar la base de datos a conectarse, en este caso es la que trae por defecto postgres, pero más adelante para el desarrollo del ejercicio y siguiendo el script adjunto, se creará una base de datos específica y un usuario con los permisos mínimos requeridos



Oracle XE 21c – Realizado por Samuel

Para crear un contenedor de Oracle en MacOS es necesario descargar la imagen, esto se hace con el comando

```
docker pull gvenzl/oracle-xe:latest
```

Esto con el fin de descargar la imagen del contenedor de Oracle XE 21c

```
(base) samuelperezh@samuelperezh22 ~ % docker pull gvenzl/oracle-xe:latest
latest: Pulling from gvenzl/oracle-xe
Digest: sha256:560ea1e99b1e4e2abf02f6bf91d9cc5d63a54466af4a4bbebf8d28d2545c9a2d
Status: Image is up to date for gvenzl/oracle-xe:latest
docker.io/gvenzl/oracle-xe:latest
```

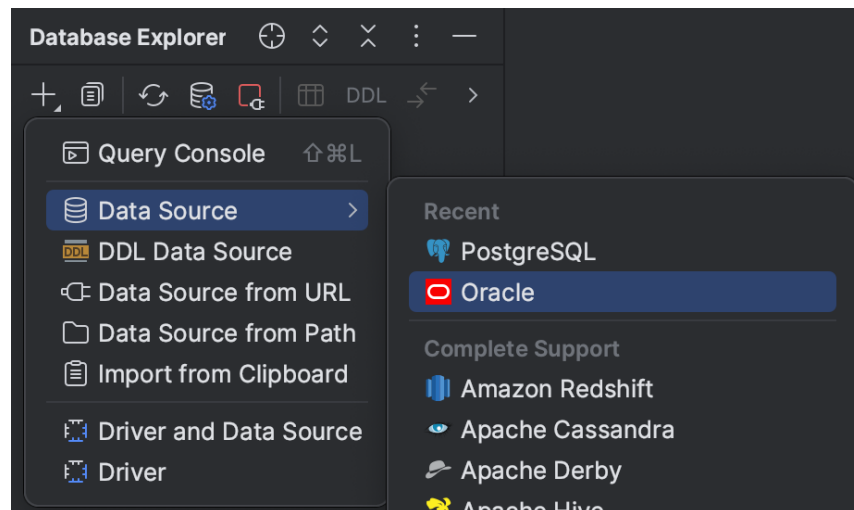
Una vez descargada la imagen, procedemos a crear el contenedor con el comando

```
docker run --name servicios-domicilios-oraxe -d -p 1521:1521 -e
ORACLE_PASSWORD=domicilios gvenzl/oracle-xe:latest
```

Esto significa que nuestro contenedor se llamará servicios-domicilios-oraxe, la contraseña será domicilios, el puerto de acceso local es el 1521 y la imagen es gvenzl/oracle-xe:latest

```
(base) samuelperezh@samuelperezh22 ~ % docker run --name servicios-domicilios-oraxe -d -p 1521:1521 -e ORACLE_PASSWORD=domicilios gvenzl/oracle-xe:latest
4578a89b115d404160b781d1a7fdd87ba8476625a60625d58bb148cab3709672
```

Una vez creado el contenedor, podemos acceder a este con permisos administrativos directamente desde un IDE como DataGrip, debemos hacer la conexión con la fuente de datos, que en este caso es Oracle



Se debe cambiar el connection type a Service Name porque así funciona en esta imagen del motor de bases de datos

Connection type: **SID** Driver: **Oracle** supports since 11.2

Host: **SID**

SID: **Service Name**

Authentication: **TNS**

Long

URL only

Word

Luego, se debe ingresar el host (en este caso localhost), el puerto local especificado al momento de crear el contenedor, el nombre del servicio que en este caso es xepdb1, el usuario administrativo system y la contraseña domicilios, pero más adelante para el desarrollo del ejercicio y siguiendo el script adjunto, se creará una base de datos específica y un usuario con los permisos mínimos requeridos

Data Sources and Drivers

Data Sources **Drivers**

Project Data Sources

- 1_auditenergia
- 1_auditenergia_oracle
- 2_nac
- servicios_domicilios_oraxe**
- servicios_domicilios

Problems

Name: **servicios_domicilios_oraxe** **Create DDL Mapping**

Comment:

General **Options** **SSH/SSL** **Schemas** **Advanced**

Connection type: **Service Name** **Driver:** **Oracle** supports since 11.2 **More Options**

Host: **localhost** **Port:** **1521**

Service: **xepdb1** **Driver:** **Thin**

Authentication: **User & Password**

User: **system**

Password: **<hidden>** **Save:** **Forever**

URL: **jdbc:oracle:thin:@//localhost:1521/xepdb1**

Overrides settings above

Succeeded **Copy**

DBMS: Oracle (ver. Oracle Database 21c Express Edition Release 21.0.0.0.0 - Production)

Version 21.3.0.0.0)

Case sensitivity: plain=upper, delimited=exact

Driver: Oracle JDBC driver (ver. 21.5.0.0.0, JDBC4.3)

Ping: 87 ms

Test Connection **✓** **Oracle 21.0.0.0.0**

Cancel **Apply** **OK**