**DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN Unidad 1 INFORME DE TRABAJO AUTÓNOMO N.°**

**Informe SII 202251 002**

**Página:** 0 de .6.



UNIVERSIDAD DE LAS FUERZASARMADAS

ESPE

**SISTEMA AVANZADO DE BASE DE DATOS**

**TEMA: EXAMEN UNIDAD UNO**

**ESTUDIANTE:** ANDRANGO CARCHIPULLA CHRISTIAN JHONATAN

**DOCENTE:** ING. ALEXIS DARIO ESTEVEZ SALAZAR

NRC: 15031

*SANGOLQUI*

*Noviembre 2023 - Marzo 2024*

**A. OBJETIVOS**

* Implementar un diseño de base de datos en el que muestre el modelamiento y sus replicación en la base de datos de PostgreSQL.
* Detallar cada una de las implementaciones que se abordó en cada de los ítems planteados en la actividad.

**B. DESARROLLO**

**Planteamiento**

Una biblioteca desea implementar un sistema de gestión de información para organizar sus libros, autores, géneros y clientes. El sistema debe permitir a los bibliotecarios registrar nuevos libros, asignar autores y géneros a cada libro, realizar préstamos a los clientes y mantener un registro de los préstamos realizados. Además, se desea que la base de datos esté diseñada siguiendo el modelo de copo de nieve para garantizar la eficiencia y la integridad de los datos.

**Requisitos del Sistema:**

* Los libros tienen un identificador único, un título, una fecha de publicación y un número de copias disponibles.
* Cada autor tiene un nombre único y puede escribir varios libros.
* Cada libro pertenece a uno o más géneros.
* Los clientes pueden realizar préstamos de libros.
* **Modelo Estrella - Modelo Conceptual**

**Diagrama

Descripción generada automáticamente**

* **Modelo Estrella - Modelo Lógico**

**Diagrama

Descripción generada automáticamente**

* **Modelo Estrella - Modelo Físico**

Diagrama

Descripción generada automáticamente

* **Tablas de Dimensiones**
* **DIM\_AUTOR**

**ID\_AUTOR:** Clave primaria única para la tabla DIM\_AUTOR.

**NOMBRE\_AUTOR:** Nombre del autor.

* **DIM\_CLIENTE**

**ID\_CLIENTE:** Clave primaria única para la tabla DIM\_CLIENTE.

**NOMBRE\_CLIENTE:** Nombre del cliente.

**DIRECCCION\_CLIENTE:** Dirección del cliente.

**NUMERO\_TELEFONO:** Número de teléfono del cliente.

* **DIM\_GENERO\_LITERARIO**

**ID\_GENERO:** Clave primaria única para la tabla DIM\_GENERO\_LITERARIO.

**NOMBRE\_GENERO:** Nombre del género literario.

* **DIM\_LIBRO**

**ID\_LIBRO:** Clave primaria única para la tabla DIM\_LIBRO.

**TITULO\_LIBRO:** Título del libro.

**FECHA\_PUBLICACION:** Fecha de publicación del libro.

**NUMERO\_COPIAS\_DISPONIBLES:** Número de copias disponibles del libro.

* **Tabla Hecho:**
* **HECHOS\_PRESTAMO**

**ID\_AUTOR:** Clave foránea que referencia a la tabla DIM\_AUTOR.

**ID\_GENERO:** Clave foránea que referencia a la tabla DIM\_GENERO\_LITERARIO.

**ID\_LIBRO:** Clave foránea que referencia a la tabla DIM\_LIBRO.

**ID\_CLIENTE:** Clave foránea que referencia a la tabla DIM\_CLIENTE.

**FECHA\_PRESTAMO:** Fecha en que se realizó el préstamo.

**FECHA\_DEVOLUCION:** Fecha en que se devolvió el libro.

**ESTADO:** Estado actual del préstamo (por ejemplo, Pendiente, Devuelto, etc.).

* **Replicación en PostgreSQL**

**Creación de la Red**

**Texto

Descripción generada automáticamente**

**Contenedor del Server 1**

Creamos el contenedor tuneado cada uno de sus parámetros y asignando un volumen y la red creada.

Texto

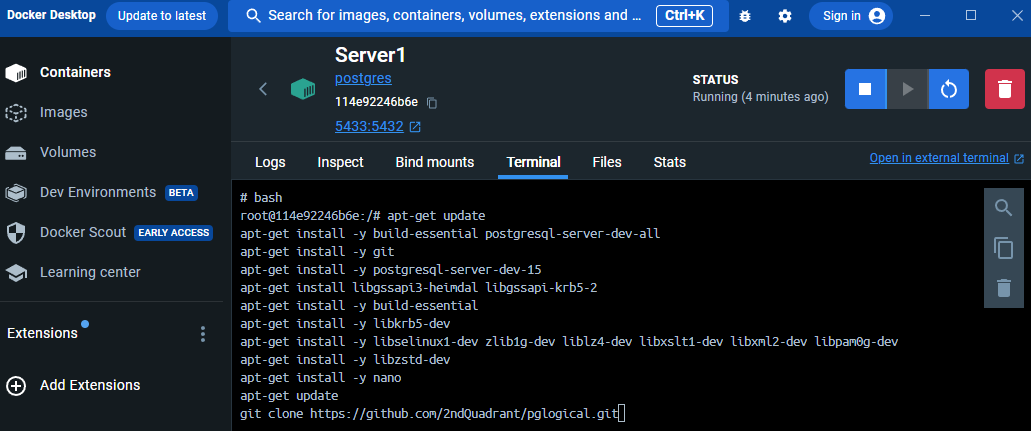
Descripción generada automáticamente

Verificamos en Docker la creación del contenedor.

Captura de pantalla de un celular

Descripción generada automáticamente

Ingresamos al bash del contenedor y asignaremos los siguientes complementos necesarios para la replicación.



Una vez terminado de instalarse los paquetes, ingresaremos a cd pglogical e instalaremos ahora el siguiente complemento.







Ahora ingresaremos al editor de texto con el comando de nano y cambiaremos los siguientes parámetros los cuales *son shared\_preload\_libraries = 'pglogical’* y *wal\_level = logical* lo podremos buscar presionando Ctrl + W. Luego de ello guardaremos los cambios con Ctrl + O y salimos con Ctrl + X.

nano /var/lib/postgresql/data/postgresql.conf

shared\_preload\_libraries = 'pglogical’

wal\_level = logical

Captura de pantalla de un celular

Descripción generada automáticamenteCaptura de pantalla de un celular

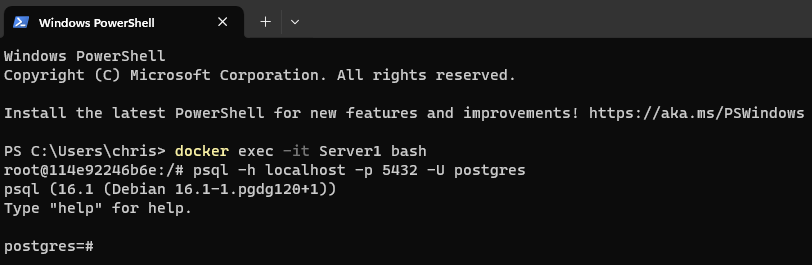
Descripción generada automáticamente

Reiniciamos nuestro contendedor para ejecutar los cambios.

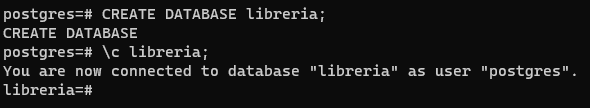
Texto

Descripción generada automáticamente

Una vez realizado las configuraciones previas, ingresaremos al contenedor de la siguiente forma:

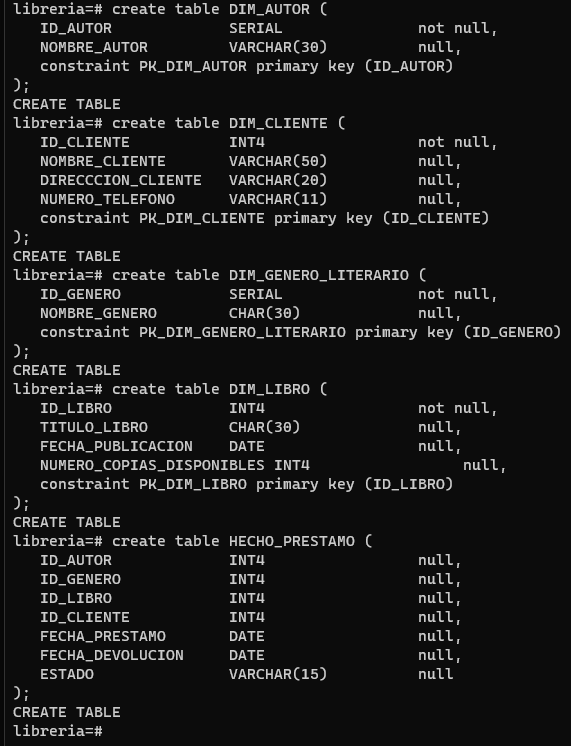


Creamos nuestra base de datos e ingresamos en ella.



Creamos las tablas que previamente modelamos.

Texto

Descripción generada automáticamente

Creamos una extensión

*CREATE EXTENSION pglogical*

Aplicamos la replicación

*SELECT pg\_create\_logical\_replication\_slot('slot\_1', 'pgoutput');*

Creamos un nodo con los siguientes parámetros

SELECT pglogical.create\_node(node\_name := 'node1', dsn := 'host=localhost port=5432 dbname=examen user=postgres password=123456');

Aplicamos un set a la replicación

*SELECT pglogical.create\_replication\_set('replication\_set\_1', true);*

Añadimos las tablas que creamos anteriormente para la replicación

*SELECT pglogical.replication\_set\_add\_table('replication\_set\_1', 'public.DIM\_AUTOR', true);*

*SELECT pglogical.replication\_set\_add\_table('replication\_set\_1', 'public.DIM\_CLIENTE', true);*

*SELECT pglogical.replication\_set\_add\_table('replication\_set\_1', 'public.DIM\_GENERO\_LITERARIO', true);*

*SELECT pglogical.replication\_set\_add\_table('replication\_set\_1', 'public.DIM\_LIBRO', true);*

*SELECT pglogical.replication\_set\_add\_table('replication\_set\_1', 'public.HECHO\_PRESTAMO', true);*

Agregamos una clave primaria en nuestra tabla de hecho\_prestamo.

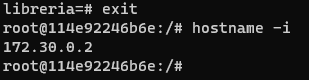
*ALTER TABLE HECHO\_PRESTAMO ADD COLUMN ID\_PRESTAMO SERIAL PRIMARY KEY;*

Texto

Descripción generada automáticamente

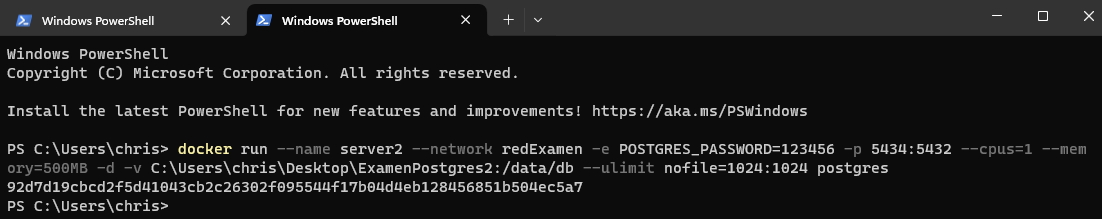


Salimos de pglogical con exit e ingresamos al root para identificar nuestro ip del contenedor del server1.

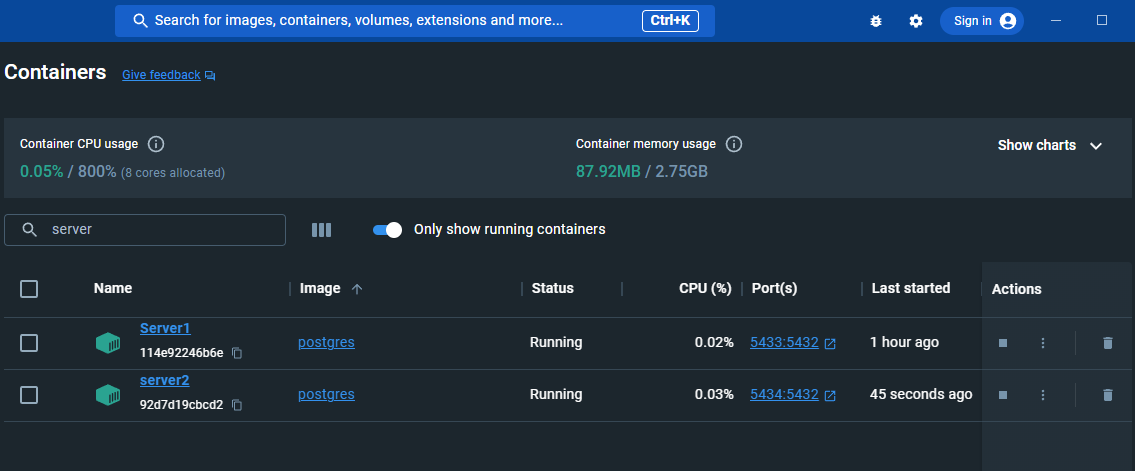


**Contenedor del Server 2**

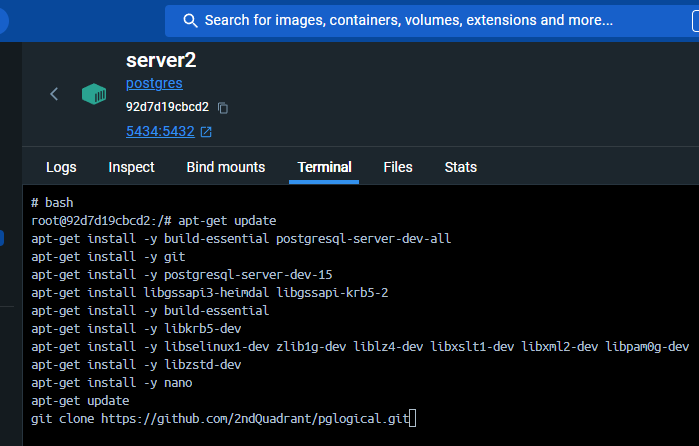
Creamos el contenedor tuneado cada uno de sus parámetros y asignando un volumen y la red creada.



Verificamos en Docker la creación del contenedor.



Ingresamos al bash del contenedor y asignaremos los siguientes complementos necesarios para la replicación.



Una vez terminado de instalarse los paquetes, ingresaremos a *cd pglogical* e instalaremos ahora los siguientes complementos.

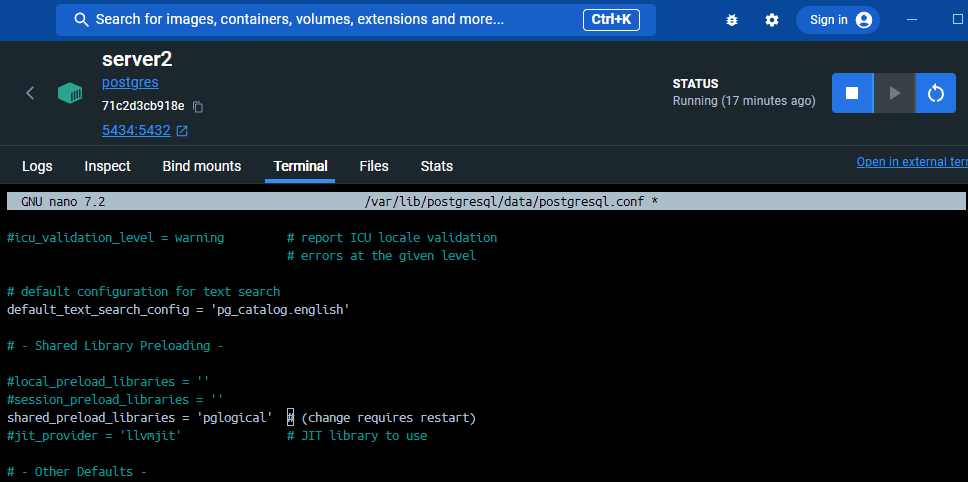






Ahora ingresaremos al editor de texto con el comando de nano y cambiaremos los siguientes parámetros los cuales son shared\_preload\_libraries = 'pglogical’ y wal\_level = logical lo podremos buscar presionando Ctrl + W. Luego de ello guardaremos los cambios con Ctrl + O y salimos con Ctrl + X.

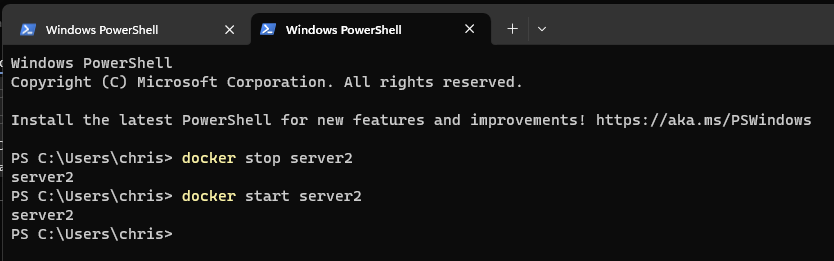




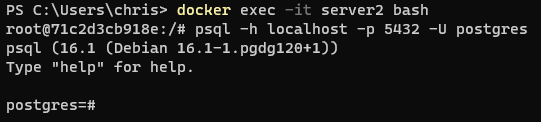
Captura de pantalla de un celular

Descripción generada automáticamente

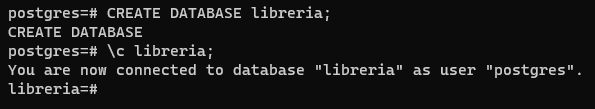
Reiniciamos nuestro contendedor para ejecutar los cambios.



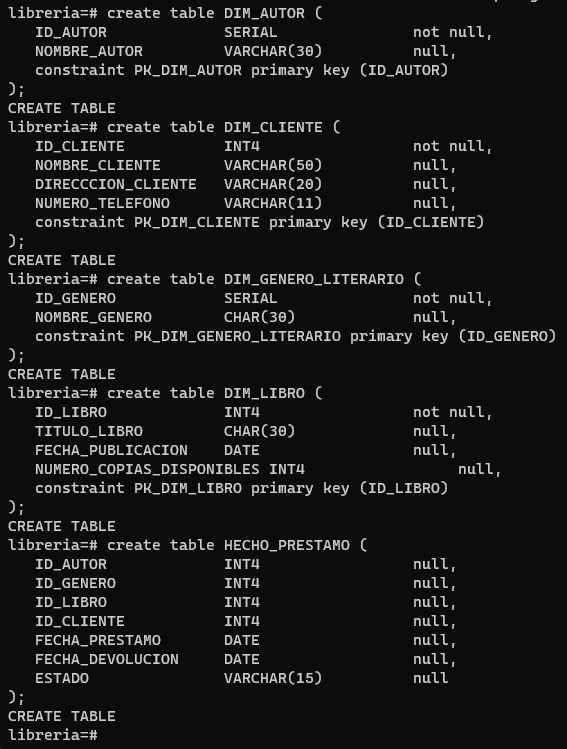
Una vez realizado las configuraciones previas, ingresaremos al contenedor de la siguiente forma:



Creamos nuestra base de datos e ingresamos en ella.



Creamos las tablas que previamente modelamos.



Creamos una extensión

*CREATE EXTENSION pglogical*

Creamos un nodo con los siguientes parámetros

SELECT pglogical.create\_node(node\_name := 'node2', dsn := 'host=localhost port=5432 dbname=libreria user=postgres password=123456');

Aplicamos un slot a la replicación

*SELECT slot\_name FROM pg\_replication\_slots WHERE plugin = 'pglogical\_output';*

Creamos la suscripción teniendo en cuenta la dirección ip del contenedor Server1

*SELECT pglogical.create\_subscription(*

*subscription\_name := 'subscription\_1',*

*provider\_dsn := 'host=172.30.0.2 port=5432 dbname=libreria user=postgres password=123456'*

*);*

Realizamos un replica set de la siguiente forma:

*SELECT pglogical.create\_replication\_set('replication\_set\_1', true);*

Configuramos la tablas que queremos replicar en nuestro set.

*SELECT pglogical.replication\_set\_add\_table('replication\_set\_1', 'public.DIM\_AUTOR', true);*

*SELECT pglogical.replication\_set\_add\_table('replication\_set\_1', 'public.DIM\_CLIENTE', true);*

*SELECT pglogical.replication\_set\_add\_table('replication\_set\_1', 'public.DIM\_GENERO\_LITERARIO', true);*

*SELECT pglogical.replication\_set\_add\_table('replication\_set\_1', 'public.DIM\_LIBRO', true);*

*SELECT pglogical.replication\_set\_add\_table('replication\_set\_1', 'public.HECHO\_PRESTAMO', true);*

Agregamos una clave primaria en nuestra tabla de hecho\_prestamo.

*ALTER TABLE HECHO\_PRESTAMO ADD COLUMN ID\_PRESTAMO SERIAL PRIMARY KEY;*

Configuramos nuestro set de replicación a la suscripción.

*SELECT pglogical.alter\_subscription\_add\_replication\_set('subscription\_1', 'replication\_set\_1');*

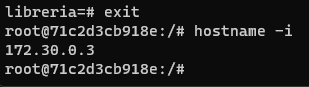
Texto

Descripción generada automáticamente

Texto

Descripción generada automáticamente

Salimos de pglogical con exit e ingresamos al root para identificar nuestro ip del contenedor del server2.



Regresamos al server1 para configurar nuestro set de replicación a la suscripción.

*SELECT pglogical.create\_subscription(*

*subscription\_name := 'subscription\_1',*

*provider\_dsn := 'host=172.30.0.3 port=5432 dbname=libreria user=postgres password=123456'*

*);*

*SELECT pglogical.alter\_subscription\_add\_replication\_set('subscription\_1', 'replication\_set\_1');*

Texto

Descripción generada automáticamente

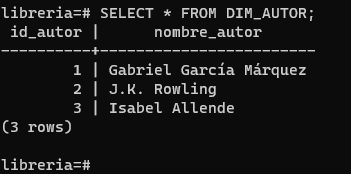
Verificamos la replicación haciendo inserciones en el Server1.

Realizaremos una inserción en la tabla de DIM\_AUTOR.

Texto

Descripción generada automáticamente

Verificaremos si la inserción mediante una consulta en el server2.



**F. NOMBRES, APELLIDOS Y FIRMA DEL ESTUDIANTE**

Texto

Descripción generada automáticamente

Christian Jhonatan Andrango Carchipulla

**G. FECHA DE ENTREGA**

Sangolquí, 13 de diciembre del 2023