



TALLER SHARDING

I. PORTADA

Tema:	Tratamiento de transacciones
Unidad de Organización Curricular:	PROFESIONAL
Nivel y Paralelo:	Quinto “A”
Alumnos participantes:	Vallejo Rengifo John David
Asignatura:	Sistemas de Bases de Datos Distribuidos
Docente:	Ing. José Caiza, Mg.

El siguiente comando se utilizará en Windows para crear las carpetas necesarias para MongoDB:

```
mkdir "%USERPROFILE%\mongodb\shard1" "%USERPROFILE%\mongodb\shard2"  
"%USERPROFILE%\mongodb\config"
```

Se ejecutará el comando anterior en el sistema operativo Windows con el fin de generar los directorios correspondientes a *shard1*, *shard2* y *config* dentro de la ruta del usuario, donde se almacenarán los datos de MongoDB.

```
C:\Windows\system32>mkdir "%USERPROFILE%\mongodb\shard1" "%USERPROFILE%\mongodb\shard2"  
Ya existe el subdirectorio o el archivo C:\Users\JOSUE\mongodb\shard1.
```

En el sistema operativo Linux se utilizará el siguiente comando para crear las carpetas necesarias para MongoDB:

```
mkdir ~/mongodb/shard1 ~/mongodb/shard2 ~/mongodb/config
```

Este comando generará los directorios *shard1*, *shard2* y *config* dentro del directorio personal del usuario, los cuales se emplearán para almacenar la información de cada componente de MongoDB.

```
josue@josue-VirtualBox:~$ mkdir "%USERPROFILE%\mongodb\shard1" "%USERPROFILE%\m  
ongodb\shard2" "%USERPROFILE%\mongodb\config"  
josue@josue-VirtualBox:~$
```

Los directorios se crean correctamente en **Ubuntu Server**. A continuación, se procederá con la **descarga e instalación de MongoDB** en el sistema **Ubuntu Desktop**.



```
Taller [Corriendo] - Oracle VirtualBox
Archivo Máquina Ver Entrada Dispositivos Ayuda
Actividades Terminal 11 de oct 22:52
joseue@joseue-VirtualBox: ~
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.
joseue@joseue-VirtualBox:~$ mkdir "%USERPROFILE%\mongodb\shard1" "%USERPROFILE%\m
ongodb\shard2" "%USERPROFILE%\mongodb\config"
joseue@joseue-VirtualBox:~$ sudo apt install mongodb-server-core
[sudo] contraseña para joseue:
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes adicionales:
libboost-program-options1.71.0 libgoogle-perftools4 libpcrecpp0v5
libsnappy1v5 libtcmalloc-minimal4 libyaml-cpp0.6
Se instalarán los siguientes paquetes NUEVOS:
libboost-program-options1.71.0 libgoogle-perftools4 libpcrecpp0v5
libsnappy1v5 libtcmalloc-minimal4 libyaml-cpp0.6 mongodb-server-core
0 actualizados, 7 nuevos se instalarán, 0 para eliminar y 385 no actualizados.
Se necesita descargar 22,4 MB de archivos.
Se utilizarán 82,5 MB de espacio de disco adicional después de esta operación.
¿Desea continuar? [S/n] s
Des:1 http://ec.archive.ubuntu.com/ubuntu focal/main amd64 libboost-program-opt
ions1.71.0 amd64 1.71.0-6ubuntu6 [342 kB]
Des:2 http://ec.archive.ubuntu.com/ubuntu focal/main amd64 libtcmalloc-minimal4
amd64 2.7-1ubuntu2 [93,0 kB]
Des:3 http://ec.archive.ubuntu.com/ubuntu focal/main amd64 libgoogle-perftools4
amd64 2.7-1ubuntu2 [195 kB]
Des:4 http://ec.archive.ubuntu.com/ubuntu focal-updates/main amd64 libpcrecpp0v
5 amd64 2:8.39-12ubuntu0.1 [15,5 kB]
Des:5 http://ec.archive.ubuntu.com/ubuntu focal/main amd64 libyaml-cpp0.6 amd64
```

Una vez instalado **MongoDB** en **Ubuntu**, se continúa con el proceso de **fragmentación de la base de datos** y la **replicación de los fragmentos** en diferentes instancias para garantizar la distribución y disponibilidad de los datos.

REPLICACION EN PG ADMIN

Primero crearemos la base de datos para postgres SQL

```
CREATE TABLE VEHICULO (
  ID_VEH SERIAL PRIMARY KEY,
  MAR_VEH VARCHAR(10) NOT NULL,
  MOD_VEH VARCHAR(10) NOT NULL,
  AÑO_VEH INT NOT NULL,
  ESTADO_DISP CHAR(1));
```

```
CREATE TABLE DOCUMENTOS (
  ID_DOC SERIAL PRIMARY KEY,
  TIP_DOC VARCHAR(10) NOT NULL,
  FEC_EXP_DOC DATE NOT NULL,
  ID_VEH_PER INT NOT NULL,
  FOREIGN KEY (ID_VEH_PER) REFERENCES VEHICULO(ID_VEH));
```

```
CREATE TABLE CONDUCTORES (
  ID_CONDUCTOR SERIAL PRIMARY KEY,
  NOM_CON VARCHAR(10) NOT NULL,
  LIC_CON VARCHAR(10) NOT NULL,
```



UNIVERSIDAD TÉCNICA DE AMBATO
FACULTAD DE INGENIERÍA EN SISTEMAS ELECTRÓNICA E INDUSTRIAL
CARRERA DE Elige un elemento.
CICLO ACADÉMICO: MARZO – JULIO 2025



FEC_VEN_LIC DATE NOT NULL,
ID_VEH_CON INT NOT NULL,
FOREIGN KEY (ID_VEH_CON) REFERENCES VEHICULO(ID_VEH));

CREATE TABLE MANTENIMIENTOS (
ID_MAN SERIAL PRIMARY KEY,
LIC_CON VARCHAR(10) NOT NULL,
FEC_VEN_LIC DATE NOT NULL,
ID_VEH_PER INT NOT NULL,
FOREIGN KEY (ID_VEH_PER) REFERENCES VEHICULO(ID_VEH));

CREATE TABLE GASOLINA (
ID_GAS SERIAL PRIMARY KEY,
TIP_GAS VARCHAR(10) NOT NULL,
DES_GAS VARCHAR(50) NOT NULL,
PRECIO_X_GALON DECIMAL(10,2) NOT NULL,
ID_VEH_PER INT NOT NULL,
FOREIGN KEY (ID_VEH_PER) REFERENCES VEHICULO(ID_VEH));

CREATE TABLE RUTAS (
ID_RUT SERIAL PRIMARY KEY,
FEC_RUT DATE NOT NULL,
ORI_RUT_VIA VARCHAR(10) NOT NULL,
DES_RUT_VIA VARCHAR(10) NOT NULL,
DIS_RRE_KM DECIMAL(10,2) NOT NULL,
DUR_EST_VIA DECIMAL(10,2) NOT NULL,
ID_VEH_PER INT NOT NULL,
FOREIGN KEY (ID_VEH_PER) REFERENCES VEHICULO(ID_VEH));



INSERCCION DE DATOS

Query Query History

```
1  -- TABLA VEHICULO
2  ✓ INSERT INTO VEHICULO (MAR_VEH, MOD_VEH, AÑO_VEH, ESTADO_DISP)
3  VALUES
4  ('Toyota', 'Hilux', 2020, 'A'),
5  ('Chevrolet', 'D-Max', 2019, 'A'),
6  ('Nissan', 'Frontier', 2021, 'I'),
7  ('Hyundai', 'Tucson', 2022, 'A'),
8  ('Kia', 'Sportage', 2018, 'I');
9
10 -- TABLA DOCUMENTOS
11 ✓ INSERT INTO DOCUMENTOS (TIP_DOC, FEC_EXP_DOC, ID_VEH_PER)
12 VALUES
13 ('SOAT', '2024-01-15', 1),
14 ('MATRICULA', '2024-02-10', 2),
15 ('REV_TEC', '2023-12-20', 3),
16 ('SOAT', '2024-03-01', 4),
17 ('MATRICULA', '2023-11-05', 5);
18
19 -- TABLA CONDUCTORES
20 ✓ INSERT INTO CONDUCTORES (NOM_CON, LIC_CON, FEC_VEN_LIC, ID_VEH_CON)
21 VALUES
22 ('Carlos', 'ABC123', '2026-05-10', 1),
23 ('María', 'BCD234', '2025-08-15', 2),
24 ('José', 'CDE345', '2026-01-20', 3),
25 ('Lucía', 'DEF456', '2025-11-30', 4),
26 ('Pedro', 'EFG567', '2024-09-25', 5);
27
28 -- TABLA MANTENIMIENTOS
29 ✓ INSERT INTO MANTENIMIENTOS (LIC_CON, FEC_VEN_LIC, ID_VEH_PER)
30 VALUES
31 ('ABC123', '2026-05-10', 1),
32 ('BCD234', '2025-08-15', 2),
33 ('CDE345', '2026-01-20', 3),
34 ('DEF456', '2025-11-30', 4),
35 ('EFG567', '2024-09-25', 5);
36
37 -- TABLA GASOLINA
38 ✓ INSERT INTO GASOLINA (TIP_GAS, DES_GAS, PRECIO_X_GALON, ID_VEH_PER)
39 VALUES
40 ('Extra', 'Gasolina regular', 2.10, 1),
41 ('Super', 'Gasolina de alto octanaje', 2.95, 2),
42 ('Diesel', 'Combustible pesado', 1.75, 3),
```



TRIGGER PARA VERIFICAR EL VENCIMIENTO DE LA LICENCIA

```
Query  Query History
1  CREATE OR REPLACE FUNCTION verificar_licencia_vencida()
2  RETURNS TRIGGER AS $$
3  BEGIN
4  -- Si la fecha de vencimiento es menor que la fecha actual, lanzar un error
5  IF NEW.FEC_VEN_LIC < CURRENT_DATE THEN
6  RAISE EXCEPTION 'No se puede registrar un conductor con licencia vencida. ' ;
7  END IF;
8  RETURN NEW;
9  END;
10 $$ LANGUAGE plpgsql;
11 CREATE TRIGGER trg_verificar_licencia_vencida
12 BEFORE INSERT OR UPDATE ON CONDUCTORES
13 FOR EACH ROW
14 EXECUTE FUNCTION verificar_licencia_vencida();
15
```

TRIGGER PARA VERIFICAR EL CONTROL Y DISPONIBILIDAD DE VEHICULOS

```
Query  Query History
1  CREATE OR REPLACE FUNCTION verificar_vehiculo_disponible()
2  RETURNS TRIGGER AS $$
3  DECLARE
4      estado CHAR(1);
5  BEGIN
6      --Obtener el estado del vehículo
7      SELECT ESTADO_DISP INTO estado FROM VEHICULO WHERE ID_VEH = NEW.ID_VEH_CON;
8      -- Si el vehículo no está disponible, generar un error
9  IF estado IS DISTINCT FROM 'A' THEN
10 RAISE EXCEPTION 'EI vehículo con ID % no está disponible para asignación.', NEW.ID_VEH_CON;
11 END IF;
12
13 RETURN NEW;
14 END;
15 $$ LANGUAGE plpgsql;
16 CREATE TRIGGER trg_verificar_vehiculo_disponible
17 BEFORE INSERT OR UPDATE ON CONDUCTORES
18 FOR EACH ROW
19 EXECUTE FUNCTION verificar_vehiculo_disponible();
```



TRIGGER PARA VER EL COSTO DEL VIAJE

```
Query Query History
1  ✓ CREATE OR REPLACE FUNCTION calcular_costo_viaje()
2  RETURNS TRIGGER AS $$
3  DECLARE
4  precio_litro DECIMAL (10,2);
5  ✓ BEGIN
6  -- Obtener el precio por litro de gasolina del vehículo asociado a la ruta
7  SELECT (PRECIO_X_GALON / 3.785) INTO precio_litro
8  FROM GASOLINA g
9  JOIN VEHICULO v ON g.ID_VEH_PER = v.ID_VEH
10 WHERE v.ID_VEH = NEW.ID_VEH_PER
11 LIMIT 1;
12 --Calcular el costo del viaje
13 NEW.COSTO_VIAJE := precio_litro * NEW.DIS_RRE_KM;
14 RETURN NEW;
15 END;
16 $$ LANGUAGE plpgsql;
17 ✓ CREATE TRIGGER trg_calcular_costo_viaje
18 BEFORE INSERT OR UPDATE ON RUTAS
19 FOR EACH ROW
20 EXECUTE FUNCTION calcular_costo_viaje();
```



UNIVERSIDAD TÉCNICA DE AMBATO
FACULTAD DE INGENIERÍA EN SISTEMAS ELECTRÓNICA E INDUSTRIAL
CARRERA DE Elige un elemento.
CICLO ACADÉMICO: MARZO – JULIO 2025



- ✓ EmpresaViajes
 - > Casts
 - > Catalogs (2)
 - > Event Triggers
 - > Extensions
 - > Foreign Data Wrappers
 - > Languages
 - > Publications
 - ✓ Schemas (1)
 - ✓ public
 - > Aggregates
 - > Collations
 - > Domains
 - > FTS Configurations
 - > FTS Dictionaries
 - > Aa FTS Parsers
 - > FTS Templates
 - > Foreign Tables
 - > Functions
 - > Materialized Views
 - > Operators
 - > Procedures
 - > 1..3 Sequences
 - ✓ Tables (6)
 - > conductores
 - > documentos
 - > gasolina
 - > mantenimientos
 - > rutas
 - > vehiculo
 - > Trigger Functions
 - > Types
 - > Views