

FACULTAD DE INGENIERÍA EN SISTEMAS ELECTRÓNICA E INDUSTRIAL







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**.

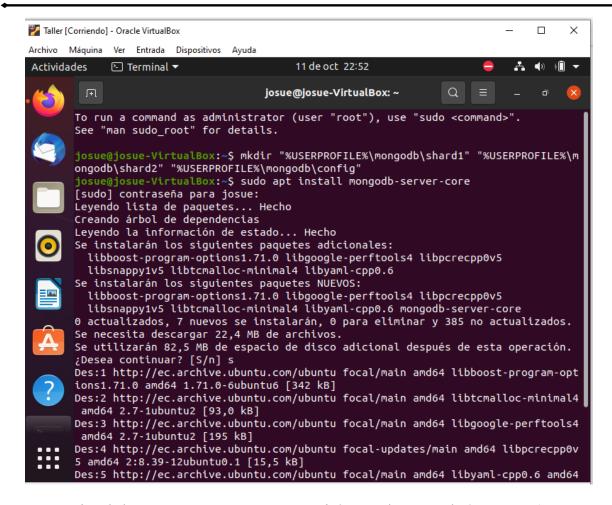


FACULTAD DE INGENIERÍA EN SISTEMAS ELECTRÓNICA E INDUSTRIAL



CARRERA DE Elige un elemento.

CICLO ACADÉMICO: MARZO - JULIO 2025



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,



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));







CARRERA DE Elige un elemento.

CICLO ACADÉMICO: MARZO – JULIO 2025

INSERCCION DE DATOS

```
Query Query History
 1 -- TABLA VEHICULO
 2 V 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'),
    ('Hyundai', 'Tucson', 2022, 'A'),
 7
    ('Kia', 'Sportage', 2018, 'I');
 8
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),
    ('MATRICULA', '2023-11-05', 5);
17
18
19
    -- TABLA CONDUCTORES
20 V INSERT INTO CONDUCTORES (NOM_CON, LIC_CON, FEC_VEN_LIC, ID_VEH_CON)
21
    VALUES
    ('Carlos', 'ABC123', '2026-05-10', \mathbf{1}),
22
   ('María', 'BCD234', '2025-08-15', 2),
23
24 ('José', 'CDE345', '2026-01-20', 3),
25 ('Lucía', 'DEF456', '2025-11-30', 4),
     ('Pedro', 'EFG567', '2024-09-25', 5);
26
27
28
    -- TABLA MANTENIMIENTOS
29 V INSERT INTO MANTENIMIENTOS (LIC_CON, FEC_VEN_LIC, ID_VEH_PER)
30 VALUES
    ('ABC123', '2026-05-10', 1),
31
    ('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 V INSERT INTO GASOLINA (TIP_GAS, DES_GAS, PRECIO_X_GALON, ID_VEH_PER)
40 ('Extra', 'Gasolina regular', 2.10, 1),
    ('Super', 'Gasolina de alto octanaje', 2.95, 2),
41
42 ('Diesel', 'Combustible pesado', 1.75, 3),
```







CARRERA DE Elige un elemento.

CICLO ACADÉMICO: MARZO - JULIO 2025

TRIGGER PARA VERIFICAR EL VENCIMIENTO DE LA LICENCIA

```
Query Query History
1 v CREATE OR REPLACE FUNCTION verificar_licencia_vencida()
    RETURNS TRIGGER AS $$
    -- Si Ia fecha de vencimiento es menor que la fecha actual, lanzar un error
    IF NEW.FEC_VEN_LIC < CURRENT_DATE THEN</pre>
     RAISE EXCEPTION 'No se puede registrar un conductor con licencia vencida. ';
7
     END IF;
8
    RETURN NEW;
    END;
9
10 $$ LANGUAGE plpgsql;
11 v CREATE TRIGGER trg_verificar_licencia_vencida
   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 v CREATE OR REPLACE FUNCTION verificar_vehiculo_disponible()
    RETURNS TRIGGER AS $$
3
    DECLARE
4
       estado CHAR(1);
5 ▼ BEGIN
6
       --Obtener el estado del vehículo
    SELECT ESTADO_DISP INTO estado FROM VEHICULO WHERE ID_VEH = NEW.ID_VEH_CON;
    -- Si el vehículo no está disponible, generar un error
9 v IF estado IS DISTINCT FROM 'A' THEN
    RAISE EXCEPTION 'EI vehículo con ID % no está disponible para asignación.', NEW.ID_VEH_CON;
10
11
    END IF;
12
13 RETURN NEW;
14 END;
    $$ LANGUAGE plpgsql;
16 v CREATE TRIGGER trg_verificar_vehiculo_disponible
17 BEFORE INSERT OR UPDATE ON CONDUCTORES
18 FOR EACH ROW
19     EXECUTE FUNCTION verificar_vehiculo_disponible();
```







CARRERA DE Elige un elemento.
CICLO ACADÉMICO: MARZO – JULIO 2025

TRIGGER PARA VER EL COSTO DEL VIAJE

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

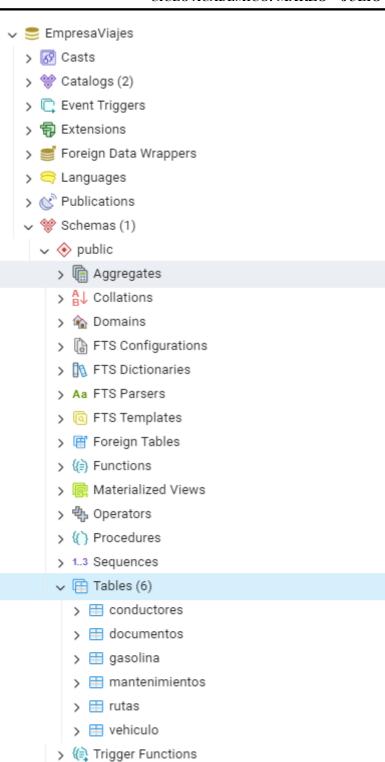


FACULTAD DE INGENIERÍA EN SISTEMAS ELECTRÓNICA E INDUSTRIAL



CARRERA DE Elige un elemento.

CICLO ACADÉMICO: MARZO - JULIO 2025



TypesViews