**Ejercicio MongoDB**

Ismenia Marcela Guevara Ortiz

Juan Esteban Quintero Rodriguez

Universidad del Quindío

Ingeniería de Sistemas y Computación

Carlos Andrés Flórez Villarraga

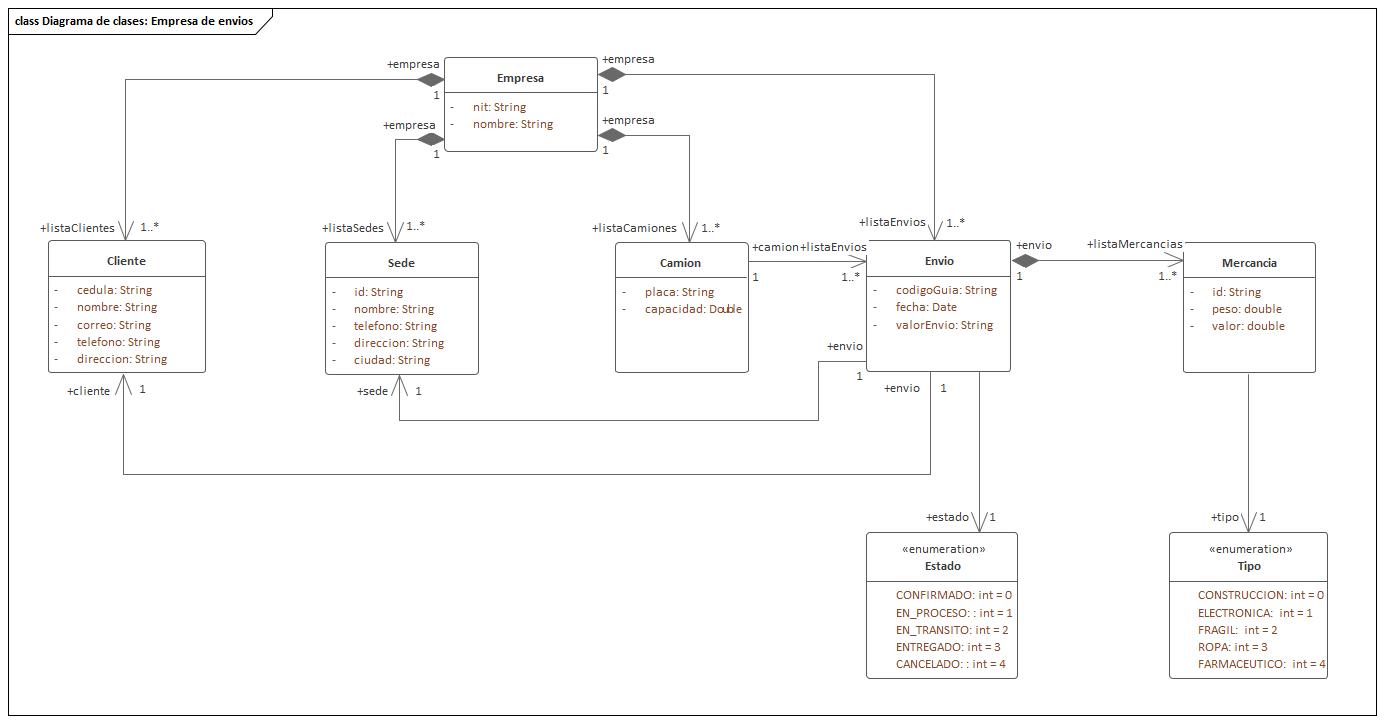
Quindío, Colombia

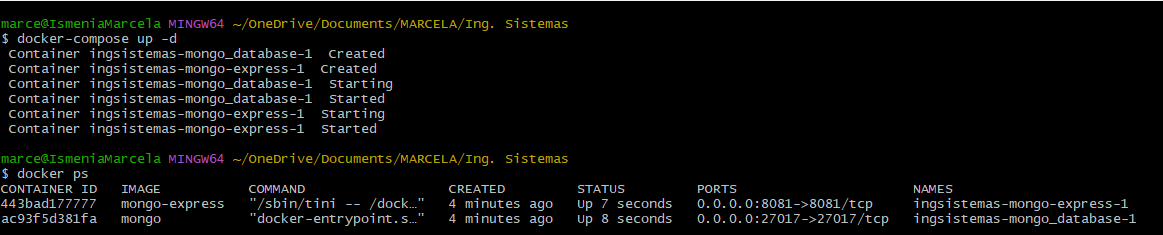
24 de febrero de 2024

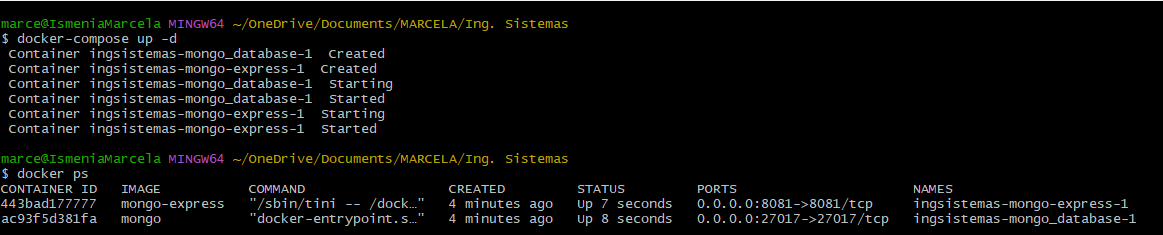
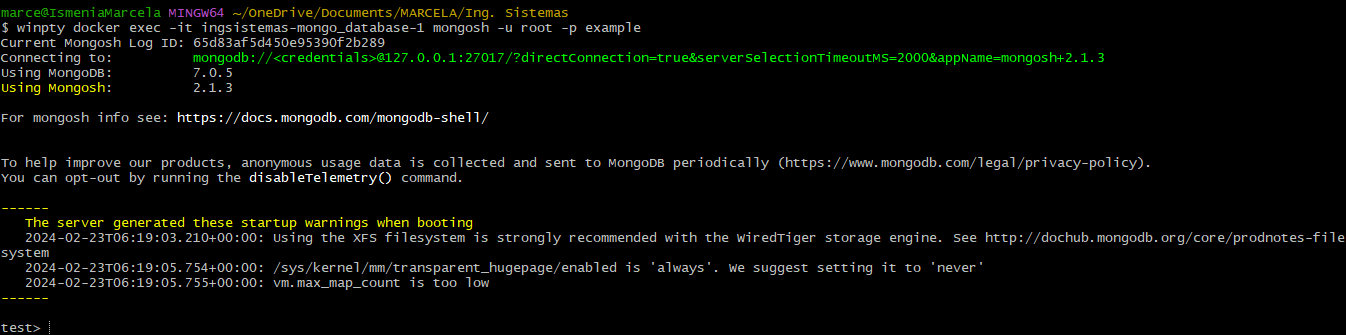
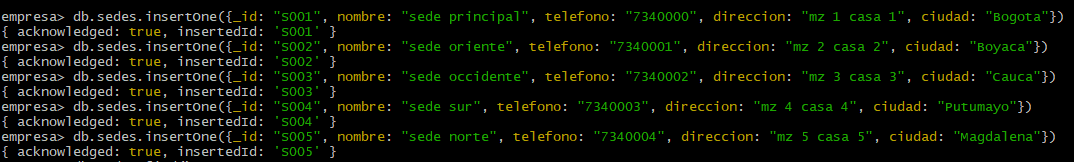
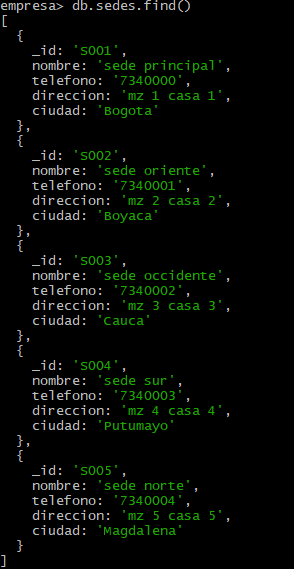
**Enunciado:**

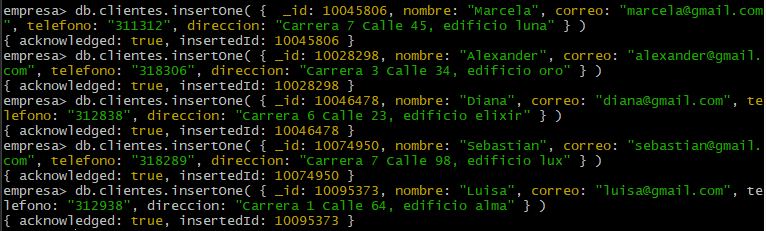
Una empresa de transporte de mercancías requiere un sistema de información para gestionar sus clientes y envíos. La empresa cuenta con unas sedes, cada sede tiene una dirección, ciudad, nombre y un número de teléfono. Para registrar el envío de la mercancía de un cliente se requiere crear una orden con la información del remitente (cédula del cliente **debe ser el \_id**, nombre, correo, teléfono y dirección), la información del destinatario, la información de la mercancía (peso, valor declarado, tipo), la sede de envío, la fecha de envío, valor envío y el código de guía. Cada orden de envío debe tener un registro de los estados por los que va pasando en el tiempo.

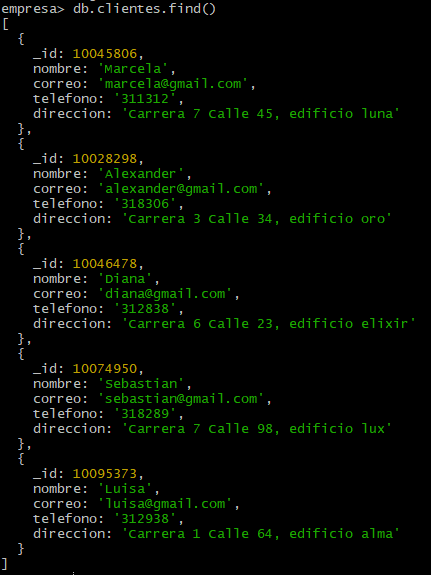
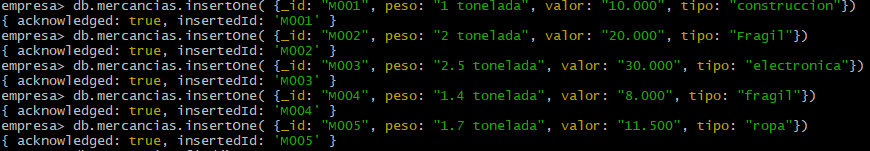
Además, la empresa cuenta con una flota de camiones, cada camión tiene una placa (**debe ser el** \_id), una capacidad (toneladas), una ruta y cada orden de envío debe ir en un determinado camión.

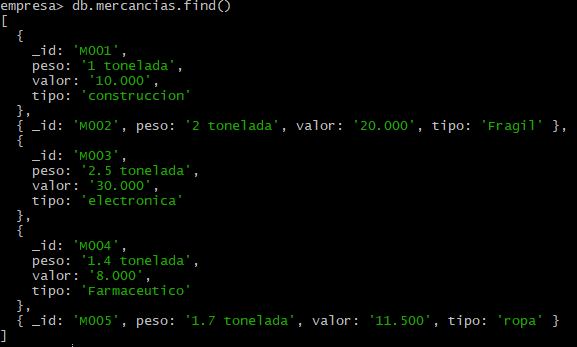
1. Se identifican las clases junto con sus atributos con el fin de crear el diagrama de clases utilizando la herramienta Enterprise Architect. Este enfoque tiene como objetivo visualizar de manera clara las relaciones existentes entre las diversas colecciones.
2. Se crea la base de datos en MongoDB con sus respectivas colecciones y con 5 documentos cada uno:
   1. Se inician los servicios definidos en docker-compose.yml con el comando:

docker-compose up -d

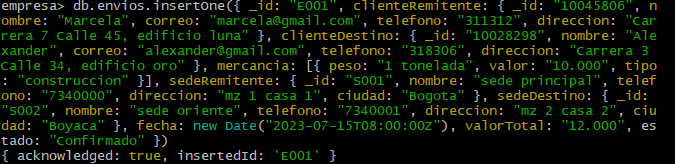
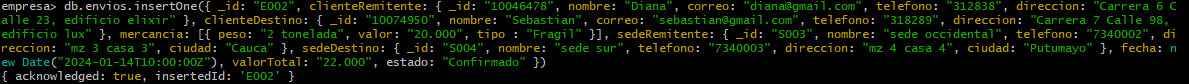
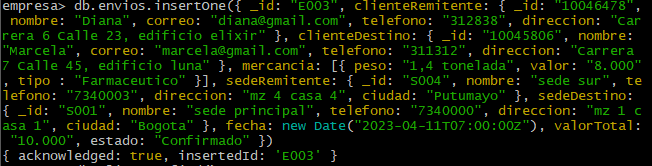
* 1. Se consultan los contenedores con el comando: docker ps
  2.  Se ejecuta la base de datos elegida con el comando: winpty docker exec -it ingsistemas-mongo\_database-1 mongosh -u root -p example
  3. Creamos la base de datos llamada empresa con el comando: use empresa
  4. Creamos la colección llamada sedes y sus documentos con el comando: db.sedes.insertOne({atributo: valor})
  5. Consultamos los documentos creados con el comando: db.sedes.find()
  6. Creamos la colección llamada clientes y sus documentos con el comando: db.clientes.insertOne({atributo: valor})

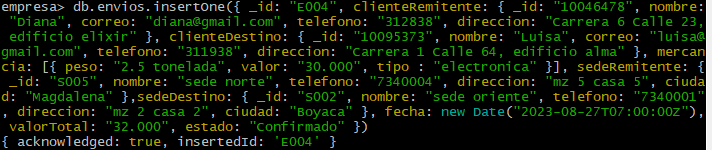


* 1. Consultamos los documentos creados con el comando: db.clientes.find()
  2. Creamos la colección llamada mercancias y sus documentos con el comando: db.mercancias.insertOne({atributo: valor}
  3. Consultamos los documentos creados con el comando: db.mercancias.find()

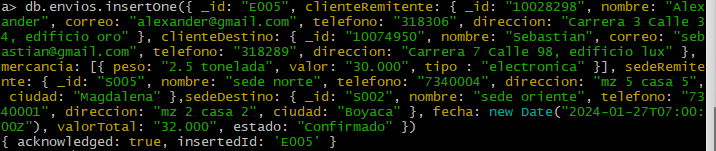


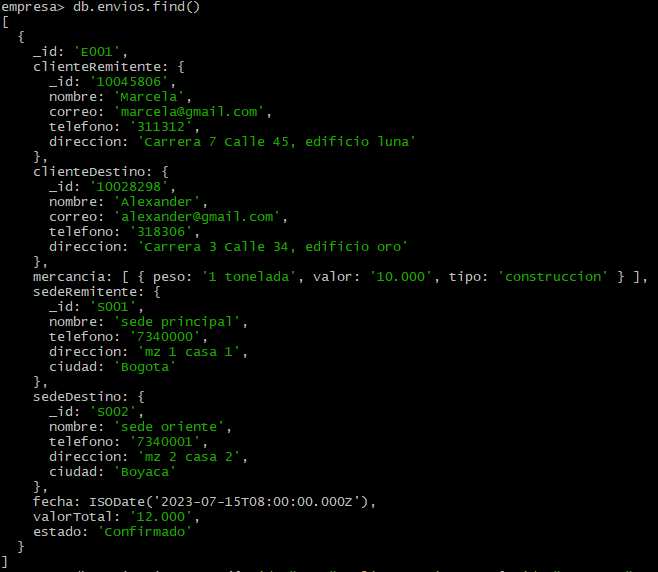
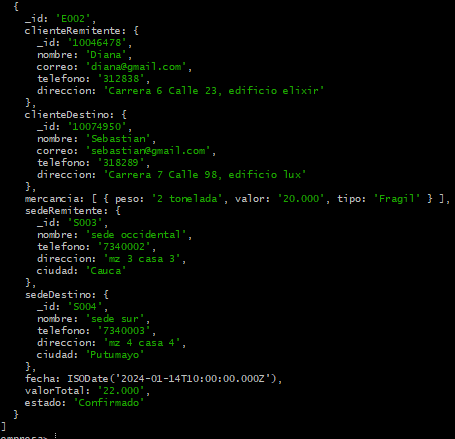
2.11. Creamos la colección llamada envíos y sus documentos con el comando: db.envios.insertOne({atributo: valor}

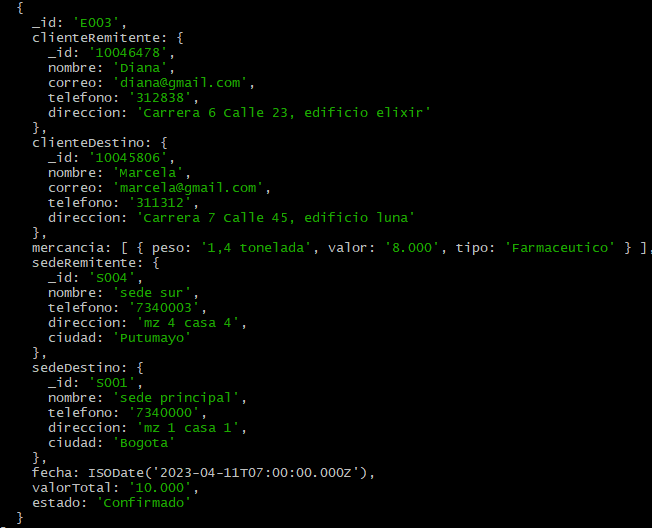
* + 1. Documento de envio E001:
    2. Documento de envio E002:
    3. Documento de envío E003:
    4. Documento de envio E004:

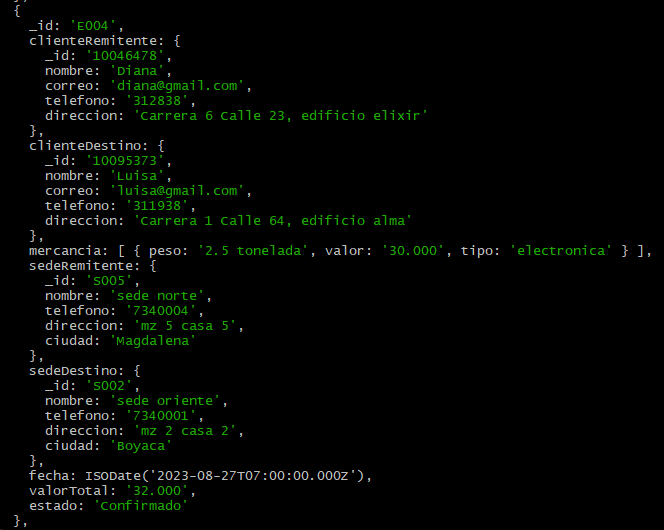


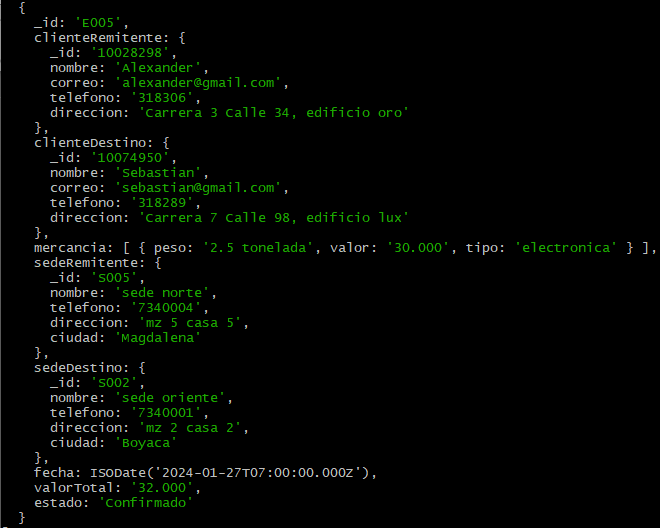
* + 1. Documento de envio E005:



* 1. Consultamos los documentos creados con el comando: db.envios.find()
     1. Documento de envio E001:
     2. Documento de envio E002:
     3. Documento de envio E003:



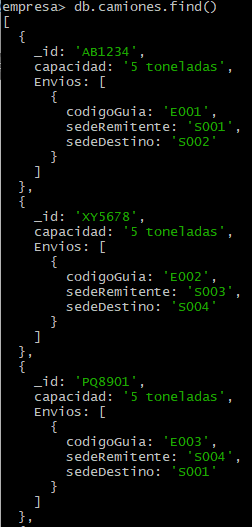
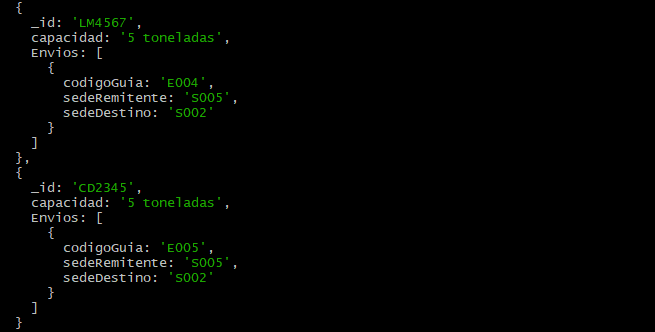
* + 1. Documento de envio E004:
    2. Documento de envio E005:



* 1. Creamos la colección llamada camiones y sus documentos con el comando: db.camiones.insertOne({atributo: valor}



* 1. Consultamos los documentos creados con el comando: db.camiones.find()



1. Escriba una consulta que, dado el \_id de un cliente, devuelva una lista con todos los envíos que ha realizado en el año 2023.

Se usa la siguiente declaración y la lista que nos muestra efectivamente concuerda con los envíos que habíamos creado, **cabe aclarar que se tuvo que actualizar los id de cada cliente de string a entero para poder que funcionara el método:**

var cedula = 10046478; var envios2023 = db.envios.find({ "clienteRemitente.\_id": cedula, fecha: { $gte: new Date("2023-01-01T00:00:00Z"), $lt: new Date("2024-01-01T00:00:00Z") } } ); printjson(envios2023.toArray())



