

Migración de Base de Datos de SQL a NoSQL

Servidor Local

A partir del estudio realizado se propone la migración del modelo relacional SQL(Relacional) al NoSQL(No Relacional) basado en las reglas del modelo no relacional.

La Migración se realizó por dos medios, por servidor local usando MongoDB Community Server y MongoAtlas en la nube, en el primer caso se usó Python con la librería pymongo y SQLite3 en el editor Visual Studio Code, mientras que en el segundo caso se usó Google Colab usando las librerías SQLite3, Pandas y Pymongo.

Previo a la explicación del procedimiento de la migración dejaremos claro cuáles fueron las analogías a nivel de estructura, restricciones y operaciones que se siguieron entre los dos modelos.

1. Estructura

En el siguiente cuadro se mostrará la analogía de ambos modelos, de SQL y NoSQL

Características	Analogía de Relacional y No relacional.	
	SQL	NoSQL
	SQLite3	MongoDB
Modelo	Base de datos	Base de Datos
	Tabla	Colección
	Fila	Documento
	Índice	Índice
Data	Estructurado	Semi estructurado/No Estructurado
Esquema	Estático	Sin esquema
Desempeño	Bajo	Alto
Confiabilidad	Alto	Bajo
Disponibilidad	Alto	Buena
Consistencia	Alto	Bajo
Almacenamiento de datos	Mediano a Grande	Grandes volúmenes de datos
Transacciones	ACID	Teorema de CAP
Escalabilidad	Vertical	Horizontal
Recursos Necesarios	Medio	Bajo

2. Restricciones

Al comunicar el modelo SQL al NoSQL no se mantendrá las características propias del primer modelo, no obstante se mantendrá los identificadores únicos a cada documento generado automáticamente por mongodb adicionalmente al que ya tiene SQL.

3. Operaciones

Las operaciones usadas para la migración se aprecian en el cuadro comparativo.

Operaciones	
SQL	NoSQL
SQLite3	MongoDB
Select	Find
Insert	Insert
Delete	Remove
Update	Update

4. Procedimiento de Migración

Previo a la migración se instalaron los entornos de visualización de SQLite3 (DB Browser for SQLite) y MongoDB(MongoDBCompass), de igual forma los entornos de trabajo para las consultas desde CMD (MongoDB Community).

4.1. Creación de base de datos en MongoDB

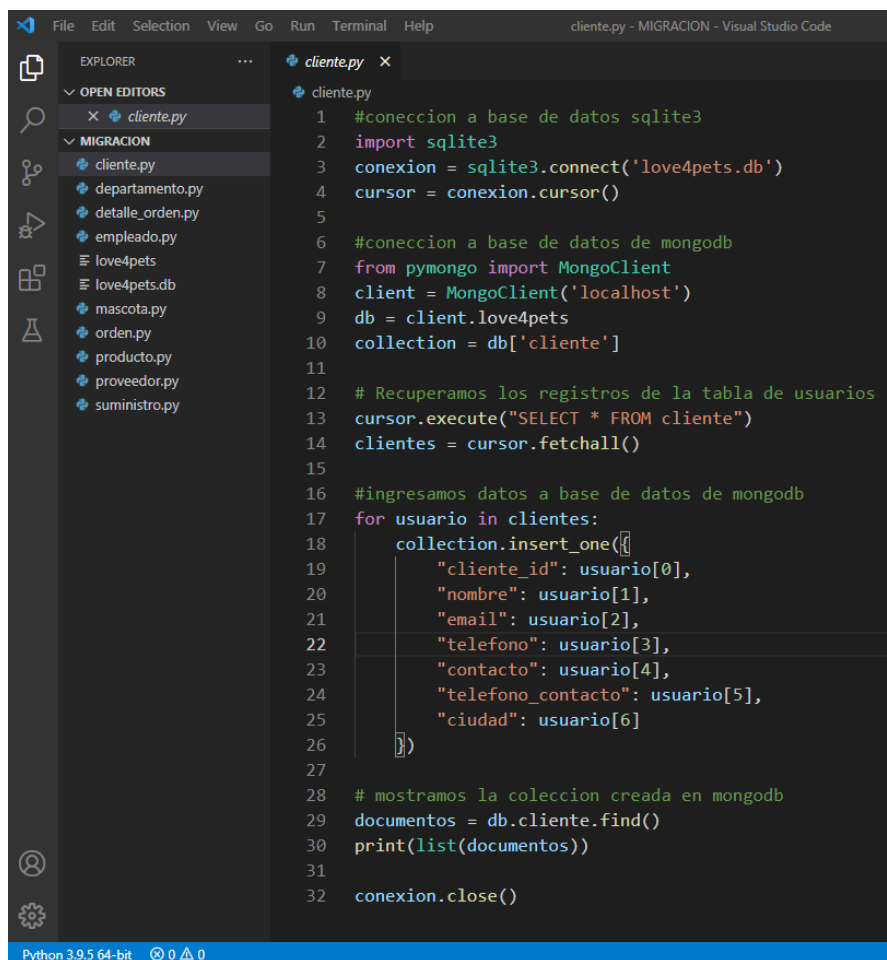
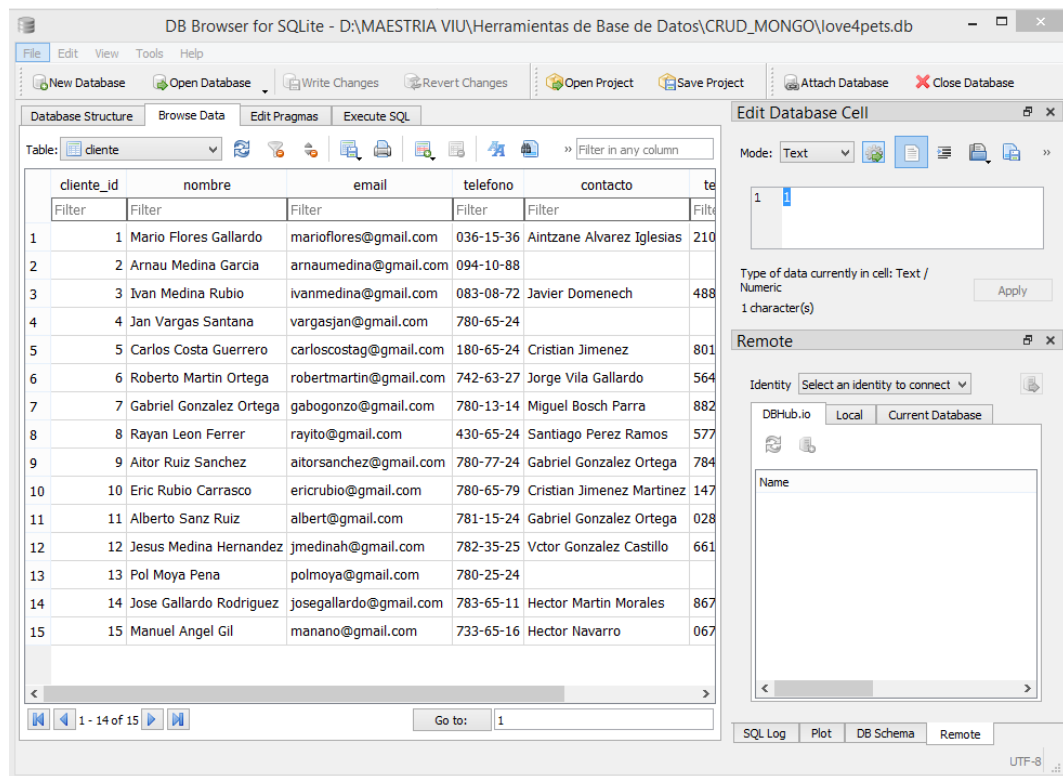
Mediante la consola de mongo se hizo la creación de la base de datos love4pets con una primera colección en blanco llamada cliente, con la siguiente sintaxis:

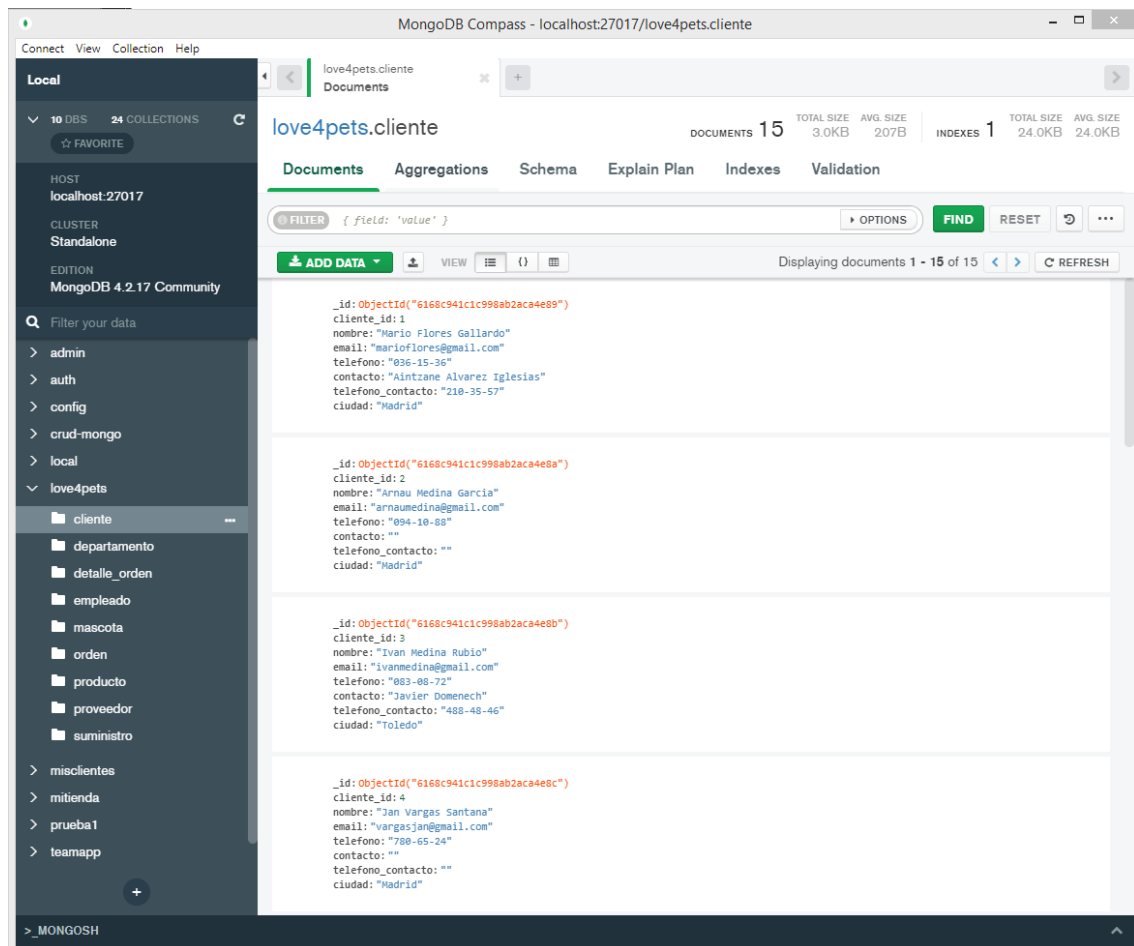
- > use love4pets
- > db.createCollection("cliente")

4.2. Desarrollo de código en python para la Migración

Básicamente es un código para la lectura de las consultas SQL y la manipulación con Python de esta para trasladar a la base de datos creada en MongoDB, para esto se necesitó de algunas librerías como pymongo y sqlite3.

Se desarrolló el código en Visual Studio Code para cada colección por separado teniendo así 9 archivos.py, el código se repite para cada archivo solo varia los nombres y campos de las colecciones a crear, a continuación se mostrará el código de la migración de la tabla cliente a la colección cliente.





4.3. Transformación de data cadena a numérico

En la columna teléfono de la colección proveedor los datos estaban en string, el dato original en SQL era numérico, así que se tuvo que convertir mediante la siguiente sintaxis:

```
collection = db.proveedor
```

```
telefono_list = list(collection.distinct('telefono'))
```

```
for num in telefono_list:
```

```
collection.update_many({'telefono' : num},{'$set': {'telefono' : int(num)}})
```

A continuación se puede ver en la imagen los cambios realizados.

MongoDB Compass - localhost:27017/love4pets.proveedor

love4pets.proveedor Documents

DOCUMENTS 4 TOTAL SIZE 481B AVG. SIZE 120B INDEXES 1 TOTAL SIZE 20.0KB AVG. SIZE 20.0KB

Documents Aggregations Schema Explain Plan Indexes Validation

FILTER { field: 'value' } OPTIONS FIND RESET

ADD DATA VIEW

Displaying documents 1 - 4 of 4 REFRESH

#	proveedor
	_id ObjectId proveedor_id Int32 nombre String email String telefono String
1	61690418f6540bb85747bbee 1 "Pets prov" "petsprov@gmail.com" "818853151"
2	61690418f6540bb85747bbef 2 "maxipet" "maxipet@gmail.com" "8650001508"
3	61690418f6540bb85747bbf0 3 "adorables mascotas" "adorablesmascotas@gmail.com" "7150383416"
4	61690418f6540bb85747bbf1 4 "huesitos" "huesitos@gmail.com" "6640163977"

> MONGOSH

MongoDB Compass - localhost:27017

love4pets.proveedor Documents

DOCUMENTS 4 TOTAL SIZE 481B AVG. SIZE 120B INDEXES 1 TOTAL SIZE 20.0KB AVG. SIZE 20.0KB

Documents Aggregations Schema Explain Plan Indexes Validation

FILTER { field: 'value' } OPTIONS FIND RESET

ADD DATA VIEW

Displaying documents 1 - 4 of 4 REFRESH

#	proveedor
	_id ObjectId proveedor_id Int32 nombre String email String telefono Mixed
1	61690418f6540bb85747bbee 1 "Pets prov" "petsprov@gmail.com" 818853151
2	61690418f6540bb85747bbef 2 "maxipet" "maxipet@gmail.com" 8650001508
3	61690418f6540bb85747bbf0 3 "adorables mascotas" "adorablesmascotas@gmail.com" 7150383416
4	61690418f6540bb85747bbf1 4 "huesitos" "huesitos@gmail.com" 6640163977

> MONGOSH

Migración de Base de Datos de SQL a NoSQL

Servidor en la Nube

Brevemente explicaré el proceso, básicamente el mayor problema que se tuvo fue la conexión de google colab a la base de datos de MongoDB Atlas debido a fallas en el reconocimiento de la ruta de la base de datos en la nube, esta no funcionaba hasta que se tuvo que reiniciar el entorno de ejecución después de instalar la librería dnspython, pymongo[srv] y pymongo[tls] para el correcto funcionamiento de la ruta en la nube, el cual se verifico mediante el código client.stats.

- Verificando la conexión de la base de datos de MongoDB Atlas

```
[2] client.stats
```

```
Database(MongoClient(host=['cluster0-shard-00-01.z6avn.mongodb.net:27017', 'cluster0-shard-00-00.z6avn.mongodb.net:27017', 'cluster0-shard-00-02.z6avn.mongodb.net:27017'], document_class=dict, t
```

Seguidamente se procedió a la conexión de las dos bases de datos de SQLite3 y MongoDB, se importaron las librerías sqlite3 y pymongo respectivamente.

- Conectando Base de Datos de SQLite3 y MongoDB Atlas

```
import sqlite3
```

```
conexion = sqlite3.connect('love4pets.db')  
cursor = conexion.cursor()
```

```
[27] import pymongo
```

```
from pymongo import MongoClient
```

```
uri = 'mongodb+srv://juank-y23:juank-y23key@cluster0.z6avn.mongodb.net/love4pets?retryWrites=true&w=majority'
```

```
client = MongoClient( uri )
```

```
db = client.love4pets
```

Migración de Tablas a Colecciones.

Básicamente este proceso es el mismo realizado en la migración local, se usó el mismo código para tal efecto en Google Colab, es proceso se repite para todas 9 tablas a migrar, en el cual solo se cambian las variables de los campos y el nombre de la colección.

- Migración de la tabla Cliente sql a mongodb

```
[4] collection = db['cliente']  
cursor.execute('SELECT * FROM cliente')  
clientes = cursor.fetchall()  
for usuario in clientes:  
    collection.insert_one({  
        "cliente_id": usuario[0],  
        "nombre": usuario[1],  
        "email": usuario[2],  
        "telefono": usuario[3],  
        "contacto": usuario[4],  
        "telefono_contacto": usuario[5],  
        "ciudad": usuario[6]  
    })  
documentos = db.cliente.find()  
print(list(documentos))
```

```
[(1, 'Mario Flores Gallardo', 'marioflores@gmail.com', '036-15-36', 'Ainzane Alvarez Iglesias', '210-35-57', 'Madrid'), (2, 'Arnau Medina Garcia', 'arnaumedina@gmail.com', '094-10-88', '', '',
```

```

- Migracion de la tabla departamento

[8] collection = db['departamento']
cursor.execute("SELECT * FROM departamento")
departamentos = cursor.fetchall()

#ingresamos datos a base de datos de mongodb
for usuario in departamentos:
    collection.insert_one({
        "departamento_id": usuario[0],
        "nombre": usuario[1]
    })
doc_names = db.departamento.find()
print(list(doc_names))

[{"_id": ObjectId("616b3b24ad37240b7afc4024"), "departamento_id": 1, "nombre": "finanzas"}, {"_id": ObjectId("616b3b25ad37240b7afc4025"), "departamento_id": 2, "nombre": "ventas"}, {"_id": ObjectId("616b3b26ad37240b7afc4026"), "departamento_id": 3, "nombre": "marketing"}]
```

Modificación de Datos en MongoDB

Dentro de la estructura de la base de datos SQL la tabla proveedor tiene una columna teléfono el cual es de tipo numérico, al momento de la migración los datos de esa columna se pasaron a mongo como cadena de texto, esto implicaba tener que hacer un cambio de todos los datos a tipo numérico en la columna teléfono, para esto se usó el siguiente código:

```

Modificacion los valores de tipo string a numerico de la columna telefono de la
coleccion proveedor

[20] collection = db.proveedor
telefono_list = list(collection.distinct('telefono'))
for num in telefono_list:
    collection.update_many({'telefono': num}, {'$set': {'telefono': int(num)}})
```

love4pets.proveedor

COLLECTION SIZE: 481B TOTAL DOCUMENTS: 4 INDEXES: 0

Find Indexes Schema Anti-Patterns 0

FILTER { field: 'value' }

QUERY RESULTS 1-4 OF 4

_id: ObjectId("616b49faad37240b7afc4089")

proveedor_id: 1

nombre: "Pets prov"

email: "petsprov@gmail.com"

telefono: "818853151"

_id: ObjectId("616b49faad37240b7afc408a")

proveedor_id: 2

nombre: "maxipet"

email: "maxipet@gmail.com"

telefono: "8650001508"

_id: ObjectId("616b49fbad37240b7afc408b")

proveedor_id: 3

nombre: "adorables mascotas"

email: "adorablesmascotas@gmail.com"

telefono: "7150383416"

love4pets.proveedor

COLLECTION SIZE: 450B TOTAL DOCUMENTS: 4 INDEXES: 0

Find Indexes Schema Anti-Patterns 0

FILTER { field: 'value' }

QUERY RESULTS 1-4 OF 4

_id: ObjectId("616b49faad37240b7afc4089")

proveedor_id: 1

nombre: "Pets prov"

email: "petsprov@gmail.com"

telefono: 818853151

_id: ObjectId("616b49faad37240b7afc408a")

proveedor_id: 2

nombre: "maxipet"

email: "maxipet@gmail.com"

telefono: 8650001508

_id: ObjectId("616b49fbad37240b7afc408b")

proveedor_id: 3

nombre: "adorables mascotas"

email: "adorablesmascotas@gmail.com"

telefono: 7150383416

Visualización de la base de datos desde MongoDB Atlas

Para la visualización de dataframes se importó la librería pandas, a continuación se muestra las tres primeras Colecciones de muestra, ya que se adjuntara el archivo **Tarea_02_MongoDB-PyMongo.ipynb** en donde se podrá revisar todas las Colecciones.

VISUALIZACION DE COLECCIONES DESDE LA BASE DE DATOS DE MONGODB ATLAS

```
[33] import pandas as pd
     pd.options.display.max_columns = None
```

Coleccion Cliente

```
[37] documentos = db.cliente.find()
     data = list(documentos)
     df = pd.DataFrame(data)
     df.head()
```

	_id	cliente_id	nombre	email	telefono	contacto	telefono_contacto	ciudad
0	616b35cead37240b7afc4014	1	Mario Flores Gallardo	marioflores@gmail.com	036-15-36	Aintzane Alvarez Iglesias	210-35-57	Madrid
1	616b35d0ad37240b7afc4015	2	Arnau Medina Garcia	arnaumedina@gmail.com	094-10-88			Madrid
2	616b35d0ad37240b7afc4016	3	Ivan Medina Rubio	ivanmedina@gmail.com	083-08-72	Javier Domenech	488-48-46	Toledo
3	616b35d0ad37240b7afc4017	4	Jan Vargas Santana	vargasjan@gmail.com	780-65-24			Madrid
4	616b35d0ad37240b7afc4018	5	Carlos Costa Guerrero	carloscostag@gmail.com	180-65-24	Cristian Jimenez	801-28-09	Madrid

Coleccion Departamento

```
[38] documentos = db.departamento.find()
     data = list(documentos)
     df = pd.DataFrame(data)
     df.head()
```

	_id	departamento_id	nombre
0	616b3b24ad37240b7afc4024	1	finanzas
1	616b3b25ad37240b7afc4025	2	ventas
2	616b3b25ad37240b7afc4026	3	recursos humanos
3	616b3b26ad37240b7afc4027	4	marketing
4	616b3b26ad37240b7afc4028	5	veterinario

Coleccion Detalle de Orden

```
[39] documentos = db.detalle_orden.find()
     data = list(documentos)
     df = pd.DataFrame(data)
     df.head()
```

	_id	detalle_id	orden_id	producto_id	cantidad	precio
0	616b3eddad37240b7afc4032	1	1	1	2	7
1	616b3eddad37240b7afc4033	2	1	3	2	3
2	616b3edead37240b7afc4034	3	1	5	1	13
3	616b3edead37240b7afc4035	4	2	3	3	3
4	616b3edead37240b7afc4036	5	2	5	1	13

BASE DE DATOS EN LA NUBE MONGODB ATLAS

Juank_'s Org - 2021-...

Access Manager

Billing

All ClustersGet HelpJuank_

Herramientas de bases ...

Atlas

Realm

Charts

DEPLOYMENT

Databases

Triggers

Data Lake

SECURITY

Database Access

Network Access

Advanced

JUANK_'S ORG - 2021-10-16 > HERRAMIENTAS DE BASES DE DATOS VIU 02 > DATABASES

Cluster0

VERSION 4.4.9

REGION GCP Sao Paulo (southamerica-east1)

OverviewReal TimeMetricsCollectionsSearchProfilerPerformance AdvisorOnline Archive

DATABASES: 1COLLECTIONS: 9

VISUALIZE YOUR DATA

REFRESH

+ Create Database

NAMESPACES

love4pets

cliente

departamento

detalle_orden

empleado

mascota

orden

producto

proveedor

suministro

love4pets

DATABASE SIZE: 14.7KBINDEX SIZE: 196KBTOTAL COLLECTIONS: 9

CREATE COLLECTION

Collection Name	Documents	Documents Size	Documents Avg	Indexes	Index Size	Index Avg
cliente	15	3.03KB	207B	1	20KB	20KB
departamento	14	932B	67B	1	36KB	36KB
detalle_orden	33	3.06KB	95B	1	20KB	20KB
empleado	12	2.33KB	200B	1	20KB	20KB
mascota	10	1.48KB	152B	1	20KB	20KB
orden	24	1.97KB	84B	1	20KB	20KB
producto	8	891B	112B	1	20KB	20KB
proveedor	4	450B	113B	1	20KB	20KB
suministro	6	618B	103B	1	20KB	20KB