
TRABAJO TEMA 4:

SGBD NoSQL.

MONGODB

Alejandro Cárdenas Barranco

Mónica Calzado Granados

Jesús García León

Álvaro Rodríguez Gallardo

Juan Manuel Rodríguez Gómez

Doble Grado en Ingeniería Informática y Matemáticas

Diseño y Desarrollo de Sistemas de Información (Grupo 2)

Curso 2022 – 2023



**UNIVERSIDAD
DE GRANADA**

Índice

1. Nombre del Grupo y Distribución del Trabajo	1
2. Descarga e Instalación de MongoDB	1
3. Descripción Básica del DDL y DML de MongoDB	7
4. Ejemplo Práctico de Uso de MongoDB	8
5. Mecanismo de Conexión a MongoDB desde Python	13
6. Uso de MongoDB en el SI de DGIIM Hotel Paradise	14

1. Nombre del Grupo y Distribución del Trabajo

- **Nombre del Grupo:**

DGIIM Hotels International

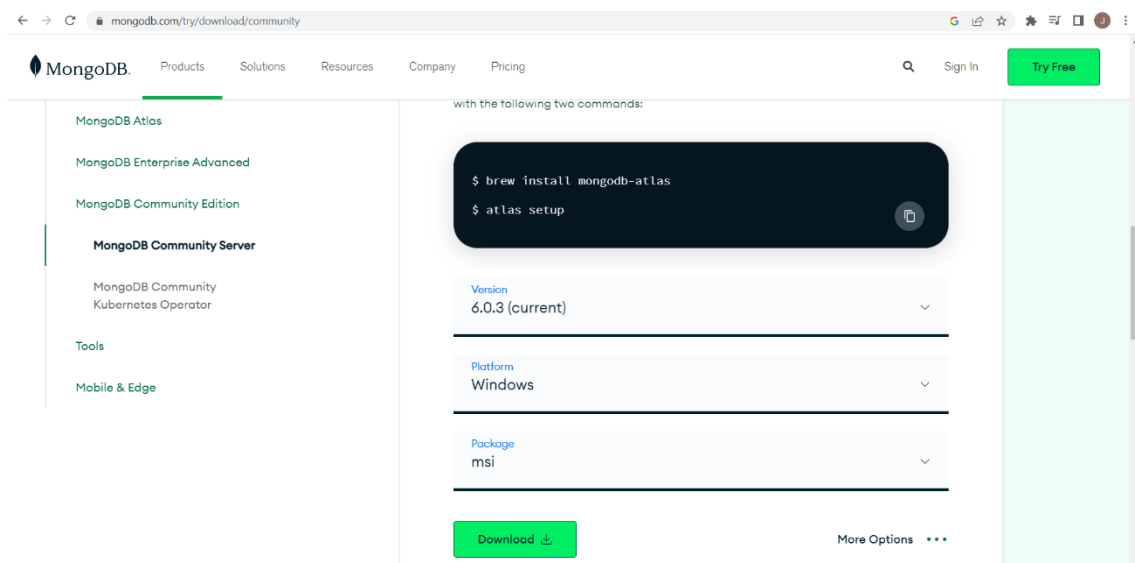
- **Distribución del Trabajo:** En pocas palabras, **todos nos hemos encargado de todo**. La metodología seguida ha sido conectarse a la plataforma Discord para poder comunicarnos todos e ir realizando el seminario a partir de las decisiones tomadas en grupo. Se puede descargar Discord gratuitamente en el siguiente enlace:

<https://discord.com/download>

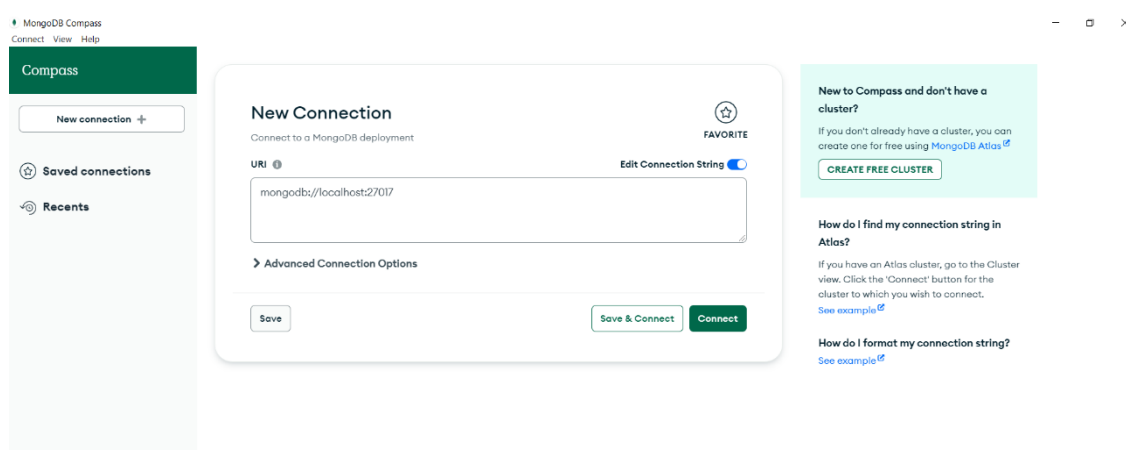
2. Descarga e Instalación de MongoDB

Comenzamos instalando MongoDB Server, cuyo instalador podemos descargar gratuitamente en el siguiente enlace de la página oficial de MongoDB:

<https://www.mongodb.com/try/download/community>



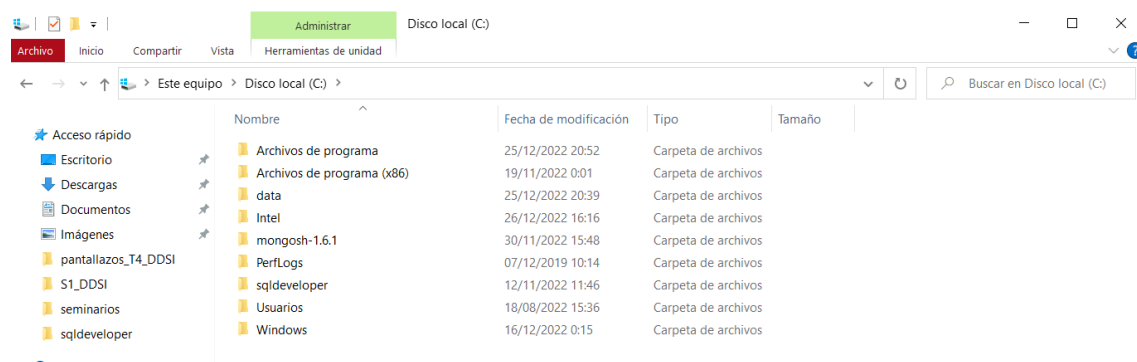
Una vez se descargue el instalador, lo ejecutamos y seguimos los pasos de instalación. Cabe destacar que da la opción de hacer una **instalación completa o personalizada** (en nuestro caso, hemos elegido la instalación completa). También podemos elegir si instalar **MongoDB Server como servicio o no** (si lo instalamos como servicio, siempre va a estar ejecutándose en segundo plano desde que arrancamos el ordenador hasta que lo apagamos. Esto es útil si vamos a usar MongoDB diariamente y así no hay que arrancarlo cada vez que queramos usarlo. En nuestro caso, sí vamos a instalarlo como servicio). Finalmente, cabe mencionar también que nos da la opción de instalar también **MongoDB Compass** (es una GUI para MongoDB, es decir, en vez de usar MongoDB en la Shell, se usa una interfaz gráfica. Es como el SQL Developer que se usa en Oracle).



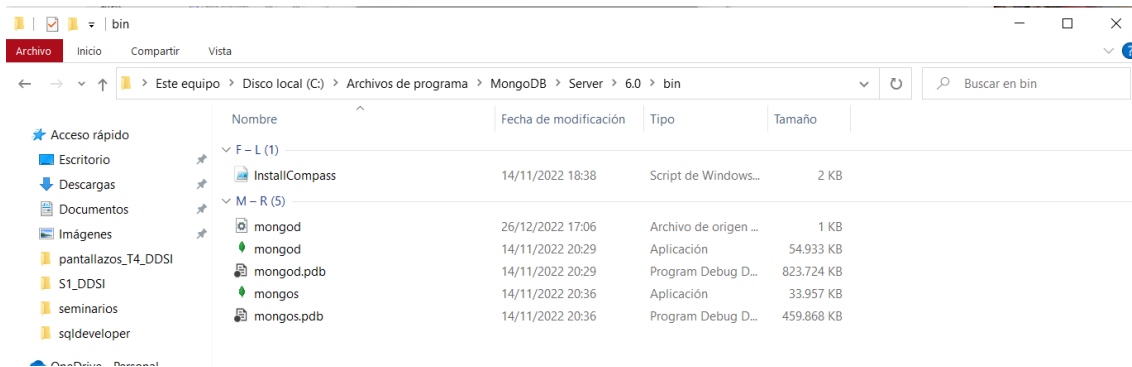
En nuestro caso, hemos instalado MongoDB Compass pero usaremos la **Shell de MongoDB** para mostrar mejor las sentencias de DDL y DML de MongoDB. La Shell hay que instalarla por separado (en el propio instalador de MongoDB Server lo menciona) mediante el siguiente enlace:

<https://www.mongodb.com/try/download/shell>

Una vez descarguemos el ZIP, tenemos que extraer la carpeta que contiene y la renombramos a *mongosh-1.6.1* para facilitar el nombre de dicha carpeta. Finalmente, movemos dicha carpeta desde el directorio donde esté a nuestro disco local (C:).

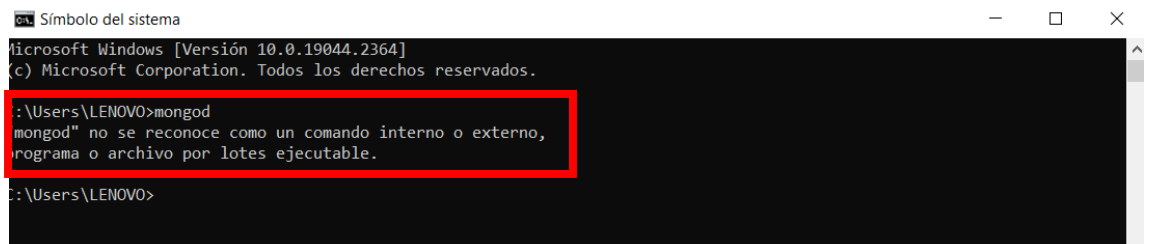


Por otro lado, para ver dónde está instalado MongoDB Server vamos a *Disco Local (C:)*
→ *Archivos de programa* → *MongoDB* → *Server* → *6.0* → *bin*.

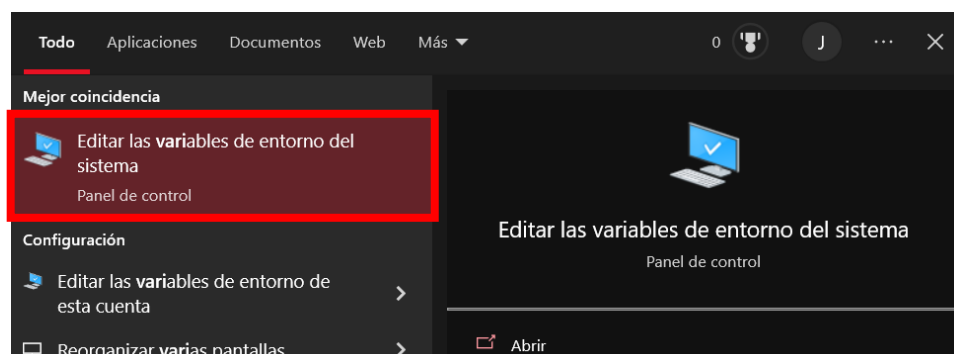


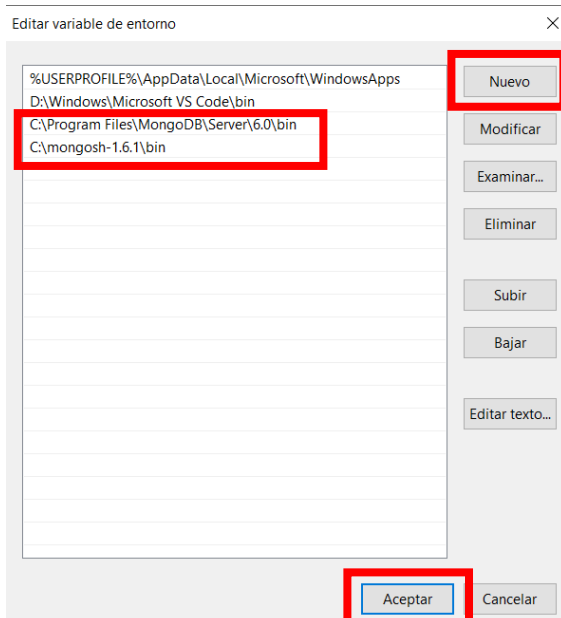
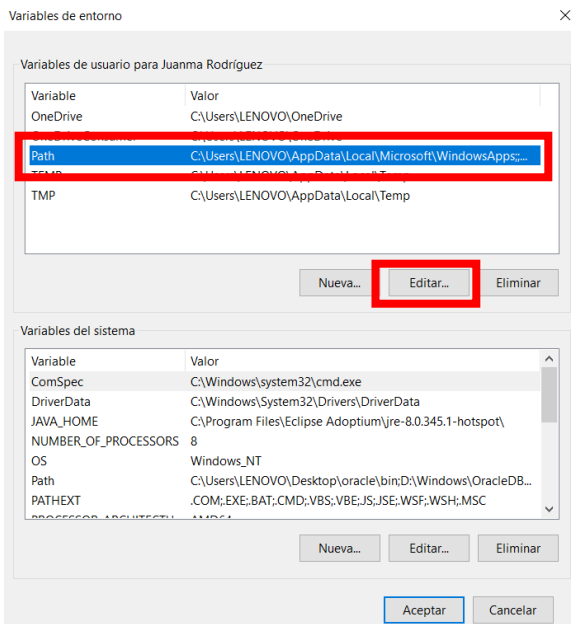
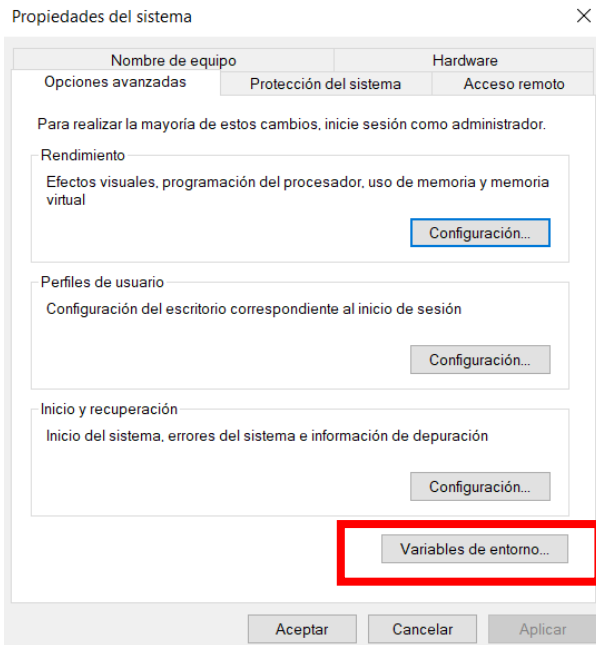
Para poder trabajar con MongoDB cómodamente en nuestro ordenador, necesitamos realizar un par de acciones extras:

- 1) Cada vez que queramos ejecutar MongoDB, tal y como está recién instalado, tendríamos que ir a los directorios mencionados anteriormente mediante la consola de Windows y ejecutar las aplicaciones desde ahí. Si lo intentáramos desde otros directorios, nos aparecería que los comandos no se reconocen.



Para poder ejecutar los comandos para iniciar MongoDB desde cualquier directorio de nuestro ordenador tenemos que definir nuevas variables de entorno en nuestro Windows. Para ello, escribimos lo siguiente en el buscador de Windows y seguimos lo indicado en las imágenes:





Una vez definidas las nuevas variables de entorno ya podemos ejecutar los comandos para iniciar MongoDB desde cualquier directorio en el que nos encontremos en la consola de Windows.

```
C:\Windows\system32\cmd.exe - mongod
Microsoft Windows [Versión 10.0.19044.2364]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\LENOVO>mongod
{"t":{"$date":"2022-12-25T20:44:56.738+01:00"},"s":"I", "c":"CONTROL", "id":23285, "ctx":"thread1","msg":"Automatica
lly disabling TLS 1.0, to force-enable TLS 1.0 specify --sslDisabledProtocols 'none'"}
{"t":{"$date":"2022-12-25T20:44:58.027+01:00"},"s":"I", "c":"NETWORK", "id":4915701, "ctx":"thread1","msg":"Initialize
d wire specification","attr":{"spec":{"incomingExternalClient":{"minWireVersion":0,"maxWireVersion":17},"incomingIntern
alClient":{"minWireVersion":0,"maxWireVersion":17},"outgoing":{"minWireVersion":6,"maxWireVersion":17},"isInternalClient":
true}}}
{"t":{"$date":"2022-12-25T20:44:58.028+01:00"},"s":"I", "c":"NETWORK", "id":4648602, "ctx":"thread1","msg":"Implicit T
CP FastOpen in use."}
{"t":{"$date":"2022-12-25T20:44:58.030+01:00"},"s":"I", "c":"REPL", "id":5123008, "ctx":"thread1","msg":"Successful
ly registered PrimaryOnlyService","attr":{"service":"TenantMigrationDonorService","namespace":"config.tenantMigrationDon
ors"}}
{"t":{"$date":"2022-12-25T20:44:58.030+01:00"},"s":"I", "c":"REPL", "id":5123008, "ctx":"thread1","msg":"Successful
ly registered PrimaryOnlyService","attr":{"service":"TenantMigrationRecipientService","namespace":"config.tenantMigratio
nRecipients"}}
{"t":{"$date":"2022-12-25T20:44:58.030+01:00"},"s":"I", "c":"REPL", "id":5123008, "ctx":"thread1","msg":"Successful
ly registered PrimaryOnlyService","attr":{"service":"ShardSplitDonorService","namespace":"config.tenantSplitDonors"}}
```

- 2) Si intentamos iniciar MongoDB con el comando *mongod* obtendremos el siguiente error:

```
Simbolo del sistema
C:\Program Files\MongoDB\Server\6.0\bin>mongod
{"t":{"$date":"2022-12-25T20:37:09.298+01:00"},"s":"I", "c":"CONTROL", "id":23285, "ctx":"thread1","msg":"Automatica
lly disabling TLS 1.0, to force-enable TLS 1.0 specify --sslDisabledProtocols 'none'"}
{"t":{"$date":"2022-12-25T20:37:10.536+01:00"},"s":"I", "c":"NETWORK", "id":4915701, "ctx":"thread1","msg":"Initialize
d wire specification","attr":{"spec":{"incomingExternalClient":{"minWireVersion":0,"maxWireVersion":17},"incomingIntern
alClient":{"minWireVersion":0,"maxWireVersion":17},"outgoing":{"minWireVersion":6,"maxWireVersion":17},"isInternalClient":
true}}}
{"t":{"$date":"2022-12-25T20:37:10.537+01:00"},"s":"I", "c":"NETWORK", "id":4648602, "ctx":"thread1","msg":"Implicit T
CP FastOpen in use."}
{"t":{"$date":"2022-12-25T20:37:10.538+01:00"},"s":"I", "c":"REPL", "id":5123008, "ctx":"thread1","msg":"Successful
ly registered PrimaryOnlyService","attr":{"service":"TenantMigrationDonorService","namespace":"config.tenantMigrationDon
ors"}}
{"t":{"$date":"2022-12-25T20:37:10.539+01:00"},"s":"I", "c":"REPL", "id":5123008, "ctx":"thread1","msg":"Successful
ly registered PrimaryOnlyService","attr":{"service":"TenantMigrationRecipientService","namespace":"config.tenantMigratio
nRecipients"}}
{"t":{"$date":"2022-12-25T20:37:10.539+01:00"},"s":"I", "c":"REPL", "id":5123008, "ctx":"thread1","msg":"Successful
ly registered PrimaryOnlyService","attr":{"service":"ShardSplitDonorService","namespace":"config.tenantSplitDonors"}}
{"t":{"$date":"2022-12-25T20:37:10.539+01:00"},"s":"I", "c":"CONTROL", "id":5945603, "ctx":"thread1","msg":"Multi threading initialized"}
{"t":{"$date":"2022-12-25T20:37:10.540+01:00"},"s":"I", "c":"CONTROL", "id":4615611, "ctx":"initandlisten","msg":"MongoDB starting","attr":{"pid":10956,"port":27017,"dbPath":"C:/data/db/","architecture":"64-bit","host":"DESKTOP-Q091AEU"}}
{"t":{"$date":"2022-12-25T20:37:10.540+01:00"},"s":"I", "c":"CONTROL", "id":23398, "ctx":"initandlisten","msg":"Target operating system minimum version","attr":{"targetMinOS":"Windows 7/Windows Server 2008 R2 SP1"}}
{"t":{"$date":"2022-12-25T20:37:10.541+01:00"},"s":"I", "c":"CONTROL", "id":23403, "ctx":"initandlisten","msg":"Build Info","attr":{"buildInfo":{"version":"6.0.3","gitVersion":"f8036813ae19817d31958965850193de067c516","modules":[],"allocator":"tcmalloc","environment":{"distmod":"windows","distarch":"x86_64","target_arch":"x86_64"}}}}
{"t":{"$date":"2022-12-25T20:37:10.541+01:00"},"s":"I", "c":"CONTROL", "id":51765, "ctx":"initandlisten","msg":"Operating System","attr":{"os":{"name":"Microsoft Windows 10","version":"10.0 (build 19044)"}}}
{"t":{"$date":"2022-12-25T20:37:10.542+01:00"},"s":"E", "c":"CONTROL", "id":20557, "ctx":"initandlisten","msg":"DBException in InitAndListen, terminating","attr":{"error":"NonExistentPath: Data directory C:/data/db/ not found. Create the missing directory or specify another path using (1) the --dbpath command line option, or (2) by adding the storage.dbpath option in the configuration file."}}
{"t":{"$date":"2022-12-25T20:37:10.543+01:00"},"s":"I", "c":"REPL", "id":4794602, "ctx":"initandlisten","msg":"Attempting to enter quiesce mode"}
{"t":{"$date":"2022-12-25T20:37:10.543+01:00"},"s":"I", "c":"REPL", "id":6371601, "ctx":"initandlisten","msg":"Shutting down the File (read thread pool)"}
{"t":{"$date":"2022-12-25T20:37:10.543+01:00"},"s":"I", "c":"SHARDING", "id":4784901, "ctx":"initandlisten","msg":"Shutting down the MirrorMaster"}
{"t":{"$date":"2022-12-25T20:37:10.543+01:00"},"s":"I", "c":"SHARDING", "id":4784902, "ctx":"initandlisten","msg":"Shutting down the WaitForMajorityService"}
```

Como indica dicho error, **se intenta buscar el directorio C:\data\db y este no existe**. Para solucionarlo, solo tenemos que crear dichos directorios siguiendo la misma ruta y con los nombres que se nos indican. Una vez lo hagamos, ya si se iniciará correctamente MongoDB (antes el comando se terminaba tras ejecutarlo y nos proporcionaba el error mencionado mientras que ahora veremos que no termina):

```
Simbolo del sistema
C:\Program Files\MongoDB\Server\6.0\bin>mongod
{"t":{"$date":"2022-12-25T20:33:34.063+01:00"},"s":"I", "c":"CONTROL", "id":23285, "ctx":"thread1","msg":"Automatica
lly disabling TLS 1.0, to force-enable TLS 1.0 specify --sslDisabledProtocols 'none'"}
{"t":{"$date":"2022-12-25T20:33:35.375+01:00"},"s":"I", "c":"NETWORK", "id":4915701, "ctx":"thread1","msg":"Initialize
d wire specification","attr":{"spec":{"incomingExternalClient":{"minWireVersion":0,"maxWireVersion":17},"incomingIntern
alClient":{"minWireVersion":0,"maxWireVersion":17},"outgoing":{"minWireVersion":6,"maxWireVersion":17},"isInternalClient":
true}}}
{"t":{"$date":"2022-12-25T20:33:35.376+01:00"},"s":"I", "c":"NETWORK", "id":4648602, "ctx":"thread1","msg":"Implicit T
CP FastOpen in use."}
{"t":{"$date":"2022-12-25T20:33:35.378+01:00"},"s":"I", "c":"REPL", "id":5123008, "ctx":"thread1","msg":"Successful
ly registered PrimaryOnlyService","attr":{"service":"TenantMigrationDonorService","namespace":"config.tenantMigrationDon
ors"}}
{"t":{"$date":"2022-12-25T20:33:35.378+01:00"},"s":"I", "c":"REPL", "id":5123008, "ctx":"thread1","msg":"Successful
ly registered PrimaryOnlyService","attr":{"service":"TenantMigrationRecipientService","namespace":"config.tenantMigratio
nRecipients"}}
{"t":{"$date":"2022-12-25T20:33:35.378+01:00"},"s":"I", "c":"REPL", "id":5123008, "ctx":"thread1","msg":"Successful
ly registered PrimaryOnlyService","attr":{"service":"ShardSplitDonorService","namespace":"config.tenantSplitDonors"}}
{"t":{"$date":"2022-12-25T20:33:35.380+01:00"},"s":"I", "c":"CONTROL", "id":5945603, "ctx":"thread1","msg":"Multi threading initialized"}
{"t":{"$date":"2022-12-25T20:33:35.380+01:00"},"s":"I", "c":"CONTROL", "id":4615611, "ctx":"initandlisten","msg":"Mongo
DB starting","attr":{"pid":7228,"port":27017,"dbPath":"C:/data/db/","architecture":"64-bit","host":"DESKTOP-Q091AEU"}}
{"t":{"$date":"2022-12-25T20:33:35.380+01:00"},"s":"I", "c":"CONTROL", "id":23398, "ctx":"initandlisten","msg":"Target
operating system minimum version","attr":{"targetMinOS":"Windows 7/Windows Server 2008 R2 SP1"}}
{"t":{"$date":"2022-12-25T20:33:35.380+01:00"},"s":"I", "c":"CONTROL", "id":23403, "ctx":"initandlisten","msg":"Build
Info","attr":{"buildInfo":{"version":"6.0.3","gitVersion":"f8036813ae19817d31958965850193de067c516","modules":[],"allocator":"tcmalloc","environment":{"distmod":"windows","distarch":"x86_64","target_arch":"x86_64"}}}}
{"t":{"$date":"2022-12-25T20:33:35.380+01:00"},"s":"I", "c":"CONTROL", "id":51765, "ctx":"initandlisten","msg":"Oper
ating System","attr":{"os":{"name":"Microsoft Windows 10","version":"10.0 (build 19044)"}}}
{"t":{"$date":"2022-12-25T20:33:35.380+01:00"},"s":"I", "c":"CONTROL", "id":21951, "ctx":"initandlisten","msg":"Outgoing
wire specification","attr":{"spec":{"incomingExternalClient":{"minWireVersion":0,"maxWireVersion":17},"incomingIntern
alClient":{"minWireVersion":0,"maxWireVersion":17},"outgoing":{"minWireVersion":6,"maxWireVersion":17},"isInternalClient":
true}}}
```

Finalmente, para ver que todo funciona correctamente, vamos a iniciar el Servicio y la Shell para trabajar con MongoDB. Para ello, abrimos una terminal de Windows y lanzamos el comando ***mongod***.

```
Microsoft Windows [Versión 10.0.19044.2364]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\LENOVO>mongod
{"t":{"$date":"2022-12-26T17:53:06.111+01:00"},"s":"I",  "c":"CONTROL",  "id":23285,   "ctx":"thread1","msg":"Automatically disabling TLS 1.0, to force-enable TLS 1.0 specify --sslDisabledProtocols 'none'"
{"t":{"$date":"2022-12-26T17:53:06.112+01:00"},"s":"I",  "c":"NETWORK",  "id":4915701, "ctx":"thread1","msg":"Initialized wire specification","attr":{"spec":{"incomingExternalClient":{"minWireVersion":0,"maxWireVersion":17},"incomingInternalClient":{"minWireVersion":0,"maxWireVersion":17},"outgoing":{"minWireVersion":6,"maxWireVersion":17},"isInternalClient":true}}}
{"t":{"$date":"2022-12-26T17:53:06.116+01:00"},"s":"I",  "c":"NETWORK",  "id":4648602, "ctx":"thread1","msg":"Implicit TCP FastOpen in use."
{"t":{"$date":"2022-12-26T17:53:06.117+01:00"},"s":"I",  "c":"REPL",     "id":5123008, "ctx":"thread1","msg":"Successfully registered PrimaryOnlyService","attr":{"service":"TenantMigrationDonorService","namespace":"config.tenantMigrationDonors"}}
{"t":{"$date":"2022-12-26T17:53:06.117+01:00"},"s":"I",  "c":"REPL",     "id":5123008, "ctx":"thread1","msg":"Successfully registered PrimaryOnlyService","attr":{"service":"TenantMigrationRecipientService","namespace":"config.tenantMigrationRecipients"}}
{"t":{"$date":"2022-12-26T17:53:06.118+01:00"},"s":"I",  "c":"REPL",     "id":5123008, "ctx":"thread1","msg":"Successfully registered PrimaryOnlyService","attr":{"service":"ShardSplitDonorService","namespace":"config.tenantSplitDonors"}}
{"t":{"$date":"2022-12-26T17:53:06.118+01:00"},"s":"I",  "c":"CONTROL",  "id":5945603, "ctx":"thread1","msg":"Multi threading initialized"}
{"t":{"$date":"2022-12-26T17:53:06.119+01:00"},"s":"I",  "c":"CONTROL",  "id":4615611, "ctx":"initandlisten","msg":"MongoDB starting","attr":{"pid":12752,"port":27017,"dbPath":"C:/data/db/","architecture":"64-bit","host":"DESKTOP-Q0914EUP"}}
{"t":{"$date":"2022-12-26T17:53:06.119+01:00"},"s":"I",  "c":"CONTROL",  "id":23398,   "ctx":"initandlisten","msg":"Target operating system minimum version","attr":{"targetMinOS":"Windows 7/Windows Server 2008 R2"}}
{"t":{"$date":"2022-12-26T17:53:06.119+01:00"},"s":"I",  "c":"CONTROL",  "id":23403,   "ctx":"initandlisten","msg":"Build Info","attr":{"buildinfo":{"version":"6.0.3","gitVersion":"f803681c3ae19817d3195896585019ade067c51e","modules":[],"allocator":"tcmalloc","environment":{"distmod":"windows","distarch":"x86_64","target_arch":"x86_64"}}}}
{"t":{"$date":"2022-12-26T17:53:06.119+01:00"},"s":"I",  "c":"CONTROL",  "id":531765,   "ctx":"initandlisten","msg":"Operating System","attr":{"os":{"name":"Microsoft Windows 10","version":"10.0 (build 19044)}}}
{"t":{"$date":"2022-12-26T17:53:06.120+01:00"},"s":"I",  "c":"CONTROL",  "id":21951,   "ctx":"initandlisten","msg":"Options set by command line","attr":{"options":{}}}
{"t":{"$date":"2022-12-26T17:53:06.124+01:00"},"s":"I",  "c":"STORAGE",  "id":22315,   "ctx":"initandlisten","msg":"Opening WiredTiger","attr":{"config":{"create,cache_size=7626M,session_max=33000,eviction=(threads_min=4,threads_max=4),config_base=false,statistics=(fast),log=(enabled=true,remove=true,path=journal,compressor=snappy),builtin_extension_config=(zstd:(compression_level=0)),file_manager=(close_idle_time=600,close_scan_interval=10,close_handle_minimum=2000),statistics_log=(wait=0),json_output=(error,message),verbose=(recovery_progress:1,checkpoint_progress:1,compact_progress:1,backup:0,checkpoint:0,compact:0,evict:0,history_store:0,recovery:0,rts:0,salvage:0,tiered:0,tiered:0,timestamp:0,transaction:0,verify:0,log[0],...)}}}
{"t":{"$date":"2022-12-26T17:53:06.213+01:00"},"s":"I",  "c":"STORAGE",  "id":4795906, "ctx":"initandlisten","msg":"WiredTiger opened","attr":{"durationMillis":89}}
{"t":{"$date":"2022-12-26T17:53:06.213+01:00"},"s":"I",  "c":"RECOVERY",  "id":23987,   "ctx":"initandlisten","msg":"WiredTiger recovery/restore","attr":{"recoveryTimestamp":{"timestamp":{"t":0,"i":0}}}}
{"t":{"$date":"2022-12-26T17:53:06.256+01:00"},"s":"W",  "c":"CONTROL",  "id":22120,   "ctx":"initandlisten","msg":"Access control is not enabled for the database. Read and write access to data and configuration is unrestricted","tags":{"startupWarnings"}}
{"t":{"$date":"2022-12-26T17:53:06.257+01:00"},"s":"W",  "c":"CONTROL",  "id":22140,   "ctx":"initandlisten","msg":"This server is bound to localhost. Remote systems will be unable to connect to this server. Start the server with --bind_ip address to specify which IP addresses it should serve responses from, or with --bind_ip_all to bind to all interfaces. If this behavior is desired, start the server with --bind_ip 22.0.0.1 to disable this warning","tags":{"startupWarnings"}}
{"t":{"$date":"2022-12-26T17:53:06.258+01:00"},"s":"I",  "c":"STORAGE",  "id":20320,   "ctx":"initandlisten","msg":"createCollection","attr":{"namespace":"admin.system.version","uuidDisposition":"provided","uuid":{"uuid":{"$uuid":"34165b89-c8ef-46c0-b538-46b4ff50871e"}},"options":{"uuid":{"$uuid":"34165b89-c8ef-46c0-b538-46b4ff50871e"}}}}}
{"t":{"$date":"2022-12-26T17:53:06.277+01:00"},"s":"I",  "c":"INDEX",    "id":20345,   "ctx":"initandlisten","msg":"Index build: done building","attr":{"buildUUID:null,"collectionUUID":{"uuid":{"$uuid":"34165b89-c8ef-46c0-b538-46b4ff50871e"},"namespace":"admin.system.version","index":{"id":"_id","ident":{"index-1--2970596417864135076"},"collectionIdent":{"collection-0--2970596417864135076"},"commitTimestamp:null}}}
{"t":{"$date":"2022-12-26T17:53:06.278+01:00"},"s":"I",  "c":"REPL",     "id":20459,   "ctx":"initandlisten","msg":"Setting featureCompatibilityVersion","attr":{"newVersion":"6.0"}}
{"t":{"$date":"2022-12-26T17:53:06.278+01:00"},"s":"I",  "c":"REPL",     "id":5853306, "ctx":"initandlisten","msg":"current featureCompatibilityVersion value","attr":{"featureCompatibilityVersion":"6.0","context":{"setFCV"}}}
{"t":{"$date":"2022-12-26T17:53:06.280+01:00"},"s":"I",  "c":"NETWORK",  "id":4915702, "ctx":"initandlisten","msg":"Updated wire specification","attr":{"oldSpec":{"incomingExternalClient":{"minWireVersion":0,"maxWireVersion":17},"incomingInternalClient":{"minWireVersion":0,"maxWireVersion":17},"outgoing":{"minWireVersion":6,"maxWireVersion":17},"isInternalClient":true},"newSpec":{"incomingExternalClient":{"minWireVersion":0,"maxWireVersion":17},"incomingInternalClient":{"minWireVersion":17,"maxWireVersion":17},"outgoing":{"minWireVersion":17,"maxWireVersion":17},"isInternalClient":true}}}
{"t":{"$date":"2022-12-26T17:53:06.282+01:00"},"s":"I",  "c":"NETWORK",  "id":4915702, "ctx":"initandlisten","msg":"Updated wire specification","attr":{"oldSpec":{"incomingExternalClient":{"minWireVersion":0,"maxWireVersion":17},"incomingInternalClient":{"minWireVersion":17,"maxWireVersion":17},"outgoing":{"minWireVersion":17,"maxWireVersion":17},"isInternalClient":true},"newSpec":{"incomingExternalClient":{"minWireVersion":0,"maxWireVersion":17},"incomingInternalClient":{"minWireVersion":17,"maxWireVersion":17},"outgoing":{"minWireVersion":17,"maxWireVersion":17},"isInternalClient":true}}}
{"t":{"$date":"2022-12-26T17:53:06.283+01:00"},"s":"I",  "c":"REPL",     "id":5853306, "ctx":"initandlisten","msg":"current featureCompatibilityVersion value","attr":{"featureCompatibilityVersion":"6.0","context":{"setFCV"}}}
```

Luego, abrimos otra terminal de Windows (sin cerrar la terminal anterior) y ejecutamos el comando ***mongosh***.

```
mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000

C:\Users\LENOVO>mongosh
Current MongoDB Log ID: 63a9d216e395d4e975ef5760
Connecting to:      mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000&appName=mongosh+1.6.1
Using MongoDB:      6.0.3
Using Mongosh:      1.6.1

For mongosh info see: https://docs.mongodb.com/mongod-shell/

To help improve our products, anonymous usage data is collected and sent to MongoDB periodically (https://www.mongodb.com/legal/privacy-policy).
You can opt-out by running the disableTelemetry() command.

-----
The server generated these startup warnings when booting
2022-12-26T17:06:47.473+01:00: Access control is not enabled for the database. Read and write access to data and configuration is unrestricted
-----

Enable MongoDB's free cloud-based monitoring service, which will then receive and display metrics about your deployment (disk utilization, CPU, operation statistics, etc).

The monitoring data will be available on a MongoDB website with a unique URL accessible to you
and anyone you share the URL with. MongoDB may use this information to make product
improvements and to suggest MongoDB products and deployment options to you.

To enable free monitoring, run the following command: db.enableFreeMonitoring()
To permanently disable this reminder, run the following command: db.disableFreeMonitoring()

-----
test>
```


Por defecto se nos realizará la conexión de MongoDB a nuestro localhost (es decir, a nuestro ordenador). Con el comando **show dbs** podemos ver todas las bases de datos que tenemos:

```
test> show dbs
admin   48.00 KiB
config 60.00 KiB
local   72.00 KiB
test> _
```

3. Descripción Básica del DDL y DML de MongoDB

En la siguiente página web podemos encontrar diferentes comandos básicos del DDL y DML de MongoDB:

<https://geekflare.com/es/mongodb-queries-examples/>

Los más básicos a destacar son los siguientes:

- 1) **Crear colección** (las colecciones son como las tablas en un SGBD Relacional):

db.createCollection("Name", Options)

- 2) **Insertar uno o varios documentos en una colección** (los documentos son como las tuplas de una tabla en un SGBD Relacional):

db.collectionName.insert([{document1}, {document2}, {document3}, ...])

- 3) **Mostrar todos los documentos almacenados en una colección:**

db.collectionName.find()

- 4) Mostrar todos los documentos almacenados en una colección en función de una cierta condición:

db.collectionName.find({condition})

- 5) Actualizar valores de las claves en documentos ya existentes en una colección:

db.collectionName.update({KeyToUpdate}, {Set Command})

- 6) Eliminar uno o varios documentos de una colección que cumplen una condición de eliminación:

db.collectionName.deleteOne({DeletionCondition})

db.collectionName.deleteMany({DeletionCondition})

- 7) Eliminar una colección:

db.collectionName.drop()

4. Ejemplo Práctico de Uso de MongoDB

```
trabajoDDSI> db.createCollection("Clientes")
{ ok: 1 }
trabajoDDSI> db.createCollection("ActividadesOcio")
{ ok: 1 }
trabajoDDSI> db.createCollection("CartaRestaurante")
{ ok: 1 }
trabajoDDSI> _
```

```
trabajoDDSI> show collections
ActividadesOcio
CartaRestaurante
Clientes
trabajoDDSI>
```

```

trabajoDDSI> db.Clientes.insert( {dni: "12345678A", nombre: "Alejandro"} )
{
  acknowledged: true,
  insertedIds: { '0': ObjectId("63a9de3531ec10dac49692b5") }
}
trabajoDDSI> db.Clientes.insert( {dni: "12345678B", nombre: "Monica", apellidos: "Calzado Granados"} )
{
  acknowledged: true,
  insertedIds: { '0': ObjectId("63a9de5031ec10dac49692b6") }
}
trabajoDDSI> db.Clientes.insert( {dni: "12345678B", nombre: "Jesus", telefono: "666010203"} )
{
  acknowledged: true,
  insertedIds: { '0': ObjectId("63a9de6f31ec10dac49692b7") }
}

```

```

trabajoDDSI> db.Clientes.insertMany( [ {dni: "12345678C", nombre: "Alvaro", correo: "alvaro@gmail.com"}, {dni: "12345678D", nombre: "Juanma", apellidos: "Rodríguez Gómez"} ] )
{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId("63a9df7431ec10dac49692b9"),
    '1': ObjectId("63a9df7431ec10dac49692ba")
  }
}
trabajoDDSI> db.Clientes.find()
[
  {
    _id: ObjectId("63a9de3531ec10dac49692b5"),
    dni: '12345678A',
    nombre: 'Alejandro'
  },
  {
    _id: ObjectId("63a9de5031ec10dac49692b6"),
    dni: '12345678B',
    nombre: 'Monica',
    apellidos: 'Calzado Granados'
  },
  {
    _id: ObjectId("63a9de6f31ec10dac49692b7"),
    dni: '12345678B',
    nombre: 'Jesus',
    telefono: '666010203'
  },
  {
    _id: ObjectId("63a9df7431ec10dac49692b9"),
    dni: '12345678C',
    nombre: 'Alvaro',
    correo: 'alvaro@gmail.com'
  },
  {
    _id: ObjectId("63a9df7431ec10dac49692ba"),
    dni: '12345678D',
    nombre: 'Juanma',
    apellidos: 'Rodríguez Gómez'
  }
]
trabajoDDSI>

```

```

trabajoDDSI> db.ActividadesOcio.insert( {nombre: "Natacion", numPlazas: "50"} );
{
  acknowledged: true,
  insertedIds: { '0': ObjectId("63a9e04e31ec10dac49692bb") }
}
trabajoDDSI> db.ActividadesOcio.insert( {nombre: "MiniGolf", numPlazas: "10", horaInicio: "10:00", horaFin: "17:00"} );
{
  acknowledged: true,
  insertedIds: { '0': ObjectId("63a9e07d31ec10dac49692bc") }
}
trabajoDDSI> db.ActividadesOcio.insert( {nombre: "Yoga", numPlazas: "15", fecha: "26/12/2022"} );
{
  acknowledged: true,
  insertedIds: { '0': ObjectId("63a9e0b331ec10dac49692bd") }
}
trabajoDDSI> db.ActividadesOcio.find()
[
  {
    _id: ObjectId("63a9e04e31ec10dac49692bb"),
    nombre: 'Natacion',
    numPlazas: '50'
  },
  {
    _id: ObjectId("63a9e07d31ec10dac49692bc"),
    nombre: 'MiniGolf',
    numPlazas: '10',
    horaInicio: '10:00',
    horaFin: '17:00'
  },
  {
    _id: ObjectId("63a9e0b331ec10dac49692bd"),
    nombre: 'Yoga',
    numPlazas: '15',
    fecha: '26/12/2022'
  }
]
trabajoDDSI>

```

```

trabajoDDSI> db.CartaRestaurante.insert( {nombrePlato: "Ensalada César", precio: "7.50€", ingredientes: ["lechuga", "tomate", "pollo", "picatostes", "salsa césar", "queso brie"] } )
acknowledged: true,
insertedIds: { '0': ObjectId("63a9e16a31ec10dac49692be") }

trabajoDDSI> db.CartaRestaurante.insert( {nombrePlato: "Gambas al Ajillo", precio: "9.00€", alergenos: "crustáceos" } )
acknowledged: true,
insertedIds: { '0': ObjectId("63a9e1bc31ec10dac49692bf") }

trabajoDDSI> db.CartaRestaurante.insert( {nombrePlato: "Pizza Carbonara", precio: "11.00€", alergenos: "lácteos", valorEnergetico: "257Kcal/100g" } )
acknowledged: true,
insertedIds: { '0': ObjectId("63a9e21231ec10dac49692c0") }

trabajoDDSI> db.CartaRestaurante.find()

[
  {
    _id: ObjectId("63a9e16a31ec10dac49692be"),
    nombrePlato: 'Ensalada César',
    precio: '7.50€',
    ingredientes: [
      'lechuga',
      'tomate',
      'pollo',
      'picatostes',
      'salsa césar',
      'queso brie'
    ]
  },
  {
    _id: ObjectId("63a9e1bc31ec10dac49692bf"),
    nombrePlato: 'Gambas al Ajillo',
    precio: '9.00€',
    alergenos: 'crustáceos'
  },
  {
    _id: ObjectId("63a9e21231ec10dac49692c0"),
    nombrePlato: 'Pizza Carbonara',
    precio: '11.00€',
    alergenos: 'lácteos',
    valorEnergetico: '257Kcal/100g'
  }
]
trabajoDDSI>

```

```

trabajoDDSI> db.Clientes.find( {dni: "12345678B"} )
[
  {
    _id: ObjectId("63a9de5031ec10dac49692b6"),
    dni: '12345678B',
    nombre: 'Monica',
    apellidos: 'Calzado Granados'
  },
  {
    _id: ObjectId("63a9de6f31ec10dac49692b7"),
    dni: '12345678B',
    nombre: 'Jesus',
    telefono: '666010203'
  }
]
trabajoDDSI>

```

```

CA: mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000
trabajoDDSI> db.Clientes.update( {dni: "12345678B", nombre: "Jesus"}, {$set : {dni: "12345678Z"} } )
DeprecationWarning: Collection.update() is deprecated. Use updateOne, updateMany, or bulkWrite.
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
trabajoDDSI> db.Clientes.find( {dni: "12345678B"} )
[
  {
    _id: ObjectId("63a9de5031ec10dac49692b6"),
    dni: '12345678B',
    nombre: 'Monica',
    apellidos: 'Calzado Granados'
  }
]
trabajoDDSI> db.Clientes.find()
[
  {
    _id: ObjectId("63a9de3531ec10dac49692b5"),
    dni: '12345678A',
    nombre: 'Alejandro'
  },
  {
    _id: ObjectId("63a9de5031ec10dac49692b6"),
    dni: '12345678B',
    nombre: 'Monica',
    apellidos: 'Calzado Granados'
  },
  {
    _id: ObjectId("63a9de6f31ec10dac49692b7"),
    dni: '12345678Z',
    nombre: 'Jesus',
    telefono: '666010203'
  },
  {
    _id: ObjectId("63a9df7431ec10dac49692b9"),
    dni: '12345678C',
    nombre: 'Alvaro',
    correo: 'alvaro@gmail.com'
  },
  {
    _id: ObjectId("63a9df7431ec10dac49692ba"),
    dni: '12345678D',
    nombre: 'Juanma',
    apellidos: 'Rodríguez Gómez'
  }
]

```

```

CA: mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000
trabajoDDSI> db.CartaRestaurante.find( {alergenos : {$exists : true} } )
[
  {
    _id: ObjectId("63a9e1bc31ec10dac49692bf"),
    nombrePlato: 'Gambas al Ajillo',
    precio: '9.00€',
    alergenos: 'crustáceos'
  },
  {
    _id: ObjectId("63a9e21231ec10dac49692c0"),
    nombrePlato: 'Pizza Carbonara',
    precio: '11.00€',
    alergenos: 'lácteos',
    valorEnergetico: '257Kcal/100g'
  }
]
trabajoDDSI> 

```

Cit. mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTime

```
trabajoDDSI> db.ActividadesOcio.deleteOne( {nombre: "Yoga"} )
{ acknowledged: true, deletedCount: 1 }
trabajoDDSI> db.ActividadesOcio.find()
[
  {
    _id: ObjectId("63a9e04e31ec10dac49692bb"),
    nombre: 'Natacion',
    numPlazas: '50'
  },
  {
    _id: ObjectId("63a9e07d31ec10dac49692bc"),
    nombre: 'MiniGolf',
    numPlazas: '10',
    horaInicio: '10:00',
    horaFin: '17:00'
  }
]
trabajoDDSI>
```

Cit. mongosh mongodb://127.0.0.1:27017/?directConnection=

```
trabajoDDSI> db.ActividadesOcio.drop()
true
trabajoDDSI> show collections
CartaRestaurante
Clientes
trabajoDDSI> █
```

5. Mecanismo de Conexión a MongoDB desde Python

Ya vimos en el Seminario 1 cómo conectarse a Oracle SQL desde Python. Ahora veamos brevemente cómo conectarse a MongoDB desde Python. Para ello, nos hemos informado mediante los siguientes enlaces:

<http://codigo-python.blogspot.com/2017/11/pymongo-parte-1.html>

https://www.youtube.com/playlist?list=PLCTD_CpMeEKR5cVnmTyFqUuzBOmEAjG4z

Básicamente, para la conexión a MongoDB desde Python se usa la **librería PyMongo**.

<https://pypi.org/project/pymongo/>

El código para realizar la conexión a la base de datos *TEST* de MongoDB sería el siguiente (dicho código se ha obtenido del primer enlace mostrado anteriormente):

```
#!/usr/bin/env python

import pymongo

MONGODB_HOST = '192.168.0.169'
MONGODB_PORT = '27017'
MONGODB_TIMEOUT = 1000
MONGODB_DATABASE = 'TEST'

URI_CONNECTION = "mongodb://" + MONGODB_HOST + ":" + MONGODB_PORT + "/"

try:
    client = pymongo.MongoClient(URI_CONNECTION, serverSelectionTimeoutMS=MONGODB_TIMEOUT, maxPoolSize=10)
    client.server_info()
    print 'OK -- Connected to MongoDB at server %s' % (MONGODB_HOST)
except pymongo.errors.ServerSelectionTimeoutError as error:
    print 'Error with mongoDB connection: %s' % error
except pymongo.errors.ConnectionFailure as error:
    print 'Could not connect to MongoDB: %s' % error
```

En el primer enlace también se nos informa acerca de los diferentes errores que podemos obtener al ejecutar el código anterior (dependiendo de si estamos en Windows o no) y cómo solucionarlo.

En dicho enlace también se proporciona un ejemplo de inserción de un documento en la colección USERS de la base de datos TEST. MongoDB guarda los documentos con una estructura BSON (Binary JSON, es una especificación similar a JSON), que, en términos de Python, podemos implementarlos como diccionarios. Luego, el código para realizar dicha inserción sería el siguiente:

```
database_entry = {}
database_entry['name'] = 'John'
database_entry['surname'] = 'Wick'
database_entry['city'] = 'New York'

# or equivalent database_entry={'name':'John', 'surname':'Wick', 'city':'New York'}

try:
    destination = 'USERS'
    collection = client[MONGODB_DATABASE][destination]
    collection.insert(database_entry)
    print "Data saved at %s collection in %s database: %s" % (destination, MONGODB_DATABASE, database_entry)
except Exception as error:
    print "Error saving data: %s" % str(error)
```

6. Uso de MongoDB en el SI de DGIIM Hotel Paradise

En nuestra opinión, tal y como tenemos diseñado el SI, no sería recomendable usar MongoDB en nuestro SI ya que como se puede observar en el ejemplo práctico de MongoDB realizado en el punto 4, no mantiene la unicidad de claves, como el DNI, el cual es imposible que se repita en la realidad y, sin embargo, introducimos dos clientes con el mismo DNI y no nos ha dado ningún error, lo cual es hace que perdamos la consistencia de nuestro sistema.

Sí es cierto que quizá podría usarse un sistema híbrido entre un SGBD Relacional y MongoDB si tuviéramos que realizar el SI de todo aquello del hotel (no solo algunos subsistemas del hotel como estamos realizando en prácticas). Por ejemplo, si implementáramos el subsistema del Restaurante del hotel, el cual tiene una carta online (con una gran variedad de platos de cada cual se almacenaría su nombre, sus ingredientes, sus alérgenos, imágenes del plato...) podríamos realizar el almacenamiento de datos de dicho subsistema mediante MongoDB. Pero, en vista del ejemplo práctico realizado en el punto 4, para el subsistema Clientes, por ejemplo, sería más recomendable usar un SGBD Relacional para garantizar así la consistencia en nuestro sistema.