

Asignatura	Datos del alumno	Fecha
Métodos de Captura y Almacenamiento de la Información	Apellidos: Mondragón Guadarrama	19-01-2020
	Nombre: José Carlos	

Actividades

Laboratorio: Administración de MongoDB con interfaz gráfica

Preparación del laboratorio

Descarga la versión gratuita del software MongoBooster (<http://mongobooster.com>) para la administración de bases de datos MongoDB. En la página del proveedor podrás seleccionar la versión del sistema operativo que utilices habitualmente: Windows, MacOS, Linux.

Descripción del laboratorio

El objetivo de laboratorio es la adquisición de las destrezas básicas de generación de bases de datos MongoDB con una aplicación de administración con interfaz gráfica. Para ello, se proporcionará un catálogo de datos con una estructura de datos no basada en JSON y un conjunto de *queries* Mongo. Deberás estructurar una base de datos e insertar los datos proporcionados de forma que las *queries* tengan el resultado esperado.

Entrega del laboratorio

Se deberá entregar:

- » Un *dump* de la base de datos generada.
- » Un informe documentando los pasos seguidos, en el que además enumeres tu valoración personal sobre ventajas e inconvenientes de usar una herramienta GUI.

Asignatura	Datos del alumno	Fecha
Métodos de Captura y Almacenamiento de la Información	Apellidos: Mondragón Guadarrama	19-01-2020
	Nombre: José Carlos	

Desarrollo

Para el desarrollo del presente documento se optó por un esquema de tres fases descrito a continuación:

	Parsing & Loading	Operations	Reporting
Technology	Node JS	Mongo - NoSQLBooster	Word
Language	Javascript	Javascript	-
Decision	No Off the shelf software to understand correct CSV parsing Generate a tool to be used afterwards Have control over the data TYPE exported (Currently, number and string) Proposed tool for managing instructions Worked on NoSQL booster as direct interface to test performance and usability		

Nota: el idioma inglés se elige dentro de consultas, programación y “README” para generar interacciones más fáciles en plataformas de repositorios como github.

Fase 0: Requerimientos

Instalación de :

- Node.js (<https://nodejs.org/en/>)
- MongoDB (<https://www.mongodb.com/download-center/community>)
- NoSQLBooster (<https://nosqlbooster.com/downloads>)

Este documento se generó durante enero 2020, por lo que las últimas versiones o superiores a dicho mes son adecuadas.

Plataforma: todos los siguientes experimentos se desarrollaron sobre Windows 10 con actualizaciones incrementales hasta la 1903.

Los archivos CSV iniciales se descargaron desde el repositorio de archivos de UNIR, basado en el requerimiento del laboratorio1 (documento pdf).

Nombre	Fecha de creación	Fecha de modificación	Modificado por	Tamaño
Agriculture horticulture informat...	15:22	15:22	BARBARO JO...	287 bytes
Agriculture land-use information ...	15:22	15:22	BARBARO JO...	566 bytes
Agriculture livestock information...	15:22	15:22	BARBARO JO...	1 KB
Business demography enterprise...	15:22	15:22	BARBARO JO...	3 KB
Business demography enterprise...	15:22	15:22	BARBARO JO...	10 KB
Business operations rates, activit...	15:22	15:22	BARBARO JO...	1 KB
LEED estimates of filled jobs, qua...	15:22	15:22	BARBARO JO...	755 bytes
LEED worker turnover rates, qua...	15:22	15:22	BARBARO JO...	632 bytes

Asignatura	Datos del alumno	Fecha
Métodos de Captura y Almacenamiento de la Información	Apellidos: Mondragón Guadarrama	19-01-2020
	Nombre: José Carlos	

Fase 1: Traducción y carga (parsing & loading)

El primer elemento que se tuvo que implementar fue la carga de los datos dentro de la base de datos NO-SQL, mongo. Cabe destacar que ya hubiera sido SQL o No SQL, los pasos son parecidos puesto que la carga de estos datos en SQL tendría que a ver generado sentencias INSERT que pudieran ejecutarse en cualquier DBMS.

Basado en stacks comunes y populares de desarrollo

(<https://www.ibm.com/cloud/learn/mean-stack-explained>) se escogieron las tecnologías.

La herramienta de traducción generada se ejecuta como cualquier utilidad del sistema (siempre que estén instalados y configurados los marcos de la Fase 0), de manera que puede recibir un documento CSV como entrada (input) y exportar un archivo en formato json validado (parámetro json de salida). Una llamada común sería la siguiente:

```
node ./converter/index.js input=AgricultureData.csv json=AgricultureObjects.json
```

En donde claramente se observa el archivo de entrada (input=archivo.csv) y el nombre del archivo de salida (json=archivodeseado.json).

El programa identifica las entradas adecuadas a partir de la función “checkOutput” en donde separa el valor de la llave como regreso de dicha operación. El parámetro indica la llave/parámetro de interés. (Orden de algoritmo para peor caso $O(N)*O(M)$ donde N es el número de argumentos colocados en la línea de comando y M es la cantidad de caracteres más larga de combinación llave-valor en la llamada original).

Una vez que ha verificado ambas entradas necesarias (si no existiera input el programa simplemente falla, mientras que si no existe la salida json, escribe la salida a la consola), divide el archivo de entrada en 2, la primer línea (encabezado) o el resto de las líneas que son los datos (de acuerdo al estándar de la IETF RFC 4180

<https://tools.ietf.org/html/rfc4180#page-2>). Es importante mencionar que para que este programa funcione se requiere la línea de encabezados, en suma por que el siguiente elemento de la arquitectura es cargarlo dentro de bases de datos que requieren de los nombres de los campos.

Una vez que se han dividido las líneas (y los encabezados han sido divididos en un arreglo con la función SPLIT, siempre y cuando los encabezados no cuenten con comas no separadores) se pasará la referencia a este arreglo hacia todos los datos que se generarán como un objeto dentro de una estructura de datos lineal. Aunque se optó por un desarrollo con objetos, la decisión fue únicamente para llevar el programa a futuro, dado que la misma operación es posible con un paradigma estructurado.

Asignatura	Datos del alumno	Fecha
Métodos de Captura y Almacenamiento de la Información	Apellidos: Mondragón Guadarrama	19-01-2020
	Nombre: José Carlos	

Las operaciones anteriores son parte básica de cualquier programa de validación de entradas. El algoritmo de traducción importante se integra como la siguiente etapa y sigue el siguiente orden:

Algoritmo general

1. Por cada línea de datos
 - a. aplicar función traducir(línea de datos)
 - b. Imprimir dato formateado en JSON en archivo de salida

Función Traducir(línea de datos)

1. Inicializar acumulador de dato actual, bandera de comillas, arreglo final de elementos, bandera de chequeo de números, contador de puntos decimales. Consideración, cualquier dato que no es número, es texto (caso de Fechas)
2. Por cada carácter de la línea completa de datos
 - a. Si es igual a “,” y no se encontró antes ninguna doble comilla, o el número de dobles comilla es par (se cerraron las dobles comillas bandera de comillas es falsa)
 - i. Si el dato hasta el momento es presumiblemente número (solo ha existido 0 o 1 punto decimal y dígitos del sistema decimal,, bandera de chequeo de números verdadera)
 1. Agrega el acumulador de dato actual convertido a número al arreglo final de elementos
 2. Reiniciar chequeo de número indicando presumiblemente verdadero
 3. Reiniciar puntos decimales a 0
 - ii. Si no:
 1. Agrega el acumulador de dato actual como cadena de texto al arreglo final de elementos
 2. Reiniciar chequeo de número indicando presumiblemente verdadero
 3. Reiniciar puntos decimales a 0
 - b. De cualquier otra manera
 - i. Si es igual a comilla y bandera de comillas es falsa
 1. Colocar bandera de comillas en verdadero
 - ii. Si es igual a comilla y bandera de comillas es verdadera
 1. Colocar bandera de comillas en falsa
 - iii. Si es igual a punto
 1. Aumentar contador de puntos decimales en 1
 2. Agregar carácter a acumulador de dato actual
 3. Si la cantidad de puntos decimales es mayor a 1
 - a. Colocar la bandera de chequeo de números en falso
 - iv. Si es igual a un número (los detalles de implementación se dejan al programa)
 1. Agregar carácter a acumulador de dato actual
 - v. Si no cumplió ninguna de las anteriores
 1. Colocar la bandera de chequeo de números en falso

Asignatura	Datos del alumno	Fecha
Métodos de Captura y Almacenamiento de la Información	Apellidos: Mondragón Guadarrama	19-01-2020
	Nombre: José Carlos	

2. Agregar carácter a acumular de dato actual

// comentario: el último dato no se alcanza a procesar por lo que se hace fuera del ciclo

3. Si la bandera de comillas es verdadera
 - a. Imprimir error: Las comillas se abrieron pero nunca se cerraron
4. Si no 3 y Si el último dato no es número
 - a. Agregar a arreglo final de elementos como cadena de texto
5. Si no 3 y no 4 :
 - a. Agregar arreglo final de elementos como número
6. Regresar arreglo final de elementos

// comentario: la agregación de elementos al arreglo final en calidad de texto, agrega el dato rodeado de doble comilla.

Una vez que se ha generado la estructura de datos, se imprime siguiendo el estándar JSON (<https://www.json.org/json-en.html>) indicando en el archivo un objeto que cuenta con los títulos de cada campo entre comillas, seguido de los dos puntos y el valor en texto (si fue cadena normal delimitado por doble comilla), número (sin doble comilla, directo) o null si es que el dato para el encabezado en la línea de datos, no se encontró.

En seguida una demostración de una línea de datos convertida a JSON:

CSV

<i>Year</i>	<i>Kiwifruit</i>	<i>Avocados</i>	<i>Wine_grapes</i>	<i>Onions</i>	<i>Squash</i>
2007	311.4	39.8	75.9	212	0.4

JSON

```
{ "Year":2007,"Kiwifruit":311.4,"Avocados":39.8,"Wine_grapes":75.9,"Onions":212,"Squash":0.4,"done":1 }
```

JSON (para el caso de tener datos como cadenas de texto y demostrando que el algoritmo actúa correctamente para datos con doble comilla cuyas “,” no son un separador)

```
{ "Year":2010,"MaoriBusinessStatus":"Maori SME","Industry":"Agriculture, Forestry and Fishing","Enterprises":45,"EmployeeCount":940,"done":1},{ "Year":2010,"MaoriBusinessStatus":"Maori SME","Industry":"Manufacturing","Enterprises":54,"EmployeeCount":840,"done":1 }
```

El ultimo valor “done” se agregó como una validación interna, no es importante para el resto de los reportes.

Importante: la estructura final del archivo se englobó en arreglos JSON, es decir:

```
[{d1},{d2},{d3}]
```

Asignatura	Datos del alumno	Fecha
Métodos de Captura y Almacenamiento de la Información	Apellidos: Mondragón Guadarrama	19-01-2020
	Nombre: José Carlos	

Una vez probadas las operaciones para traducir a un archivo JSON validado, se generó un archivo de procesamiento en lotes para poder generar el proceso cuantas veces se necesitará de manera rápida. He aquí el archivo (contenidos de .bat para windows) :

```
node ./converter/index.js input=Agriculture_horticulture_information_for_Maori_farms_annual.csv
json=Agriculture_horticulture_information_for_Maori_farms_annual.json
node ./converter/index.js input=Agriculture_land-use_information_for_Maori_farms_annual.csv json=Agriculture_land-
use_information_for_Maori_farms_annual.json
node ./converter/index.js input=Agriculture_livestock_information_for_Maori_farms_annual.csv
json=Agriculture_livestock_information_for_Maori_farms_annual.json
node ./converter/index.js input=Business_demography_enterprises_for_Maori_SMEs_annual.csv
json=Business_demography_enterprises_for_Maori_SMEs_annual.json
node ./converter/index.js input=Business_operations_rates_activities_annual.csv
json=Business_operations_rates_activities_annual.json
node ./converter/index.js input=Busines_demography_enterprises_for_Maori_authorities_annual.csv
json=Busines_demography_enterprises_for_Maori_authorities_annual.json
node ./converter/index.js input=LEED_estimates_of_filled_jobs_quarterly.csv
json=LEED_estimates_of_filled_jobs_quarterly.json
node ./converter/index.js input=LEED_worker_turnover_rates_quarterly.csv
json=LEED_worker_turnover_rates_quarterly.json
```

Con estos archivos listos, fue posible la importación a NoSQLBooster.

La herramienta cuenta con un importador de archivos json que genera el siguiente código

[VER ANEXO 1](#)

Independientemente de todo el código generado, cabe destacar la función parseBSON que es el procedimiento que finalmente estructura y prepara el dato final para integrarlo a la base de datos. La mayor parte de toda la operación generada es código pregenerado para poder revisar errores comunes.

Asignatura	Datos del alumno	Fecha
Métodos de Captura y Almacenamiento de la Información	Apellidos: Mondragón Guadarrama	19-01-2020
	Nombre: José Carlos	

Fase 2: Operaciones para reporte (Operations)

En seguida se enumeran los procedimientos a realizar en la base de datos:

1. Eliminar de todas las colecciones aquellos documentos que contienen campos con valores null.
2. Para la colección LEED worker turnover rates, crear un nuevo campo de tipo objeto para todos los documentos, con la información de los campos originales
"Maori_authority_worker_turnover_rate", "Maori_SME_worker_turnover_rate" y
"Maori_tourism_worker_turnover_rate".
3. Eliminar de todos los documentos de la colección, los tres campos originales
4. Para la colección Agriculture horticulture information, cambiar el tipo String a número flotante de los campos Kiwifruit, Avocados, Wine_grapes, Onions y Squash, para todos los documentos de la colección.
5. Para la colección Business operations rates, activities, obtener los documentos con Type = "Maori_tourism" y un Tourism_percent mayor de 80.
6. Para la colección "Business demography enterprises for Maori authorities", comprobar que el valor total del campo Enterprises es el valor correcto. Ese valor se encuentra en el documento con el campo Industry = "Total"
7. Convertir dos colecciones en una sola colección. Veamos la estructura de los documentos de las colecciones "Business demography enterprises for Maori SME" y "Business demography enterprises for Maori authorities"
8. Para la nueva colección calcular el total de empleados (atributo EmployeeCount) que incluya los totales de Maori authority y Maori SME. Los totales se encuentran en los documentos con el valor de Industry igual a Total.
9. Suponiendo que la variable leed se refiere a la colección "Maori_SME_filled_jobs", reescriba la siguiente consulta utilizando el operador \$eq.
leed.find({"Maori_SME_filled_jobs" : "7780.00"}).pretty()
10. Para la colección "Agriculture land-use information for Maori farms", encuentre los documentos con Year diferente de 2017 y proyecte exclusivamente los campos Year y Horticulture.

Para cada una de las operaciones se incluye el código utilizado, la descripción y una captura de pantalla mostrando el resultado.

Asignatura	Datos del alumno	Fecha
Métodos de Captura y Almacenamiento de la Información	Apellidos: Mondragón Guadarrama	19-01-2020
	Nombre: José Carlos	

Para cada una de las operaciones se incluye el código utilizado, la descripción y una captura de pantalla mostrando el resultado.

Fase 2. Operación 1: Eliminar de todas las colecciones aquellos documentos que contienen campos con valores null.

Dado que en la fase 1 se pudieron identificar todos aquellos elementos no correctos, a la base de datos no se integraron datos nulos. Sin embargo, la sentencia para borrar sería la siguiente:

```
db.agriculture_table.deleteMany({Year:null});
```

Solo válido en el caso de tener muchos datos con valores null. Se hizo una revisión de CSVs para identificar si podrían haber datos nulos, siendo negativa la respuesta.

Fase 2. Operación 2: Para la colección LEED worker turnover rates, crear un nuevo campo de tipo objeto para todos los documentos, con la información de los campos originales "Maori_authority_worker_turnover_rate", "Maori_SME_worker_turnover_rate" y "Maori_tourism_worker_turnover_rate"

En general, para esta operación, se hizo un recorrido por cada uno de los datos generando una nueva estructura que pudiera ser ingresada como un nuevo campo dentro de cada documento.

```
db.LEED_worker_turnover_rates_quarterly.find().forEach((it)=> {
  var newField = {
    Maori_authority_worker_turnover_rate:it.Maori_authority_worker_turnover_rate,
    Maori_SME_worker_turnover_rate:it.Maori_SME_worker_turnover_rate,
    Maori_tourism_worker_turnover_rate:it.Maori_tourism_worker_turnover_rate
  };

  db.LEED_worker_turnover_rates_quarterly.update({_id:it._id},
    {$set:
      {turnover_rate:newField}
    }
  )
});
```


Asignatura	Datos del alumno	Fecha
Métodos de Captura y Almacenamiento de la Información	Apellidos: Mondragón Guadarrama	19-01-2020
	Nombre: José Carlos	

Cabe destacar que los nuevos campos son también una estructura JSON con el nombre del campo y su valor. Una vez que el nuevo dato está listo, se utiliza la función UPDATE para poder hacer el \$set del nuevo campo (con el nombre nuevo) en todos los documentos que incluyen el id igual al id al del documento que recién entregó los datos (variable id._id).

Posterior a la ejecución del código, la colección se ve de la siguiente manera:

Key	Value
(1) ObjectId("5e209676f1f313da5895542b")	{ 7 attributes }
_id	ObjectId("5e209676f1f313da5895542b")
Quarter	mar-13
Maori_authority_worker_turnover_rate	16.53
Maori_SME_worker_turnover_rate	18.11
Maori_tourism_worker_turnover_rate	20.42
done	1
turnover_rate	{ 3 attributes }
Maori_authority_worker_turnover_rate	16.53
Maori_SME_worker_turnover_rate	18.11
Maori_tourism_worker_turnover_rate	20.42
(2) ObjectId("5e209676f1f313da5895542c")	{ 7 attributes }
(3) ObjectId("5e209676f1f313da5895542d")	{ 7 attributes }

Donde claramente se puede ver el nuevo diseño con un documento embebido.

Fase 2. Operación 3 Eliminar de todos los documentos de la colección, los tres campos originales

Ya dentro de esta fase, bastaba la ejecución de una sola línea para eliminar el campo en los elementos originales. Se decidió usar la operación \$unset. Dicho objeto de entrada recibe los campos que deben ser borrados colocando un 0 para aquellos a borrar. El parámetro multi:true es muy importante para que lo ejecute en todo el documento, y no pare el algoritmo en el primero encontrado. El primer parámetro del update se encuentra vacío por que no se necesita encontrar instancias específicas, si no que se aplica a toda la colección.

```
db.LEED_worker_turnover_rates_quarterly.update({}, { $unset: {
"Maori_authority_worker_turnover_rate":0,
"Maori_SME_worker_turnover_rate":0,
"Maori_tourism_worker_turnover_rate":0
} } , {multi:true});
```

Asignatura	Datos del alumno	Fecha
Métodos de Captura y Almacenamiento de la Información	Apellidos: Mondragón Guadarrama	19-01-2020
	Nombre: José Carlos	

La base de datos muestra como resultado:

▲ (1) ObjectId("5e209676f1f313da5895543e")	{ 4 attributes }
🔑 _id	ObjectId("5e209676f1f313da5895543e")
📅 Quarter	Dec-17
📌 done	1
▲ (3) turnover_rate	{ 3 attributes }
📌 Maori_authority_worker_turnover_rate	14.81
📌 Maori_SME_worker_turnover_rate	19.24
📌 Maori_tourism_worker_turnover_rate	19.58

Fase 2. Operación 4. Para la colección Agriculture horticulture information, cambiar el tipo String a número flotante de los campos Kiwifruit, Avocados, Wine_grapes, Onions y Squash, para todos los documentos de la colección.

Para ejecutar esta operación, se tuvo que regresar los valores de los campos a un formato String y posteriormente volverlos a convertir a Números. La razón es que la herramienta de traducción ya había generado los campos con el tipo de datos correcto (no colocar comillas para los números).

```
db.Agriculture_horticulture_information_for_Maori_farms_annual.find().forEach((it)=> {
  var kw = ""+(it.Kiwifruit);
  var av = ""+ (it.Avocados);
  var wg =""+ (it.Wine_grapes);
  var ons = ""+ (it.Onions);
  var sq = ""+ (it.Squash);

  db.Agriculture_horticulture_information_for_Maori_farms_annual.update({_id:it._id},{ $set
: {
    Kiwifruit:kw,
    Avocados:av,
    Wine_grapes:wg,
    Onios:ons,
    Squash:sq
  }});
});
```

Lo que resultó en la siguiente estructura:

Asignatura	Datos del alumno	Fecha
Métodos de Captura y Almacenamiento de la Información	Apellidos: Mondragón Guadarrama	19-01-2020
	Nombre: José Carlos	

(1) ObjectId("5e20966ff1f313da58955304")	{ 9 attributes }	Document
_id	ObjectId("5e20966ff1f313da58955304")	ObjectId
Year	2,007 (2.0K)	Double
Kiwifruit	311.4	String
Avocados	39.8	String
Wine_grapes	75.9	String
Onions	212	Double
Squash	0.4	String
done	1	Double
Onios	212	String

Posteriormente se ejecutó la siguiente instrucción que es una copia de la anterior pero usando la instrucción de javascript para convertir a Número (esta función utiliza las operaciones de “parsing” para intentar hacerlo con la mayor resolución posible).

```
db.Agriculture_horticulture_information_for_Maori_farms_annual.find().forEach((it)=> {
  var kw = Number(it.Kiwifruit);
  var av = Number(it.Avocados);
  var wg = Number(it.Wine_grapes);
  var ons = Number(it.Onions);
  var sq = Number(it.Squash);

  db.Agriculture_horticulture_information_for_Maori_farms_annual.update({_id:it._id},{ $set
: {
    Kiwifruit:kw,
    Avocados:av,
    Wine_grapes:wg,
    Onions:ons,
    Squash:sq
  }});
});
```

Finalmente la estructura de la colección se estableció como sigue:

(1) ObjectId("5e20966ff1f313da58955304")	{ 9 attributes }	Document
_id	ObjectId("5e20966ff1f313da58955304")	ObjectId
Year	2,007 (2.0K)	Double
Kiwifruit	311.4	Double
Avocados	39.8	Double
Wine_grapes	75.9	Double
Onions	212	Double
Squash	0.4	Double
done	1	Double
Onios	212	Double

Asignatura	Datos del alumno	Fecha
Métodos de Captura y Almacenamiento de la Información	Apellidos: Mondragón Guadarrama	19-01-2020
	Nombre: José Carlos	

Fase 2. Operación 5. Para la colección Business operations rates, activities, obtener los documentos con Type = "Maori_tourism" y un Tourism_percent mayor de 80.

Siendo una operación muy estándar de selección dentro de la colección, la operación ejecutada se basó en FIND colocando los parámetros para Type y Tourism_percent

```
db.Business_operations_rates_activities_annual.find(
  {Type:"Maori_tourism" ,
   Tourism_percent:{$gt:80} });
```

Mostrando los siguientes valores

1	ObjectId("5e209674f1f313da58955412")	2,014 (2.0K)	Maori_tourism	25	88	50	13	1
2	ObjectId("5e209674f1f313da58955414")	2,016 (2.0K)	Maori_tourism	21	93	50	7	1
3	ObjectId("5e209674f1f313da58955415")	2,017 (2.0K)	Maori_tourism	29	94	47	12	1
4	ObjectId("5e209674f1f313da58955416")	2,018 (2.0K)	Maori_tourism	21	84	53	11	1

Fase 2. Operación 6. Para la colección "Business demography enterprises for Maori authorities", comprobar que el valor total del campo Enterprises es el valor correcto. Ese valor se encuentra en el documento con el campo Industry = "Total"

El proceso para esta operación es un poco más complejo. Además de una operación de selección, se debía integrar la agregación para generar la suma. Si bien se podría haber integrado como un proceso de FIND y FOR-EACH, se eligió el marco de AGREGACION para que el responsable de la cuenta fuera Mongo de manera directa en lugar de un programa de aplicación. De esta forma, dicho marco funciona con al menos dos elementos particulares, un predicado \$match y un \$group. El primero funcionará para seleccionar aquellos documentos interesantes y la segunda para aplicar las funciones necesarias a cada uno de ellos integrando/acumulando la operación indicada.

El código generado fue el siguiente:

```
// get all the documents NOT EQUAL to "TOTAL", else, sum would include sum*2
(supposedly)
var res = db.Busines_demography_enterprises_for_Maori_authorities_annual.aggregate([
  { $match: {Industry: {$ne: "Total"}}},
  { $group: { _id:null, sum: {$sum: "$Enterprises"} }}
]);
// get all the documents that are equal to "TOTAL". Decided like this instead of find to
integrate all Total rows that could be inside de collection
var resT = db.Busines_demography_enterprises_for_Maori_authorities_annual.aggregate([
  {$match: {Industry:{$eq: "Total"}}},
  { $group: { _id:null, sum: {$sum: "$Enterprises"} }}
]);
```

Asignatura	Datos del alumno	Fecha
Métodos de Captura y Almacenamiento de la Información	Apellidos: Mondragón Guadarrama	19-01-2020
	Nombre: José Carlos	

```

});

var tot1 = 0;
while( res.hasNext() ){
    var r = res.next();
    console.log(r);
    // should have only one result
    tot1 = r.sum;
}

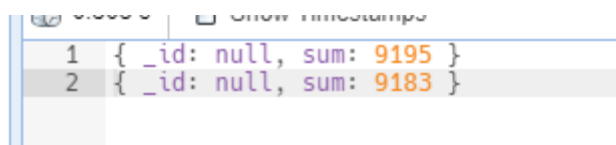
var tot2 = 0;
while( resT.hasNext() ){
    var r = resT.next();
    console.log(r);
    tot2 = r.sum;
}

if (tot1 != tot2){
    console.log("They're NOT equal");
}else{
    console.log("They ARE equal");
}

```

Dentro de este código ya se integra la revisión de ambos colocando código para verificar todos los resultados que hubiesen resultado de la agregación. Dado que solo debería resultar 1, el código bien podría resolverse como `res.next()[0] == resT.next()[0]` para verificar si son o no iguales.

El resultado, que claramente muestra que no son iguales, se muestra en la siguiente captura:



1	{ _id: null, sum: 9195 }
2	{ _id: null, sum: 9183 }

Siendo el primer resultado el total calculado por la agregación y el segundo el Total en el campo Total.

Fase 2. Operación 7. Convertir dos colecciones en una sola colección. Veamos la estructura de los documentos de las colecciones "Business demography enterprises for Maori SME" y "Business demography enterprises for Maori authorities"

Asignatura	Datos del alumno	Fecha
Métodos de Captura y Almacenamiento de la Información	Apellidos: Mondragón Guadarrama	19-01-2020
	Nombre: José Carlos	

Para esta operación se utilizó la opción merge del marco de agregación de mongo que permite integrar en una nueva colección, documentos de una existente (en este caso, todos).

Así, se utilizaron dos procedimientos iguales con parámetros diferentes:

```
db.Business_demography_enterprises_for_Maori_authorities_annual.aggregate(
[
  {$match: {}},
  {$merge: { into: "Business_demography_enterprises"}}
]
);

db.Business_demography_enterprises_for_Maori_SMEs_annual.aggregate(
[
  {$match: {}},
  {$merge: { into: "Business_demography_enterprises"}}
]
);
```

Para identificar que efectivamente el total de los documentos se integraron en la nueva colección, se obtuvo el reporte de la suma de ambos y se contrastó con el total de la nueva.

```
db.Busines_demography_enterprises_for_Maori_authorities_annual.count() +
db.Business_demography_enterprises_for_Maori_SMEs_annual.count() ==
db.Business_demography_enterprises.count() /// and it is ;D
```

Así la colección quedó de la siguiente manera:

71	ObjectId("5e209673f1f313da589553f9")	110	3	Public Administration and Safety	Maori SME	2018	1
72	ObjectId("5e209673f1f313da589553fa")	460	30	Education and Training	Maori SME	2018	1
73	ObjectId("5e209673f1f313da589553fb")	850	33	Health Care and Social Assistance	Maori SME	2018	1
74	ObjectId("5e209673f1f313da589553fc")	660	33	Arts and Recreation Services	Maori SME	2018	1
75	ObjectId("5e209673f1f313da589553fd")	190	24	Other Services	Maori SME	2018	1
76	ObjectId("5e209673f1f313da589553fe")	8,300 (8.3K)	453	Total	Maori SME	2018	1
77	ObjectId("5e209672f1f313da58955322")	1,150 (1.1K)	201	1 Agriculture	Maori authority	2010	1
78	ObjectId("5e209672f1f313da58955323")	60	57	2 Other primary industry	Maori authority	2010	1
79	ObjectId("5e209672f1f313da58955324")	200	306	3 Non-Residential Property Operators	Maori authority	2010	1
80	ObjectId("5e209672f1f313da58955325")	6,700 (6.7K)	327	4 All other industry	Maori authority	2010	1
81	ObjectId("5e209672f1f313da58955326")	8,100 (8.1K)	888	Total	Maori authority	2010	1
82	ObjectId("5e209672f1f313da58955327")	1,050 (1.1K)	195	1 Agriculture	Maori authority	2011	1
83	ObjectId("5e209672f1f313da58955328")	90	60	2 Other primary industry	Maori authority	2011	1
84	ObjectId("5e209672f1f313da58955329")	170	309	3 Non-Residential Property Operators	Maori authority	2011	1

Demostrando como hay documentos que previamente estaban en las otras colecciones.

Asignatura	Datos del alumno	Fecha
Métodos de Captura y Almacenamiento de la Información	Apellidos: Mondragón Guadarrama	19-01-2020
	Nombre: José Carlos	

Fase 2. Operación 8. Para la nueva colección calcular el total de empleados (atributo EmployeeCount) que incluya los totales de Maori authority y Maori SME. Los totales se encuentran en los documentos con el valor de Industry igual a Total.

La consulta generada para esta operación tomó en cuenta tres aspectos no identificados en la restricción original. Los totales de la industria se encuentran en la primer consulta que a través del marco de agregación, en donde se suma “EmployeeCount” de todos los documentos que si campo de Industria sea igual a “Total”. Por otra parte, y para identificar si es que los totales corresponden (parecido al ejercicio anterior), se obtienen los totales por industria, agrupados por cada nombre.

```
db.Business_demography_enterprises.aggregate([
  {$match: {Industry: "Total"}},
  {$group: { _id: null, employee_sum:{$sum: "$EmployeeCount" }}}
]);

// detail

db.Business_demography_enterprises.aggregate([
  {$match: {}},
  {$group: { _id: "$Industry", employee_sum:{$sum: "$EmployeeCount" }}}
]);
```

Mostrando la información:

Asignatura	Datos del alumno	Fecha
Métodos de Captura y Almacenamiento de la Información	Apellidos: Mondragón Guadarrama	19-01-2020
	Nombre: José Carlos	

_id	employee_sum
1 3 Non-Residential Property Operators	1,750 (1.8K)
2 4 All other industry	68,400 (68.4K)
3 Rental, Hiring and Real Estate Services	2,490 (2.5K)
4 Electricity, Gas, Water and Waste Services	370
5 2 Other primary industry	5,595 (5.6K)
6 Accommodation and Food Services	4,840 (4.8K)
7 Administrative and Support Services	4,110 (4.1K)
8 Wholesale Trade	2,320 (2.3K)
9 1 Agriculture	10,250 (10.3K)
10 Retail Trade	1,560 (1.6K)
11 Financial and Insurance Services	1,015 (1.0K)
12 Health Care and Social Assistance	7,060 (7.1K)
13 Total	152,300 (0.15M)
14 Construction	6,740 (6.7K)
15 Arts and Recreation Services	5,160 (5.2K)
16 Manufacturing	8,440 (8.4K)
17 Mining	99
18 Information Media and Telecommunications	1,085 (1.1K)
19 Other Services	1,680 (1.7K)
20 Education and Training	3,290 (3.3K)
21 Professional, Scientific and Technical Services	3,380 (3.4K)
22 Public Administration and Safety	1,005 (1.0K)
23 Transport, Postal and Warehousing	2,100 (2.1K)
24 Agriculture, Forestry and Fishing	9,590 (9.6K)

Así los totales son 152,300 mientras que el resto de la consulta identifica las sumas por industria. Cabe destacar que el valor total no es igual a la suma de la cuenta de empleados por cada industria (esta es la razón dentro del análisis por lo que se incluyó la consulta).

Fase 2. Operación 9. Suponiendo que la variable leed se refiere a la colección "Maori_SME_filled_jobs", reescriba la siguiente consulta utilizando el operador \$eq. leed.find({"Maori_SME_filled_jobs": "7780.00"}).pretty()

Suponiendo que la colección es LEED_estimates_of_filled_jobs_quarterly (única colección en donde existe Maori SME filled Jobs), la consulta reescrita es.

```
db.LEED_estimates_of_filled_jobs_quarterly.find( {Maori_SME_filled_jobs:{$eq:7780.00 }}
).pretty();
```

De manera que el resultado es:

LEED_estimates_of_filled_jobs_quarterly 0.336 s 1 Doc							
	_id	Quarter	Maori_authority_filled_jobs	Maori_SME_filled_jobs	Maori_tourism_filled_jobs	done	
1	ObjectId("5e209675f1f313da58955417")	mar-13	10,231 (10.2K)	7,780 (7.8K)	5,136 (5.1K)	1	

Asignatura	Datos del alumno	Fecha
Métodos de Captura y Almacenamiento de la Información	Apellidos: Mondragón Guadarrama	19-01-2020
	Nombre: José Carlos	

Fase 2. Operación 10. Para la colección "Agriculture land-use information for Maori farms", encuentre los documentos con Year diferente de 2017 y proyecte exclusivamente los campos Year y Horticulture.

Finalmente para seleccionar todos los documentos diferentes a 2017 y meramente con los campos year y horticulture:

```
db["Agriculture_land-use_information_for_Maori_farms_annual"].find( {"Year":{$ne:2017}}).projection({"_id":0,"Year":1,"Horticulture":1});
```

Cuya salida es:

	Year ↕	Horticulture ↕	
1	2006	1,146.2 (1.1K)	
2	2007	2,095.7 (2.1K)	
3	2008	2,645.5 (2.6K)	
4	2009	2,398.7 (2.4K)	
5	2010	2,360.7 (2.4K)	
6	2011	2,723 (2.7K)	
7	2012	2,973.4 (3.0K)	
8	2013	2,759.6 (2.8K)	
9	2014	2,850 (2.9K)	
10	2015	2,773.9 (2.8K)	
11	2016	2,667.7 (2.7K)	

Asignatura	Datos del alumno	Fecha
Métodos de Captura y Almacenamiento de la Información	Apellidos: Mondragón Guadarrama	19-01-2020
	Nombre: José Carlos	

Conclusión de reporte

Posterior a la ejecución de los “queries” y a la experiencia dentro del programa de NoSQLBooster, se pueden enmarcar las siguientes ventajas:

- El manejo de errores es más “amigable”: cuando se toman herramientas de línea de comandos, la salida en caso de problemas tiene poco formato y se debe analizar detalladamente. En una herramienta que provee una interface gráfica, se incrementa la productividad gracias a la facilidad de colores y formato.
- Existen herramientas incluidas: normalmente dentro de las herramientas de interface gráfica se agregan otras que de cualquier otra forma tendrían que ser integradas por elementos externos, dejando al desarrollador toda la responsabilidad de arquitectura.
- Existen visualizaciones con formato: mientras que Mongo si trata de mostrar la salida de queries de forma “amigable”, la realidad es que la pantalla de una línea de comandos no permite una gran flexibilidad. El contar con una herramienta de interface gráfica, agrega todos los elementos con los que cuenta el sistema operativo para mostrar información.
- Autocompletado de código: particularmente NoSQLBooster agrega datos, ayuda interactiva y completado de código a las clausulas haciendo mucho más fácil aprender y evitar errores.
- Capacidad de recuperar el trabajo: en herramientas de línea de comandos normalmente no se puede continuar el trabajo donde se dejó por que en realidad no hay forma de salvar todo lo que se hizo. Las herramientas de interface gráfica, permiten mantener el estado dado que son programas parte del sistema operativo que pueden, a través del OS API, guardar, recuperar y regenerar operaciones.
- Visualización de tablas: las líneas de comandos permiten comandos muy poderosos pero fallan en la constante visualización. Las interfaces graficas normalmente mostrarán detalles de la base de datos que incluso en otros métodos están escondidas.

Así mismo existen algunos inconvenientes:

- Ocultamiento de detalles: mientras que para trabajar dentro del ambiente la interface gráfica es adecuada, puede no llegar a mostrar detalles que en caso de utilizar la base de datos con un marco de desarrollo (como Node.js) son

Asignatura	Datos del alumno	Fecha
Métodos de Captura y Almacenamiento de la Información	Apellidos: Mondragón Guadarrama	19-01-2020
	Nombre: José Carlos	

necesarios. Como ejemplo, un error de conexión no se mostrará como si lo haría en un API de desarrollo.

- Acostumbramiento al ambiente: cuando no se cuentan con las herramientas en las que normalmente se trabaja, los usuarios pueden llegar a olvidar procedimientos “manuales”.
- Demasiada ayuda: dado que hay muchos detalles que se obvian y el ambiente completa información, la práctica que se obtiene sobre la herramienta original es menor y por ende es más difícil la integración.

Asignatura	Datos del alumno	Fecha
Métodos de Captura y Almacenamiento de la Información	Apellidos: Mondragón Guadarrama	19-01-2020
	Nombre: José Carlos	

ANEXO 1:

Código de importación para datos generado y ejecutado por NoSQLBooster

```
import * as fs from "fs";

const BATCH_SIZE = 2000;

const connection = "localhost";
const database = "maori";
let fromType = "file";

//idPolicy:
overwrite_with_same_id|always_insert_with_new_id|insert_with_new_id_if_id_exists|skip_documents_with_existing_id|abort_if_id_already_exists|drop_collection_first|log_errors
let toImportContents = [
  { content:
"C:\\Users\\Joe\\Documents\\Work\\master\\develop\\unir_master_data_science\\datacapture
class\\lab1\\JSON\\Agriculture_horticulture_information_for_Maori_farms_annual.json",
collection: "Agriculture_horticulture_information_for_Maori_farms_annual", idPolicy:
"overwrite_with_same_id" },
  { content:
"C:\\Users\\Joe\\Documents\\Work\\master\\develop\\unir_master_data_science\\datacapture
class\\lab1\\JSON\\Agriculture_land-use_information_for_Maori_farms_annual.json",
collection: "Agriculture_land-use_information_for_Maori_farms_annual", idPolicy:
"overwrite_with_same_id" },
  { content:
"C:\\Users\\Joe\\Documents\\Work\\master\\develop\\unir_master_data_science\\datacapture
class\\lab1\\JSON\\Agriculture_livestock_information_for_Maori_farms_annual.json",
collection: "Agriculture_livestock_information_for_Maori_farms_annual", idPolicy:
"overwrite_with_same_id" },
  { content:
"C:\\Users\\Joe\\Documents\\Work\\master\\develop\\unir_master_data_science\\datacapture
class\\lab1\\JSON\\Busines_demography_enterprises_for_Maori_authorities_annual.json",
collection: "Busines_demography_enterprises_for_Maori_authorities_annual", idPolicy:
"overwrite_with_same_id" },
  { content:
"C:\\Users\\Joe\\Documents\\Work\\master\\develop\\unir_master_data_science\\datacapture
class\\lab1\\JSON\\Business_demography_enterprises_for_Maori_SMEs_annual.json",
collection: "Business_demography_enterprises_for_Maori_SMEs_annual", idPolicy:
"overwrite_with_same_id" },
  { content:
"C:\\Users\\Joe\\Documents\\Work\\master\\develop\\unir_master_data_science\\datacapture
class\\lab1\\JSON\\Business_operations_rates_activities_annual.json", collection:
"Business_operations_rates_activities_annual", idPolicy: "overwrite_with_same_id" },
  { content:
"C:\\Users\\Joe\\Documents\\Work\\master\\develop\\unir_master_data_science\\datacapture
class\\lab1\\JSON\\LEED_estimates_of_filled_jobs_quarterly.json", collection:
"LEED_estimates_of_filled_jobs_quarterly", idPolicy: "overwrite_with_same_id" }
];
```

Asignatura	Datos del alumno	Fecha
Métodos de Captura y Almacenamiento de la Información	Apellidos: Mondragón Guadarrama	19-01-2020
	Nombre: José Carlos	

```

const totalImportResult = {
  result: {},
  fails: [],
}

for (let item of toImportContents) {
  totalImportResult.result[item.collection] = {
    nInserted: 0,
    nModified: 0,
    nSkipped: 0,
    failed: 0,
  };
}

function importFile({ path, batchSize, callback }) {
  const fs = require('fs'),
    readline = require('readline');

  return new Promise((resolve, reject) => {
    const fileSizeInBytes = fs.statSync(path).size;
    const digits = fileSizeInBytes > 1024 * 1024 * 1024 ? 1 : 0; //1G

    const rd = readline.createInterface({
      input: fs.createReadStream(path),
      crlfDelay: Infinity
    });

    const isValidEndLine = (line) => {
      if ((line[0] === "{" && (line[line.length - 1] === "}")) { //mongoexport
format
        return true;
      }

      if ((line === "},")) {
        return true;
      } //mongo shell export

      return false;
    }

    let objCounter = 0;
    let chunk = "";
    let readFileSize = 0;
    let hasError = false;

```

Asignatura	Datos del alumno	Fecha
Métodos de Captura y Almacenamiento de la Información	Apellidos: Mondragón Guadarrama	19-01-2020
	Nombre: José Carlos	

```

    rd.on('line', (line) => {
        if (hasError) return;

        chunk = chunk + line + "\n";
        readFileSize += (line + "\n").length;

        if (line && isValidEndLine(line)) {
            objCounter++;

            if (objCounter === batchSize) {
                callback(chunk, objCounter, ((readFileSize / fileSizeInBytes) *
100).toFixed(digits)).catch(err => {
                    hasError = true;
                    reject(err);
                    rd.close();
                });
                objCounter = 0;
                chunk = "";
            }
        }
    });

    rd.on("close", async () => {
        if (hasError) return;

        try {
            if (chunk.length > 0) {
                objCounter = objCounter > 0 ? objCounter : 1;
                await (callback(chunk, objCounter, 100));
            }

            resolve();

        } catch (err) {
            reject(err);
        }
    })

    rd.on("error", (err) => {
        reject(err)
    })
})

function importContent({ content, collection, idPolicy, percent }) {
    const collectionRst = totalImportResult.result[collection];

    let docs = mb.parseBSON(content);

```

Asignatura	Datos del alumno	Fecha
Métodos de Captura y Almacenamiento de la Información	Apellidos: Mondragón Guadarrama	19-01-2020
	Nombre: José Carlos	

```

    let writeResult = await(mb.writeToDb({ connection, db: database, collection, docs,
idPolicy }));

    let failed = writeResult.errors.length;
    let success = writeResult.nInserted + writeResult.nModified;

    collectionRst.nInserted += writeResult.nInserted;
    collectionRst.nModified += writeResult.nModified;
    collectionRst.nSkipped += writeResult.nSkipped;
    collectionRst.failed += failed;

    percent = (percent === undefined) ? 100 : percent;

    console.log(`import into ${database}.${collection}: ${percent}% ,
${collectionRst.nInserted} docs inserted, ${collectionRst.nModified} docs overwritten,
${collectionRst.failed} docs failed.`);
    if (failed) {
        console.log("Failed objects", writeResult.errors);
    }

    totalImportResult.fails = [...totalImportResult.fails, ...writeResult.errors];

    sleep(10)
}

for (let item of toImportContents) {
    if (item.idPolicy === "drop_collection_first") {
        mb.dropCollection({ connection, db: database, collection: item.collection });
        console.log(`drop collection ${database}.${item.collection}`);
    }

    console.log(`import into ${database}.${item.collection} start...`);

    if (fromType === "file") {
        await(importFile({
            path: item.content, batchSize: BATCH_SIZE,
            callback: (content, objCount, percent) => {
                return async(() => {
                    await(importContent({ content, collection: item.collection,
idPolicy: item.idPolicy, percent }));
                })()
            }
        )))
    }

    if (fromType === "clipboard") {
        await(importContent({ ...item, percent: 100 }))
    }
}

```

Asignatura	Datos del alumno	Fecha
Métodos de Captura y Almacenamiento de la Información	Apellidos: Mondragón Guadarrama	19-01-2020
	Nombre: José Carlos	

```

    }

    sleep(1000)
    console.log(`import into ${database}.${item.collection} finished.\n`);
}

if (toImportContents.length > 1) {
    console.log("");
    if (totalImportResult.result) {
        let succeeded = 0;
        let failed = 0;
        let collections = _.keys(totalImportResult.result);
        collections.forEach((key) => {
            let obj = totalImportResult.result[key];
            succeeded += obj.nInserted + obj.nModified;
            failed += obj.failed;
        });
        console.log(`${succeeded} document(s) have been imported into
${collections.length} collection(s).`);
        console.log(JSON.stringify(totalImportResult.result, null, 2));
        if (failed) {
            console.log(`${failed} document(s) haven't been imported, please check
failed list below.`);
        } else {
            console.log("All documents imported successfully.");
        }
    }

    if (totalImportResult.fails.length) {
        console.log("All failed objects", totalImportResult.fails);
    }
}

```