

Explorando MongoDB Compass

Índice

- 01 [Acciones iniciales](#)
- 02 [Guía Documents](#)
- 03 [Guía Aggregation y Pipelines](#)
- 04 [Guía Schema](#)
- 05 [Guía Explain](#)
- 06 [Guía Indexes](#)
- 07 [Guía Validation](#)



01

Acciones iniciales

Entrando en la herramienta

Al ingresar a la herramienta, el panel mostrará los datos de conexión, las bases de datos del sistema y aquellas que hayan sido creadas previamente usando el Shell.

Para recopilar los datos de conexión, simplemente hagan clic en el icono de flecha, indicado en la imagen.



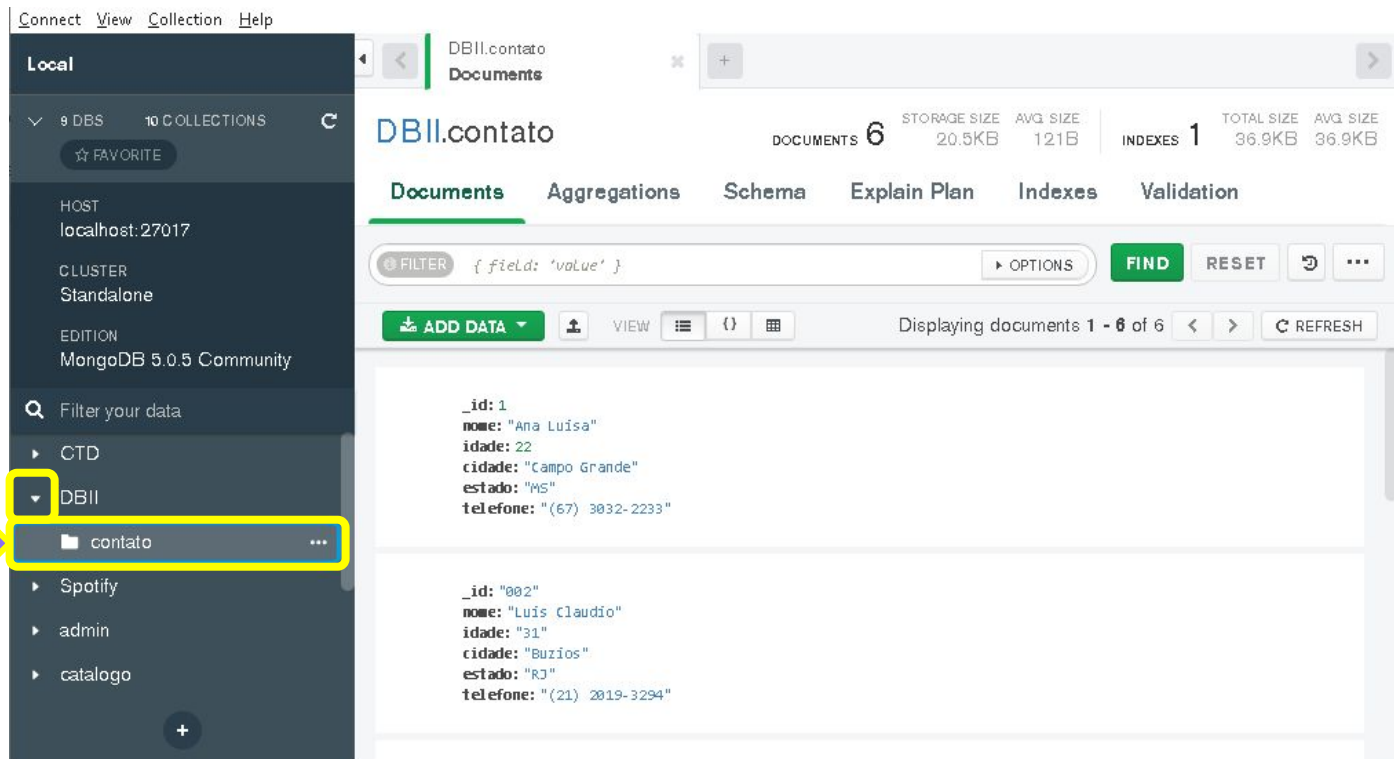
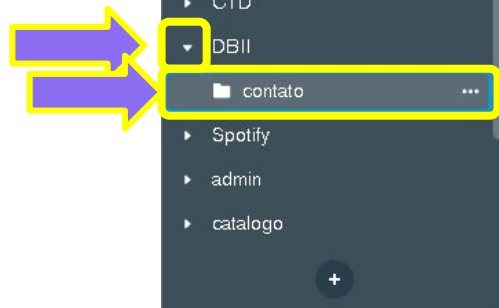
02

Menú Documents

Visualizando documentos de una colección

Hagan clic en el icono de flecha junto a la base de datos deseada (en el ejemplo, DBII).

Luego seleccionen una colección para ver los documentos.



Encontrando una base de datos

Para encontrar una base de datos más rápido, ingresen el nombre de la base de datos en el campo **Filter your data**.



Menú Documents

En el panel de la figura siguiente, visualicen el nombre de la base de datos y la colección seleccionada, el número de documentos e índices.

The screenshot displays the MongoDB Compass interface. At the top, the database name 'DBII.contato' is shown on the left. To its right, summary statistics are provided: 'DOCUMENTS 6' with 'STORAGE SIZE 20.5KB' and 'AVG SIZE 121B', followed by 'INDEXES 1' with 'TOTAL SIZE 36.9KB' and 'AVG SIZE 36.9KB'. Below these statistics is a horizontal menu with tabs: 'Documents' (which is highlighted with a green underline), 'Aggregations', 'Schema', 'Explain Plan', 'Indexes', and 'Validation'. The main area below the menu contains a search bar with a 'FILTER' button and the query '{ field: 'value' }'. To the right of the search bar are buttons for 'OPTIONS', 'FIND' (in green), 'RESET', a refresh icon, and a three-dot menu. At the bottom of the interface, there is a toolbar with an 'ADD DATA' button (in green), an upload icon, a 'VIEW' label, and three view mode icons (list, code, grid). On the right side of the toolbar, it says 'Displaying documents 1 - 6 of 6' with left and right navigation arrows, and a 'REFRESH' button.

| Category | Count | Storage Size | Avg Size |
|-----------|-------|--------------|----------|
| DOCUMENTS | 6 | 20.5KB | 121B |
| INDEXES | 1 | 36.9KB | 36.9KB |

Database: DBII.contato

Documents: 6 (Storage Size: 20.5KB, Avg Size: 121B)

Indexes: 1 (Total Size: 36.9KB, Avg Size: 36.9KB)

Filter: { field: 'value' }

Buttons: FILTER, OPTIONS, FIND, RESET, Refresh, ...

Toolbar: ADD DATA, VIEW, List View, Code View, Grid View

Displaying documents 1 - 6 of 6

Buttons: ADD DATA, VIEW, List View, Code View, Grid View, 1 - 6 of 6, REFRESH

Menú Documents

Utilicen el **filtro** para encontrar cualquier término que pertenezca a la colección.

Deben ingresar las claves, el nombre del campo y el valor: ej: **{estado: "DF"}**. Luego, hagan clic en **Find** para encontrar.

The screenshot shows the DBII.contato MongoDB interface. At the top, there are statistics: DOCUMENTS 6, STORAGE SIZE 20.5KB, AVG SIZE 121B, INDEXES 1, TOTAL SIZE 36.9KB, and AVG SIZE 36.9KB. Below these are navigation tabs: Documents (selected), Aggregations, Schema, Explain Plan, Indexes, and Validation. The main area has a FILTER input with the query {estado: "DF"}, an OPTIONS button, and FIND, RESET, and a refresh icon. Below the filter bar, there are buttons for ADD DATA, VIEW, and icons for list, JSON, and table views. The status bar indicates 'Displaying documents 1 - 1 of 1' with navigation arrows and a REFRESH button. The document content is displayed in a code block:

```
_id: "003"
nome: "Carlos Alberto"
idade: "26"
cidade: "Brasília"
estado: "DF"
telefone: "(61) 3233-3487"
```

Botón Options para filtrar el documento

Haciendo clic en **Options**, también tenemos la opción de filtrar nuestro documento, elegir las columnas que queremos que se muestren y ordenar nuestro resultado.



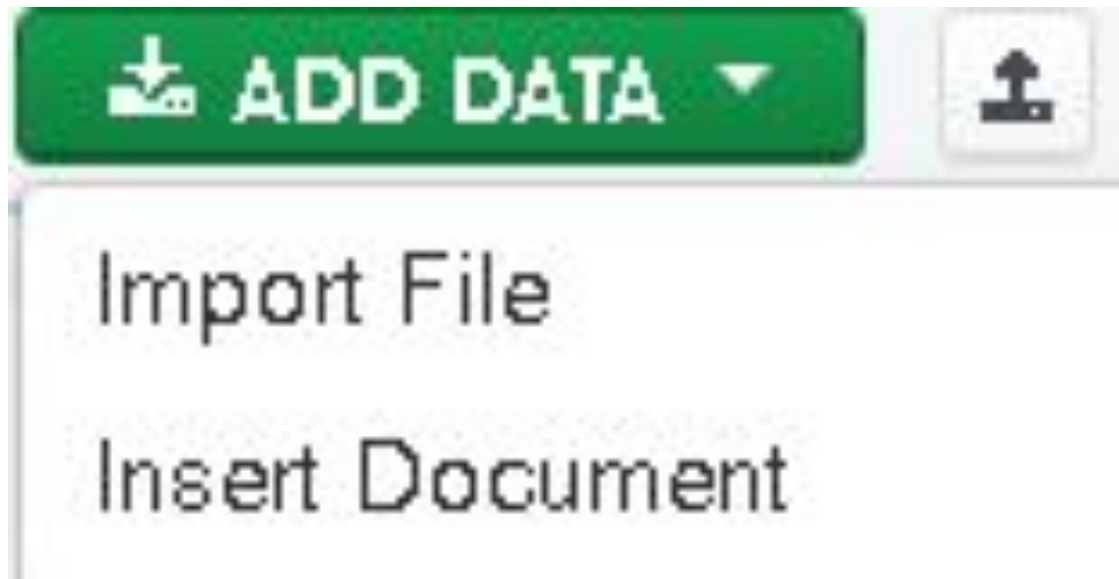
The image shows a MongoDB query builder interface. It features several configuration sections: **FILTER** with the query `{ field: 'value' }`, **PROJECT** with `{ field: 0 }`, **SORT** with `{ field: -1 } or [['field', -1]]`, and **COLLATION** with `{ locale: 'simple' }`. On the right side, there are settings for **MAX TIME MS** (60000), **SKIP** (0), and **LIMIT** (0). A purple rectangular box highlights the **Options** button in the top right corner, and a yellow arrow points directly to it.

Creando un nuevo documento

Para crear un nuevo documento, hagan clic en **ADD Data** y elijan una de las opciones:



- **Import File** - para importar un archivo.
- **Insert Document** - para crear un documento en la colección.

Al seleccionar la opción **Insert Document**, pueden elegir la manera en la que quieren incluir el documento: **modo JSON** o modo **Editor campo-por-campo**.



Ingresando documentos: formato JSON

Insert to Collection DBII.contato

VIEW  

La imagen muestra la
**inserción de un
documento en formato
Json.**

```
1 ▾ /**
2   * Paste one or more documents here
3   */
4 ▾ {
5   ▾   "_id": {
6       "$oid": "61face35ab8fed2d8b275564"
7     }
8 }
```

Ingresando documentos: modo Editor campo-por-campo

La imagen muestra la inserción de un documento en el modo de **editor de campo por campo**.

Insert to Collection DBII.contato

VIEW { } ≡

1

`_id: ObjectId("61face35ab8fed2d8b275564")`

ObjectId

Cancel

Insert

03

Guía Aggregation y Pipelines

Aggregation

Utilizamos el menú **Aggregation** para crear un **Pipeline de Agregado**.

Agregamos etapas para usar funciones o filtros en nuestra colección según sea necesario, como seleccionar los campos que queremos que se muestren, contar documentos, agrupar datos, concatenar campos y mucho más.

The screenshot shows the MongoDB Compass interface. At the top, the database name 'DBII.contato' is displayed. On the right, statistics are shown: 6 documents, 20.5KB storage size, 121B avg size, 1 index, 36.9KB total size, and 36.9KB avg size. The 'Aggregations' tab is selected in the top navigation bar. Below the navigation bar, there's a toolbar with buttons for 'Collation', 'Untitled', 'Save', and 'Sample Mode' (which is turned on). The main area shows a preview of 6 documents in the collection. A sample document is displayed on the right:

```
{
  "_id": 1,
  "nome": "Ana Luisa",
  "idade": 22,
  "cidade": "Campo Grande",
  "estado": "MS",
  "telefone": "(67) 3032-2233"
}
```

At the bottom, there's a 'Select...' dropdown and a 'Sample Mode' toggle (which is turned on). A note at the bottom right states: 'A sample of the aggregated results from this stage will be shown below'.

Aggregation

Ejemplo de implementación por etapas: utilizando la colección **contacto**, vamos a seleccionar la etapa **\$project**, donde definiremos los campos que queremos mostrar. Observen la imagen.



The screenshot shows the MongoDB Compass interface. On the left, the aggregation pipeline editor displays a `$project` stage with a toggle switch turned on. The pipeline definition is as follows:

```
1 ▾ /**
2   * specifications: The fields to
3   *   include or exclude.
4   */
5 ▾ {
6   _id: 0,
7   nome: 1,
8   idade: 1,
9   estado: 1,
10
11 }
```

On the right, the output window shows the result after the `$project` stage, displaying a sample of 6 documents. The first document is:

```
nome: "Ana Luísa"
idade: 22
estado: "MS"
```


Aggregation

Nuestra primera etapa ha sido definida. Solo seleccionamos los campos de nombre, edad y estado a mostrar.


Automáticamente, en el lado derecho, ya muestra cómo se verá la visualización.

Vamos a **agregar una etapa más**, haciendo clic en el botón **ADD STAGE**:



Aggregation

Vamos a definir, ahora, un filtro para nuestra visualización. Solo incluiremos **contactos mayores de 30 años**. Para ello utilizaremos la función **\$match**:



The screenshot shows a MongoDB aggregation pipeline editor. The left pane displays a query with a comment and a \$match stage. The right pane shows the output of the \$match stage, which is a sample of 3 documents. The output is a JSON document with the following fields:

```
1 ▾ /**  
2   * query: The query in MQL.  
3   */  
4 ▾ {  
5   idade: { $gt: 30 }  
6 }
```

Output after \$match stage ⓘ (Sample of 3 documents)

```
nome: "Luís claudio"  
idade: 31  
estado: "RJ"
```

Aggregation

Incluso podemos ordenar nuestra lista. Agreguemos una etapa más, ordenando la columna de nombre en orden ascendente. A continuación, pueden observar que los documentos se mostrarán en orden alfabético.



The screenshot shows a MongoDB aggregation pipeline editor. On the left, a code editor displays a pipeline with a single stage: `$sort` with the field `nome` and order `1` (ascending). The right panel shows the output after the `$sort` stage, displaying a sample of 3 documents. The first document shown is:

```
nome: "José Roberto"
idade: 35
estado: "SP"
```

Para cada operación que queramos agregar a nuestra consulta, creamos una etapa.

Aggregation

Ahora agrupemos los contactos por estado. En este ejemplo, mostramos la cantidad de contactos existentes, mayores de 30, en cada estado.

The screenshot shows a MongoDB aggregation pipeline editor. The top toolbar includes a dropdown menu, a folder icon, a plus icon, a 'COLLATION' button, a 'SAVE' button, and a 'SAMPLE MODE' toggle. The main interface is divided into two sections. The left section contains a code editor with the following aggregation pipeline:

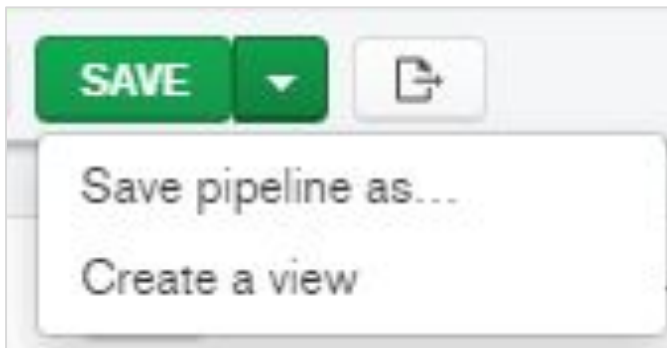
```
1 /**
2  * _id: The id of the group.
3  * fieldN: The first field name.
4  */
5 {
6   _id: "$estado",
7   Total: {
8     $sum: 1
9   }
10 }
```

The right section displays the output after the `$group` stage, showing a sample of 3 documents. The output is presented in two boxes, each representing a group:

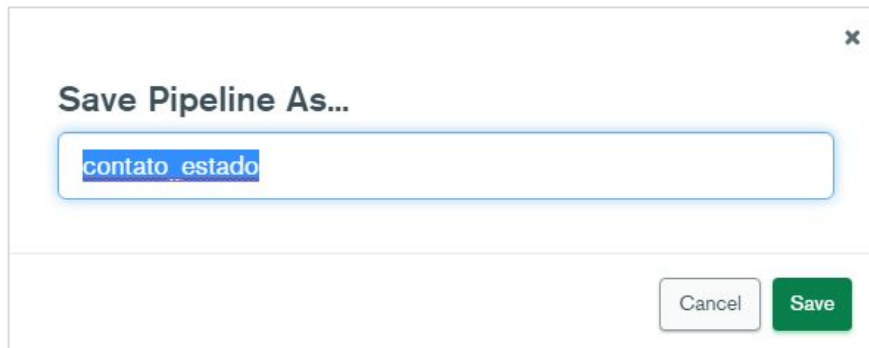
- Group 1: `_id: "MG"`, `Total: 1`
- Group 2: `_id: "RJ"`, `Total: 1`

Guardando un Pipeline

Ahora que hemos utilizado todos los filtros y funciones que creemos que son necesarios para nuestra consulta, tenemos la opción de guardar pipeline o crear una view. Miren las imágenes a continuación.



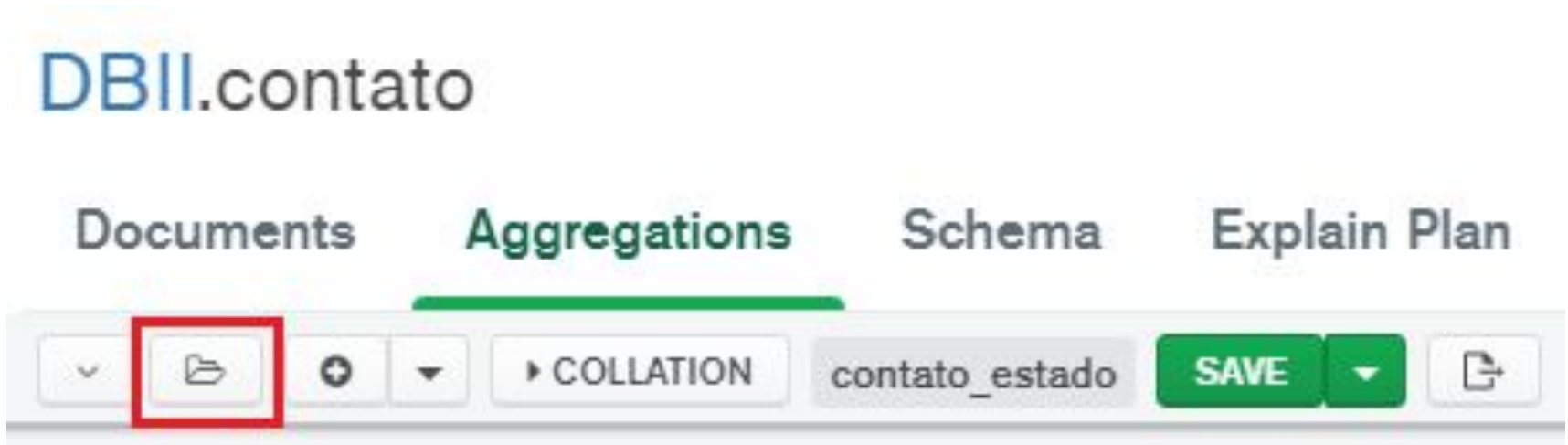
Seleccionando la opción **Save pipeline as...** podemos guardar nuestra consulta para usar en otro momento.



Elegí un nombre sugestivo y hacé clic en el botón **Save**.

Abriendo un Pipeline

Para abrir el pipeline que crearon, hagan clic en el ícono carpeta que se encuentra abajo de las pestañas, en el rincón superior izquierdo de la pantalla.



Abriendo un Pipeline

Seleccionen el pipeline deseado y hagan clic en OPEN. Ahora, deberán desplazarse por la pantalla para ver las etapas establecidas.




Creando una View

La opción **Create a view** permite crear una visualización a partir del resultado del pipeline, pero no guarda el pipeline propiamente dicho.



Creando una View

Ingresen un nombre para su view y hagan clic en **Create**. La view de los resultados del pipeline será creada en la misma base de datos en la que el pipeline fue creado.



Create a View

Enter a View Name

Cancel

Create

Visualizando los resultados

Después de crear la view de resultados de pipeline, observen en la base de datos utilizada que la view ya se muestra. Cuando sea necesario, simplemente hagan clic en él para que se muestre el resultado.



Exportando tu pipeline

El Compass aún permite que el pipeline sea exportado para un lenguaje de programación, para ser utilizado en sus proyecto. Elijan entre Java, Node, C# ou Python y aceleren su desarrollo.

También pueden simplemente copiar su código para ejecutarlo en MongoDB Shell.



Exportando tu pipeline

Para copiar el código del pipeline, hagan clic en el ícono **Export To Language**.

Luego hagan clic en el ícono Copiar, cierren la ventana y péguenlo en la terminal.

Para ejecutarlo, escriban el código antes: **db.collection.aggregate(** , cerrando el paréntesis al final del código).

Recuerden: **collection** es el nombre de la colección en la que están trabajando.

Export Pipeline To Language

My Pipeline:

```
1  [{
2    $group: {
3      _id: '$genero',
4      musicas: {
5        $sum: 1
6      },
7      titulo: {
8        $push: '$musica'
9      }
10   }
11  ]}]
```



Exportando tu pipeline

Para exportar el código del pipeline, hagan clic en el ícono **Export To Language**.

Seleccionen el lenguaje deseado y hagan clic en copiar. El código estará en el portapapeles. Abran sus proyectos y péguenlos en la ubicación deseada.

Opcional: incluyan declaraciones de importación y sintaxis del controlador si lo desean.

Marquen la opción **Include Import Statements** para incluir las declaraciones de importación necesarias para el lenguaje de programación seleccionado.

Marquen la opción **Include Driver Syntax** para incluir el código de la aplicación para el lenguaje de programación seleccionado.

Export Pipeline To: PYTHON 3

```
1  [
2  {
3    '$group': {
4      '_id': '$genero',
5      'musicas': {
6        '$sum': 1
7      },
8      'titulo': {
9        '$push': '$musica'
10     }
11   }
12 }
13 ]
```

☐ Include Import Statements

☐ Include Driver Syntax

CLOSE

04

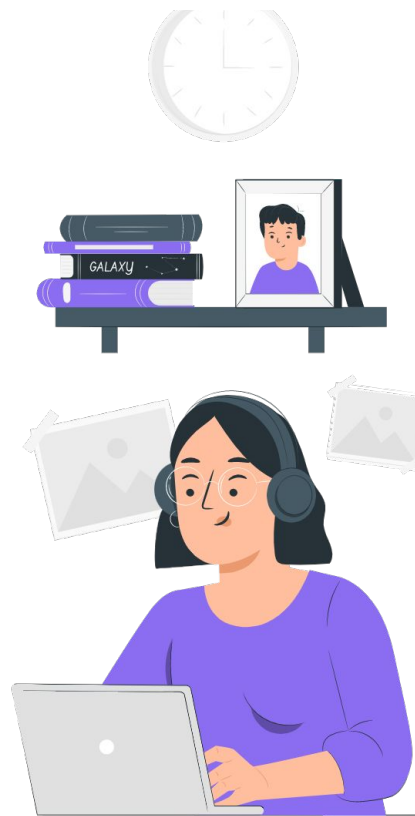
Guía Schema

Guía Schema

La guía **Schema** nos da una visión general del **tipo de datos** y de la **forma de los campos** en una **colección específica**.

Para ese análisis, no es necesario mostrar todos los datos de la colección. El Compass trabaja con **muestras**.

En nuestro ejemplo, analizaremos la colección contacto. Podrán verla en la imagen de la pantalla a continuación.



FILTER

{ field: 'value' }

► OPTIONS

ANALYZE

RESET



This report is based on a sample of 6 documents. ⓘ

_id

string

double



- 004
- 006
- 005

cidade

string



- Campo Grande

Colección contacto

Tené en cuenta que el análisis trae el nombre del campo, el tipo de datos y el contenido.

En el campo **_id**, se nos informa que tiempos algunos campos string y otros double.

FILTER

{ field: 'value' }

OPTIONS

FIND

RESET



ADD DATA



VIEW



Displaying documents 1 - 6 of 6



REFRESH

```
1  _id: 1
2  nome: "Ana Luisa"
3  idade: 22
4  cidade: "Campo Grande"
5  estado: "MS"
6  telefone: "(67) 3032-2233"
```

Double

String

Int32

String

String

String

CANCEL

UPDATE

```
1  _id: "002"
2  nome: "Luís Claudio"
3  idade: 31
4  cidade: "Buzios"
5  estado: "RJ"
6  telefone: "(21) 3333-3333"
```

string

string

Int32

string

string

Guía Documents

Luego, podemos cambiar fácilmente, en la pestaña Documents, el tipo de datos del campo de id seleccionando el tipo de datos correcto.

05

Guía Explain

Guía Explain

En la guía **Explain Plain** (Planes de Ejecución), podrán entender cómo se ejecutaron sus consultas. Y también si se puede mejorar este comportamiento.

En el ejemplo al lado, enumeramos los datos de la colección de contactos. Vean la imagen para comprenderlo mejor.

DBII.contato

Documents Aggregations Schema **Explain Plan** Indexes Validation

FILTER { field: 'value' } OPTIONS EXPLAIN RESET ↺ ...

VIEW DETAILS AS VISUAL TREE RAW JSON

Query Performance Summary

| | |
|------------------------|--------------------------------------|
| Documents Returned: 6 | Actual Query Execution Time (ms): 20 |
| Index Keys Examined: 0 | Sorted in Memory: no |
| Documents Examined: 6 | ⚠ No index available for this query. |

COLLSCAN
nReturned: 6

El Explain nos informa que la consulta devolvió 6 documentos, en 20 ms (milisegundos), y que no fue utilizado el índice.

06

Guía Indexes

Guía Indexes

La pestaña Indexes muestra los índices que se crearon en la colección seleccionada y cuáles se usaron.

Esta es información importante ya que puede sugerir cambios en su query o índice.

En este caso tenemos un índice llamado `_id_`, de tipo único, de 36,9 kb de tamaño, usado 23 veces.

DBII.contato

| Documents | Aggregations | Schema | Explain Plan | Indexes | Validation |
|--|--------------|------------------------|-----------------------------|------------|------------|
| CREATE INDEX | | | | | |
| Name and Definition | Type | Size | Usage | Properties | |
| <code>_id_</code> <code>_id...</code> | REGULAR ⓘ | 36.9 KB <div></div> | 23 since Wed Feb 02 2022 | UNIQUE ⓘ | |

**Nota: Una única colección no puede tener más de 64 índices.
No puede haber más de 32 campos en un índice compuesto.**

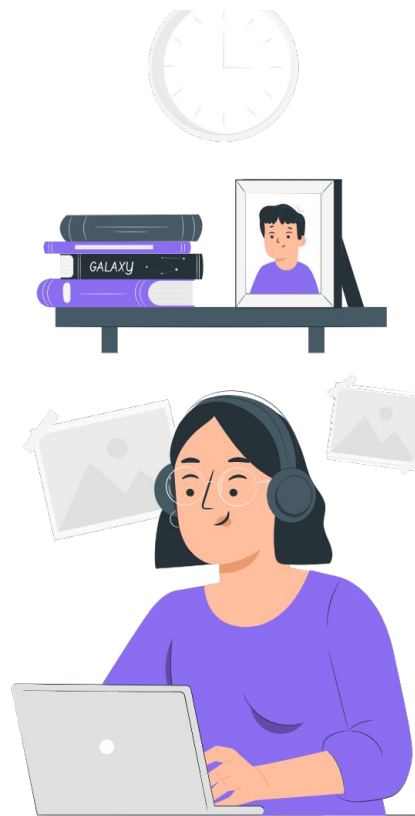
07

Guía Validation

Guía Validation

La guía **Validation** sirve para definir **reglas de validación** para sus **schemas**.

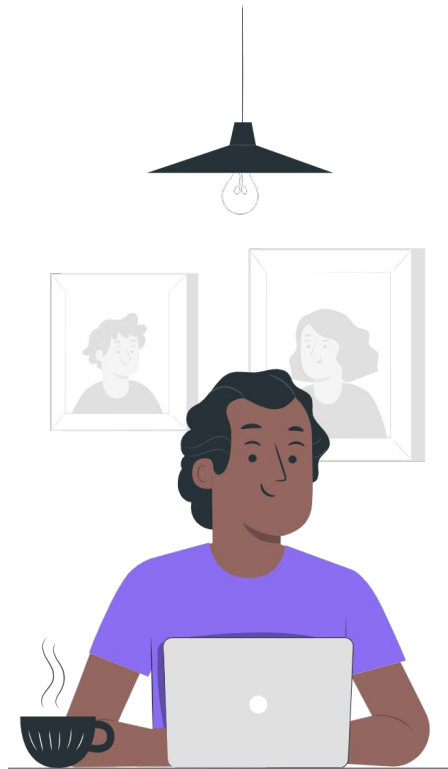
La validación del esquema garantiza que todos los documentos de una colección sigan un conjunto definido de reglas. Como, por ejemplo, ajustarse a una forma específica o solo permitir un rango específico de valores en los campos



Guía Validation

El editor de la validación admite la validación de **esquema JSON** y validación con expresiones de consulta.

A medida que edita, Compass se actualiza en tiempo real para mostrar un documento en su colección que pasa la validación y un documento que falla.



Guía Validation

Observen el ejemplo de validación al lado.

En este ejemplo, definimos como regla que el campo **_id** debe ser del **tipo double**.

Validation Action ⓘ

ERROR ▾

Validation Level ⓘ

STRICT ▾

```
1 {  
2   $jsonSchema: {  
3     required: [  
4       '_id'  
5     ],  
6     properties: {  
7       _id: {  
8         bsonType: 'double',  
9         description: 'Inclua um valor do tipo double'  
10      }  
11    }  
12  }  
13 }
```

Guía Validation

En el ejemplo de la pantalla anterior, definimos como regla que el campo **_id** es del **tipo double**. Vimos, en la guía Schema, que tenemos dos tipos de datos en el campo **_id**: **double** e **string**. Luego, nuestro documento de **_id = 1** fue validado porque sólo tenía **_id** del tipo double. Los demás no cumplen con la regla de validación.



Sample Document That Passed Validation

```
_id: 1
nome: "Ana Luisa"
idade: 22
cidade: "Campo Grande"
estado: "MS"
telefone: "(67) 3032-2233"
```



Sample Document That Failed Validation

```
_id: "002"
nome: "Luís Claudio"
idade: 31
cidade: "Buzios"
estado: "RJ"
telefone: "(21) 2019-3294"
```

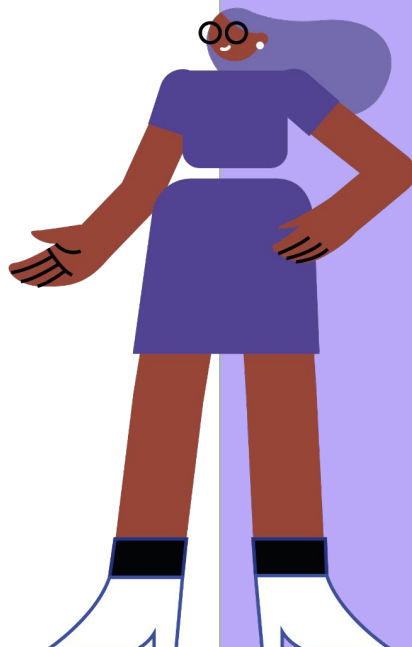
Conclusiones

MongoDB Compass es un gran aliado para el desarrollador, dadas todas las facilidades y posibilidades que esta herramienta pone a su disposición.

Vimos aquí que Compass permite crear una base de datos, colecciones, documentos e importar datos. Ya sea en formato Json o CSV.

También podemos agregar filtros y funciones de forma simplificada, en la pestaña Aggregate. Además, permite guardar la canalización, exportar a un lenguaje de programación o crear una vista para su uso posterior.

Compass también nos permite visualizar datos importantes como Schema, indexar e incluso crear reglas de validación para nuestra base de datos.



¡Muchas gracias!