

Laboratorio 05 - Introducción a MongoDB

Integrantes

- Abby Donis - 22440
- Mathew Cordero - 22982
- Josué Say - 220801

Ejercicio 1 - Preparación de ambiente

1.1 Crear la base de datos lab05 en MongoDB Atlas

- Ingrese a su cuenta de MongoDB Atlas.
- Desde la interfaz gráfica de MongoDB Atlas, cree una nueva base de datos llamada lab05.

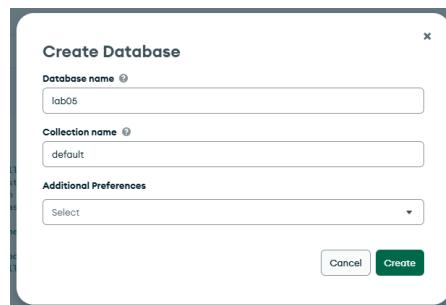


Figure 1: Creación de Base de Datos

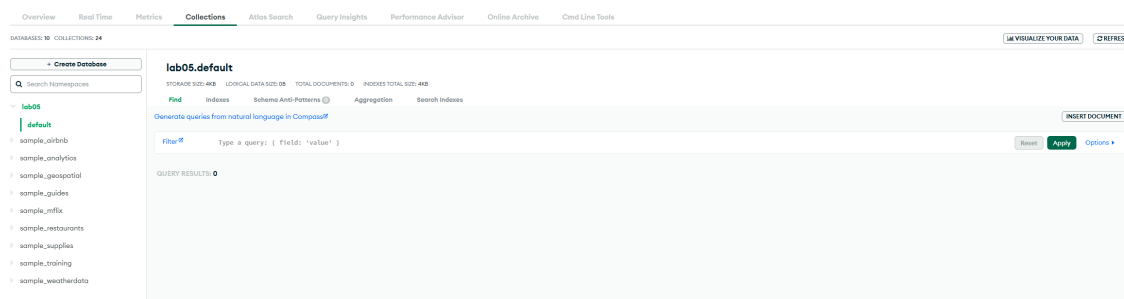


Figure 2: Confirmación de Creación de Base de Datos

1.2 Obtener la URI de conexión

- Obtenga la URI de conexión de su base de datos en MongoDB Atlas.

1.3 Crear las colecciones recetas y usuarios

- Desde su IDE NoSQL, cree las colecciones recetas y usuarios dentro de la base de datos lab05.

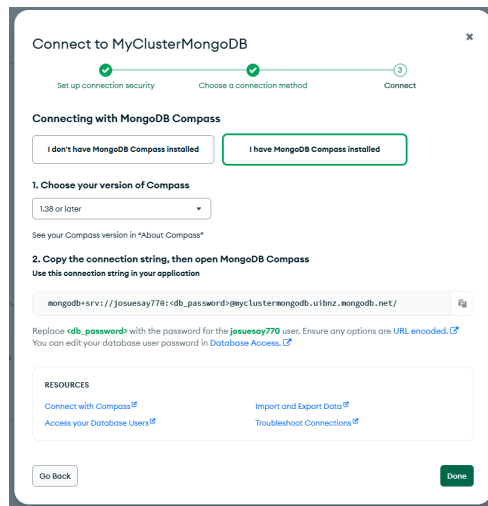


Figure 3: Obtención de URI para Conexión

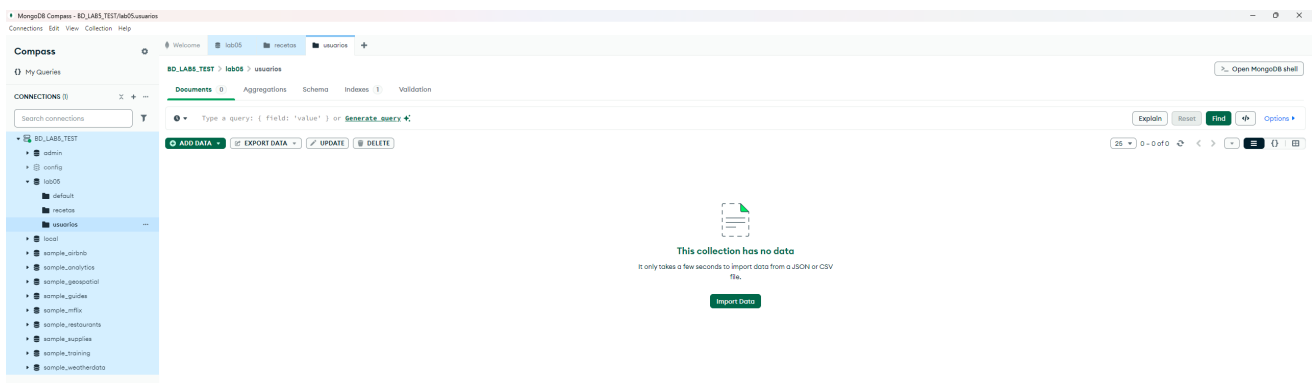
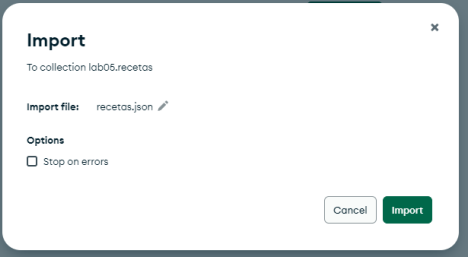
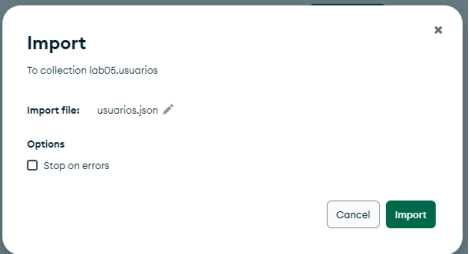


Figure 4: Creación de Colecciones

1.4 Importar los datos a las colecciones

- Desde su IDE, importe los datos correspondientes a las colecciones recetas y usuarios en su base de datos lab05.

Imagen	Descripción
	Importación de datos de la colección recetas.
	Importación de datos de la colección usuarios.

Ejercicio 2 - Exploración y manipulación de datos

2.1 Crear una consulta para obtener todas las recetas

- Realice una consulta en MongoDB para obtener todos los documentos de la colección recetas.

```
db["recetas"].find();
```

[illegible]

Figure 5: Ver Todas las Recetas

2.2 Crear una consulta para obtener todos los usuarios

- Realice una consulta en MongoDB para obtener todos los documentos de la colección usuarios.

```
db["usuarios"].find();
```

[illegible]

Figure 6: Ver Todas los Usuarios

2.3 Crear un nuevo documento en la colección recetas

- Con base en la estructura observada en la colección `recetas`, cree un nuevo documento que contenga los siguientes campos:

- Título, Descripción, Tiempo de Cocción, Etc

- Si se usa el documento JSON el documento ya existe por lo que se puede usar el siguiente comando para eliminarlo y volverlo a crear

```
db["recetas"].deleteOne({ _id: "6" });
```

- Luego creamos la nueva receta:

```
db["recetas"].insertOne({
  _id: "6",
  nombre: "Tacos de Carnitas",
  descripcion:
    "Tacos tradicionales mexicanos con carne de cerdo cocida lentamente hasta quedar
    ↪ dorada y jugosa.",
  "tiempo total minutos": 180,
  Porciones: 6,
  calorías: "500",
  dificultad: "Media",
})
```

```

vegetariana: false,
ingredientes: [
  { nombre: "carne de cerdo (pierna o lomo)", cantidad: "1 kg" },
  { nombre: "naranja", cantidad: "1 unidad (jugo)" },
  { nombre: "ajo", cantidad: "3 dientes" },
  { nombre: "hoja de laurel", cantidad: "1 unidad" },
  { nombre: "sal", cantidad: "1 cucharada" },
  { nombre: "pimienta negra", cantidad: "1 cucharadita" },
  { nombre: "comino", cantidad: "1 cucharadita" },
  { nombre: "agua", cantidad: "500 ml" },
  { nombre: "tortillas de maíz", cantidad: "12 unidades" },
  { nombre: "salsa verde o roja", cantidad: "al gusto" },
],
preparación: [
  "Corta la carne de cerdo en trozos medianos y colócala en una olla grande.",
  "Añade el jugo de naranja, los ajos enteros, la hoja de laurel, sal, pimienta y",
  ↪ comino.",
  "Vierte el agua hasta que la carne quede cubierta y cocina a fuego medio hasta que",
  ↪ el agua se evapore (aproximadamente 1.5 horas).",
  "Una vez que el agua se haya evaporado, deja que la carne se dore en su propia grasa",
  ↪ hasta que esté bien dorada y crujiente.",
  "Retira la hoja de laurel y desmenuza ligeramente la carne con un tenedor.",
  "Calienta las tortillas de maíz en un comal y arma los tacos con la carne de",
  ↪ carnitas.",
  "Decora con cebolla picada, cilantro y salsa al gusto. ¡Disfruta!",
],
ratings: [5, 5],
rating_promedio: 4.0,
});

```

```

// Receta de Carnitas
{
  id: 1,
  nombre: "Receta de Carnitas",
  descripcion:
    "Receta tradicional mexicana con carne de cerdo cocida lentamente hasta quedar dorada y jugosa.",
  tiempo_cooking: 180,
  dificultad: "Fácil",
  ingredientes: [
    { nombre: "Carne de cerdo (pierna o lomo)", cantidad: "1 kg" },
    { nombre: "Naranja", cantidad: "1 unidad (jugo)" },
    { nombre: "Ajo", cantidad: "3 dientes" },
    { nombre: "Hoja de laurel", cantidad: "1 unidad" },
    { nombre: "Sal", cantidad: "1 cucharada" },
    { nombre: "Pimienta negra", cantidad: "1 cucharadita" },
    { nombre: "Comino", cantidad: "1 cucharadita" },
    { nombre: "Agua", cantidad: "500 ml" },
    { nombre: "Tortillas de maíz", cantidad: "12 unidades" },
    { nombre: "Salsa verde o roja", cantidad: "al gusto" },
  ],
  preparacion: [
    "Corta la carne de cerdo en trozos medianos y colócala en una olla grande.",
    "Vierte el jugo de naranja, los ajos enteros, la hoja de laurel, sal, pimienta y comino.",
    "Vierte el agua hasta que la carne quede cubierta y cocina a fuego medio hasta que el agua se evapore (aproximadamente 1.5 horas).",
    "Una vez que el agua se haya evaporado, deja que la carne se dore en su propia grasa hasta que esté bien dorada y crujiente.",
    "Retira la hoja de laurel y desmenuza ligeramente la carne con un tenedor.",
    "Calienta las tortillas de maíz en un comal y arma los tacos con la carne de carnitas.",
    "Decora con cebolla picada, cilantro y salsa al gusto. ¡Disfruta!",
  ],
  ratings: [5, 5],
  rating_promedio: 4.0,
},
// Fin de la receta
}

```

Figure 7: Creación de Nueva Receta

2.4 Crear una consulta que busque la receta que acaba de crear

- Realice una consulta para buscar el documento de la receta que acaba de agregar.

```
db.recetas.find({ nombre: "Tacos de Carnitas" });
```



```
db.recetas.find({ nombre: "Tacos de Carnitas" });
{
  "_id": "1",
  "nombre": "Tacos de Carnitas",
  "descripcion": "Tacos tradicionales mexicanos con carne de cerdo cocida lentamente hasta quedar tierna y jugosa.",
  "tiempo total minutos": 180,
  "porciones": 4,
  "calorias": 1800,
  "dificultad": "Medio",
  "ingredientes": [
    {
      "nombre": "carne de cerdo (carnitas o lomo)",
      "cantidad": "1 kg",
      "orden": 1
    },
    {
      "nombre": "cebolla",
      "cantidad": "1 unidad (grande)",
      "orden": 2
    },
    {
      "nombre": "ajonjol",
      "cantidad": "3 dientes",
      "orden": 3
    },
    {
      "nombre": "hoja de laurel",
      "cantidad": "1 unidad",
      "orden": 4
    },
    {
      "nombre": "sal",
      "cantidad": "1 cucharada",
      "orden": 5
    },
    {
      "nombre": "pimentón negro",
      "cantidad": "1 cucharadita",
      "orden": 6
    },
    {
      "nombre": "comino",
      "cantidad": "1 cucharadita",
      "orden": 7
    },
    {
      "nombre": "agua",
      "cantidad": "100 ml",
      "orden": 8
    },
    {
      "nombre": "tortillas de maíz",
      "cantidad": "12 unidades",
      "orden": 9
    },
    {
      "nombre": "salsa verde o roja",
      "cantidad": "al gusto",
      "orden": 10
    }
  ]
}
```

Figure 8: Buscar Receta Creada

2.5 Crear una consulta en la que liste las recetas, mostrando únicamente el título y su tiempo de cocción

- Realice una consulta que liste solo los campos título y tiempo de cocción de todas las recetas.

```
db.recetas.find(
  {},
  { "Titulo Receta": "$nombre", "Tiempo Coccion": "$tiempo total minutos" }
);
```



```
db.recetas.find(
  {},
  { "Titulo Receta": "$nombre", "Tiempo Coccion": "$tiempo total minutos" }
);
{
  "Titulo Receta": "Tacos de Carnitas",
  "Tiempo Coccion": 180,
  "_id": "1"
},
{
  "Titulo Receta": "Tacos de Pollo",
  "Tiempo Coccion": 120,
  "_id": "2"
},
{
  "Titulo Receta": "Tacos de Pasa",
  "Tiempo Coccion": 120,
  "_id": "3"
},
{
  "Titulo Receta": "Tacos de Berenjena",
  "Tiempo Coccion": 120,
  "_id": "4"
},
{
  "Titulo Receta": "Tacos de Cerveza",
  "Tiempo Coccion": 120,
  "_id": "5"
},
{
  "Titulo Receta": "Tacos de Berenjena",
  "Tiempo Coccion": 120,
  "_id": "6"
},
{
  "Titulo Receta": "Tacos de Berenjena",
  "Tiempo Coccion": 120,
  "_id": "7"
},
{
  "Titulo Receta": "Tacos de Berenjena",
  "Tiempo Coccion": 120,
  "_id": "8"
},
{
  "Titulo Receta": "Tacos de Berenjena",
  "Tiempo Coccion": 120,
  "_id": "9"
},
{
  "Titulo Receta": "Tacos de Berenjena",
  "Tiempo Coccion": 120,
  "_id": "10"
},
{
  "Titulo Receta": "Tacos de Berenjena",
  "Tiempo Coccion": 120,
  "_id": "11"
},
{
  "Titulo Receta": "Tacos de Berenjena",
  "Tiempo Coccion": 120,
  "_id": "12"
},
{
  "Titulo Receta": "Tacos de Berenjena",
  "Tiempo Coccion": 120,
  "_id": "13"
},
{
  "Titulo Receta": "Tacos de Berenjena",
  "Tiempo Coccion": 120,
  "_id": "14"
},
{
  "Titulo Receta": "Tacos de Berenjena",
  "Tiempo Coccion": 120,
  "_id": "15"
},
{
  "Titulo Receta": "Tacos de Berenjena",
  "Tiempo Coccion": 120,
  "_id": "16"
},
{
  "Titulo Receta": "Tacos de Berenjena",
  "Tiempo Coccion": 120,
  "_id": "17"
},
{
  "Titulo Receta": "Tacos de Berenjena",
  "Tiempo Coccion": 120,
  "_id": "18"
},
{
  "Titulo Receta": "Tacos de Berenjena",
  "Tiempo Coccion": 120,
  "_id": "19"
},
{
  "Titulo Receta": "Tacos de Berenjena",
  "Tiempo Coccion": 120,
  "_id": "20"
}
```

Figure 9: Buscar Título y Tiempo de Cocción de Recetas

2.6 Crear una consulta en la que se listen las recetas ordenadas por mayor tiempo de cocción

- Realice una consulta que liste las recetas, ordenadas de mayor a menor según el tiempo de cocción.

```
db.recetas
    .find(
        {},
        { "Titulo Receta": "$nombre", "Tiempo Coccion": "$tiempo total minutos" }
    )
    .sort({ "tiempo total minutos": -1 });
```



Figure 10: Tiempos de Cocción Ordenadas

2.7 Investigar la instrucción `update()` para agregar un rating más a una receta y actualizar el rating promedio

- Investigue la instrucción `update()` en MongoDB y cree una consulta que agregue un nuevo `rating` a una receta, y actualice el `rating` promedio de la receta.

```
db.recetas.updateOne(
  { nombre: "Tacos de Carnitas" },
  {
    $push: { ratings: 5 },
    $set: {
      rating_promedio: 5,
    },
  }
);
```

- Comprobando el resultado:

```
db.recetas.find(
```

```
> db.recetas.find(
  { nombre: "Tacos de Carnitas" },
  { nombre: 1, rating_promedio: 1, _id: 0, ratings: 1 }
);
< {
  nombre: 'Tacos de Carnitas',
  ratings: [
    5,
    5,
    5
  ],
  rating_promedio: 5
}
```

Figure 11: Comprobación al Agregar Nuevos Campos

2.8 Crear una consulta en la que elimine un ingrediente de la lista de ingredientes de una receta en específico

- Realice una consulta para eliminar un ingrediente específico de la lista de ingredientes de una receta determinada.

```
db.recetas.updateOne(
  { nombre: "Tacos de Carnitas" },
  { $pull: { ingredientes: { nombre: "cilantro" } } }
);
```

- Comprobación de cambios.

```
db.recetas.find({ nombre: "Tacos de Carnitas" });
```

```

100  "id":
101    "type": "string",
102    "description": "ID de la actividad",
103    "required": true,
104    "examples": ["1", "2", "3"]
105  },
106  "name":
107    "type": "string",
108    "description": "Nombre de la actividad",
109    "required": true,
110    "examples": ["Caminar", "Correr", "Saltar"]
111  },
112  "description":
113    "type": "string",
114    "description": "Descripción detallada de la actividad",
115    "required": false,
116    "examples": ["Caminar rápido durante 1 hora"]
117  },
118  "start":
119    "type": "string",
120    "description": "Fecha y hora de inicio",
121    "required": true,
122    "examples": ["2023-10-27T08:00:00"]
123  },
124  "end":
125    "type": "string",
126    "description": "Fecha y hora de fin",
127    "required": true,
128    "examples": ["2023-10-27T09:00:00"]
129  },
130  "location":
131    "type": "string",
132    "description": "Ubicación de la actividad",
133    "required": false,
134    "examples": ["Parque Central"]
135  },
136  "status":
137    "type": "string",
138    "description": "Estado de la actividad",
139    "required": true,
140    "examples": ["Pendiente", "Completada", "Cancelada"]
141  },
142  "created_at":
143    "type": "string",
144    "description": "Fecha de creación",
145    "required": true,
146    "examples": ["2023-10-27T07:30:00"]
147  },
148  "updated_at":
149    "type": "string",
150    "description": "Fecha de actualización",
151    "required": true,
152    "examples": ["2023-10-27T07:30:00"]
153  },
154  "user_id":
155    "type": "string",
156    "description": "ID del usuario",
157    "required": true,
158    "examples": ["1", "2", "3"]
159  },
160  "category":
161    "type": "string",
162    "description": "Categoría de la actividad",
163    "required": false,
164    "examples": ["Deporte", "Arte", "Música"]
165  },
166  "duration":
167    "type": "integer",
168    "description": "Duración en minutos",
169    "required": true,
170    "examples": [15, 30, 45, 60]
171  },
172  "difficulty":
173    "type": "string",
174    "description": "Nivel de dificultad",
175    "required": false,
176    "examples": ["Fácil", "Medio", "Difícil"]
177  },
178  "tags":
179    "type": "array",
180    "description": "Etiquetas asociadas",
181    "required": false,
182    "examples": ["#Deporte", "#Salud", "#Bienestar"]
183  },
184  "notes":
185    "type": "string",
186    "description": "Notas adicionales",
187    "required": false,
188    "examples": ["Me gustó mucho"]
189  },
190  "rating":
191    "type": "integer",
192    "description": "Puntuación de calificación",
193    "required": false,
194    "examples": [1, 2, 3, 4, 5]
195  },
196  "image_url":
197    "type": "string",
198    "description": "URL de la imagen",
199    "required": false,
200    "examples": ["https://example.com/image.jpg"]
201  },
202  "shareable":
203    "type": "boolean",
204    "description": "¿Se puede compartir?",
205    "required": true,
206    "examples": [true, false]
207  },
208  "visibility":
209    "type": "string",
210    "description": "Visibilidad de la actividad",
211    "required": true,
212    "examples": ["Pública", "Privada"]
213  },
214  "calendar_sync":
215    "type": "boolean",
216    "description": "¿Sincronizar con calendario?",
217    "required": true,
218    "examples": [true, false]
219  },
220  "reminder":
221    "type": "string",
222    "description": "Recordatorio",
223    "required": false,
224    "examples": ["Recordatorio: ¡No olvides ir!", "Recordatorio: ¡No olvides ir a la hora indicada!"]
225  },
226  "attachments":
227    "type": "array",
228    "description": "Archivos adjuntos",
229    "required": false,
230    "examples": [{"id": 1, "name": "archivo1.pdf", "size": 1024}, {"id": 2, "name": "archivo2.jpg", "size": 2048}]
231  },
232  "metadata":
233    "type": "object",
234    "description": "Metadatos adicionales",
235    "required": false,
236    "examples": [{"source": "web", "device": "mobile", "os": "android"}]
237  },
238  "created_by":
239    "type": "string",
240    "description": "Usuario que creó la actividad",
241    "required": true,
242    "examples": ["usuario1", "usuario2"]
243  },
244  "updated_by":
245    "type": "string",
246    "description": "Usuario que actualizó la actividad",
247    "required": false,
248    "examples": ["usuario1", "usuario2"]
249  },
250  "deleted_by":
251    "type": "string",
252    "description": "Usuario que eliminó la actividad",
253    "required": false,
254    "examples": ["usuario1", "usuario2"]
255  },
256  "deleted_at":
257    "type": "string",
258    "description": "Fecha de eliminación",
259    "required": false,
260    "examples": ["2023-10-27T10:00:00"]
261  },
262  "soft_deleted":
263    "type": "boolean",
264    "description": "¿Eliminada de forma suave?",
265    "required": true,
266    "examples": [true, false]
267  },
268  "is_public":
269    "type": "boolean",
270    "description": "¿Es pública?",
271    "required": true,
272    "examples": [true, false]
273  },
274  "is_featured":
275    "type": "boolean",
276    "description": "¿Destacada?",
277    "required": true,
278    "examples": [true, false]
279  },
280  "is_archived":
281    "type": "boolean",
282    "description": "¿Archivada?",
283    "required": true,
284    "examples": [true, false]
285  },
286  "is_favorite":
287    "type": "boolean",
288    "description": "¿Favorita?",
289    "required": true,
290    "examples": [true, false]
291  },
292  "is_shared":
293    "type": "boolean",
294    "description": "¿Compartida?",
295    "required": true,
296    "examples": [true, false]
297  },
298  "is_completed":
299    "type": "boolean",
300    "description": "¿Completada?",
301    "required": true,
302    "examples": [true, false]
303  },
304  "is_cancelled":
305    "type": "boolean",
306    "description": "¿Cancelada?",
307    "required": true,
308    "examples": [true, false]
309  },
310  "is_pending":
311    "type": "boolean",
312    "description": "¿Pendiente?",
313    "required": true,
314    "examples": [true, false]
315  },
316  "is_active":
317    "type": "boolean",
318    "description": "¿Activa?",
319    "required": true,
320    "examples": [true, false]
321  },
322  "is_deleted":
323    "type": "boolean",
324    "description": "¿Eliminada?",
325    "required": true,
326    "examples": [true, false]
327  },
328  "is_hidden":
329    "type": "boolean",
330    "description": "¿Ocultada?",
331    "required": true,
332    "examples": [true, false]
333  },
334  "is_visible":
335    "type": "boolean",
336    "description": "¿Visible?",
337    "required": true,
338    "examples": [true, false]
339  },
340  "is_accessible":
341    "type": "boolean",
342    "description": "¿Accesible?",
343    "required": true,
344    "examples": [true, false]
345  },
346  "is_available":
347    "type": "boolean",
348    "description": "¿Disponible?",
349    "required": true,
350    "examples": [true, false]
351  },
352  "is_enabled":
353    "type": "boolean",
354    "description": "¿Habilitada?",
355    "required": true,
356    "examples": [true, false]
357  },
358  "is_disabled":
359    "type": "boolean",
360    "description": "¿Deshabilitada?",
361    "required": true,
362    "examples": [true, false]
363  },
364  "is_locked":
365    "type": "boolean",
366    "description": "¿Bloqueada?",
367    "required": true,
368    "examples": [true, false]
369  },
370  "is_unlocked":
371    "type": "boolean",
372    "description": "¿Desbloqueada?",
373    "required": true,
374    "examples": [true, false]
375  },
376  "is_readonly":
377    "type": "boolean",
378    "description": "¿Solo lectura?",
379    "required": true,
380    "examples": [true, false]
381  },
382  "is_writable":
383    "type": "boolean",
384    "description": "¿Escritura permitida?",
385    "required": true,
386    "examples": [true, false]
387  },
388  "is_executable":
389    "type": "boolean",
390    "description": "¿Ejecutable?",
391    "required": true,
392    "examples": [true, false]
393  },
394  "is_configurable":
395    "type": "boolean",
396    "description": "¿Configurable?",
397    "required": true,
398    "examples": [true, false]
399  },
400  "is_customizable":
401    "type": "boolean",
402    "description": "¿Personalizable?",
403    "required": true,
404    "examples": [true, false]
405  },
406  "is_mutable":
407    "type": "boolean",
408    "description": "¿Modificable?",
409    "required": true,
410    "examples": [true, false]
411  },
412  "is_immutable":
413    "type": "boolean",
414    "description": "¿Inmodificable?",
415    "required": true,
416    "examples": [true, false]
417  },
418  "is_stable":
419    "type": "boolean",
420    "description": "¿Estable?",
421    "required": true,
422    "examples": [true, false]
423  },
424  "is_unstable":
425    "type": "boolean",
426    "description": "¿Inestable?",
427    "required": true,
428    "examples": [true, false]
429  },
430  "is_secure":
431    "type": "boolean",
432    "description": "¿Segura?",
433    "required": true,
434    "examples": [true, false]
435  },
436  "is_insecure":
437    "type": "boolean",
438    "description": "¿Insegura?",
439    "required": true,
440    "examples": [true, false]
441  },
442  "is_safe":
443    "type": "boolean",
444    "description": "¿Segura de usar?",
445    "required": true,
446    "examples": [true, false]
447  },
448  "is_unsafe":
449    "type": "boolean",
450    "description": "¿Insegura de usar?",
451    "required": true,
452    "examples": [true, false]
453  },
454  "is_valid":
455    "type": "boolean",
456    "description": "¿Válida?",
457    "required": true,
458    "examples": [true, false]
459  },
460  "is_invalid":
461    "type": "boolean",
462    "description": "¿Inválida?",
463    "required": true,
464    "examples": [true, false]
465
```

Figure 12: Comprobación de Eliminación de Ingredientes

2.9 Investigar la opción `skip()` de la instrucción `find()` y crear una consulta en la que obtenga la tercera receta con mejor rating promedio

- Investigue la opción `skip()` de la instrucción `find()`. Luego, cree una consulta que obtenga la tercera receta con el mejor rating promedio.

db.recetas

```
.find({}, { Nombre: "$nombre", "Rating promedio": "$rating_promedio" })
.sort({ rating_promedio: -1 })
.skip(2)
.limit(1);
```

```
db.recetas
  .find({}, { Nombre: "$nombre", "Rating promedio": "$rating_promedio" })
  .sort({ rating_promedio: -1 })
  .skip(2)
  .limit(1);

< {
  _id: '7',
  Nombre: 'Espagueti a la Carbonara',
  'Rating promedio': 4.75
}
```

Figure 13: Obtener Mejor Rating

2.10 Crear una consulta que busque las recetas que tienen comentarios

- Realice una consulta que liste todas las recetas que tengan comentarios.

```
db.recetas.find({ comentarios: { $exists: true, $ne: [] } });
```

[illegible]

Figure 14: Recetas con Comentarios

2.11 Crear una consulta en la que liste las recetas que son de postres

- Dado que no tenemos el campo Tipo para obtener los postres debemos hacer un update de los campos:

```

db.recetas.updateOne(
  { nombre: "Tacos de Carnitas" },
  { $set: { tipo: "plato fuerte" } }
);

db.recetas.updateOne(
  { nombre: "Pizza Margarita" },
  { $set: { tipo: "plato fuerte" } }
);

db.recetas.updateOne(
  { nombre: "Sopa de Lentejas" },
  { $set: { tipo: "entrada" } }
);

db.recetas.updateOne(
  { nombre: "Ensalada César" },
  { $set: { tipo: "entrada" } }
);

db.recetas.updateOne(
  { nombre: "Espagueti a la Carbonara" },
  { $set: { tipo: "plato fuerte" } }
);

db.recetas.updateOne(
  { nombre: "Brownies de Chocolate" },
  { $set: { tipo: "postre" } }
);

db.recetas.updateOne(
  { nombre: "Pollo al Curry" },
  { $set: { tipo: "plato fuerte" } }
);

db.recetas.updateOne(
  { nombre: "Ensalada Griega" },
  { $set: { tipo: "entrada" } }
);

```

```
db.recetas.updateOne(
  { nombre: "Pasta Alfredo" },
  { $set: { tipo: "plato fuerte" } }
);
```

```
db.recetas.updateOne(
  { nombre: "Sopa de Tomate" },
  { $set: { tipo: "entrada" } }
);
```

- Comprobando los cambios:

```
db["recetas"].find({}, { nombre: 1, tipo: 1, _id: 0 });
```

- Realice una consulta que liste todas las recetas que sean de tipo postres.

```
db.recetas.find({ tipo: "postre" });
```

```
> db.recetas.find(
  { tipo: "postre" },
)
{
  _id: '5',
  nombre: 'Brownies de Chocolate',
  'tiempo total minutos': 45,
  Porciones: 8,
  calorías: '450',
  dificultad: 'Media',
  vegetariana: true,
  ingredientes: [
    {
      nombre: 'chocolate negro',
      cantidad: '200 g'
    },
    {
      nombre: 'mantequilla',
      cantidad: '100 g'
    },
    {
      nombre: 'azúcar',
      cantidad: '1 taza'
    },
    {
      nombre: 'harina',
      cantidad: '1 taza'
    },
    {
      nombre: 'huevos',
      cantidad: '3 unidades'
    },
    {
      nombre: 'polvo para hornear',
      cantidad: '1 cucharadita'
    },
    {
      nombre: 'sal',
      cantidad: '1 pizca'
    }
  ],
  'preparación': [
    'Derrite el chocolate con la mantequilla a baño maría.',
    'En un bol, bate los huevos con el azúcar hasta obtener una mezcla espumosa.',
    'Añade el chocolate derretido y mezcla bien.',
    'Incorpora la harina, el polvo para hornear y la sal.',
    'Vierte la mezcla en un molde engrasado y hornea a 180°C por 30 minutos.',
    'Deja enfriar, corta en cuadros y disfruta.'
  ],
  ratings: [],
  rating_promedio: 3,
  tipo: 'postre'
}
```

Figure 15: Obtener Postres

2.12 Crear 3 nuevos documentos de usuarios en una sola instrucción

- Con base en la estructura observada en la colección usuarios, cree 3 nuevos documentos de usuarios con los siguientes campos:

- Nombre
- Apellido
- Correo electrónico
- Contraseña

```
db.usuarios.insertOne({
  nombre: "Josue",
  apellido: "Say",
  correo: "say_josue@correo.com",
  contraseña: "123",
});
```

```
db.usuarios.insertOne({
  nombre: "Mathew",
  apellido: "Cordero",
  correo: "cordero_mathew@correo.com",
  contraseña: "456",
});
```

```
db.usuarios.insertOne({
  nombre: "Abby",
  apellido: "Donis",
  correo: "donis_abby@correo.com",
  contraseña: "789",
});
```

- Comprobando resultados.

```
db.usuarios.find({
  correo: {
    $in: [
      "say_josue@correo.com",
      "cordero_mathew@correo.com",
      "donis_abby@correo.com",
    ],
  },
});
```

```

> db.usuarios.find({
  correo: {
    $in: [
      "say_josue@correo.com",
      "cordero_mathew@correo.com",
      "donis_abby@correo.com",
    ],
  },
});
< {
  _id: ObjectId('67e129a318729d0b8c18b90c'),
  nombre: 'Josue',
  apellido: 'Say',
  correo: 'say_josue@correo.com',
  'contraseña': '123'
}
{
  _id: ObjectId('67e129a318729d0b8c18b90d'),
  nombre: 'Mathew',
  apellido: 'Cordero',
  correo: 'cordero_mathew@correo.com',
  'contraseña': '456'
}
{
  _id: ObjectId('67e129a318729d0b8c18b90e'),
  nombre: 'Abby',
  apellido: 'Donis',
  correo: 'donis_abby@correo.com',
  'contraseña': '789'
}

```

Figure 16: Comprobación de Creación de Nuevos Usuarios

2.13 Crear las consultas para agregar la receta favorita a cada uno de los usuarios creados anteriormente

- Cree una consulta para agregar la receta favorita de cada uno de los usuarios creados en el ejercicio anterior.

```

db.usuarios.updateOne(
  { correo: "say_josue@correo.com" },
  { $set: { receta_favorita: "Tacos de Carnitas" } }
);

```

```

db.usuarios.updateOne(
  { correo: "donis_abby@correo.com" },
  { $set: { receta_favorita: "Brownies de Chocolate" } }
);

```

```

db.usuarios.updateOne(
  { correo: "cordero_mathew@correo.com" },
  { $set: { receta_favorita: "Pasta Alfredo" } }
);

```

- Comprobando cambios.

```

db.usuarios.find(

```

```

{
  correo: {
    $in: [
      "say_josue@correo.com",
      "donis_abby@correo.com",
      "cordero_mathew@correo.com",
    ],
  },
},
{ nombre: 1, receta_favorita: 1, _id: 0 }
);

```

```

> db.usuarios.find(
  {
    correo: {
      $in: [
        "say_josue@correo.com",
        "donis_abby@correo.com",
        "cordero_mathew@correo.com",
      ],
    },
  },
  { nombre: 1, receta_favorita: 1, _id: 0 }
);
< {
  nombre: 'Josue',
  receta_favorita: 'Tacos de Carnitas'
}
{
  nombre: 'Mathew',
  receta_favorita: 'Pasta Alfredo'
}
{
  nombre: 'Abby',
  receta_favorita: 'Brownies de Chocolate'
}

```

Figure 17: Comprobando Nuevo Campo Agregado

2.14 Crear una consulta para consultar los distintos nombres de usuarios

- Realice una consulta que liste todos los nombres distintos de los usuarios. Se utilizó este **enlace** para ver el funcionamiento de distinct.

```
db.usuarios.distinct("nombre");
```

```

> db.usuarios.distinct("nombre");
< [
  'Abby',
  'Ana Gómez',
  'Carlos Rodríguez',
  'Josue',
  'Juan Pérez',
  'Laura Martínez',
  'Mathew',
  'Miguel Torres'
]

```

Figure 18: Nombres Distintos en Usuarios

2.15 Agregar un campo de actividad a los usuarios

- Agregue un campo llamado actividad a los usuarios para indicar si están activos o inactivos, utilizando un valor booleano.

```
db.usuarios.updateMany({}, { $set: { actividad: true } });
```

- Comprobación de cambios.

```
db.usuarios.find({}, { nombre: 1, actividad: 1, _id: 0 });
```

```
> db.usuarios.find({}, { nombre: 1, actividad: 1, _id: 0 });
< {
  nombre: 'Juan Pérez',
  actividad: true
}
{
  nombre: 'Ana Gómez',
  actividad: true
}
{
  nombre: 'Carlos Rodríguez',
  actividad: true
}
{
  nombre: 'Laura Martínez',
  actividad: true
}
{
  nombre: 'Miguel Torres',
  actividad: true
}
{
  nombre: 'Josue',
  actividad: true
}
{
  nombre: 'Mathew',
  actividad: true
}
{
  nombre: 'Abby',
  actividad: true
}
```

Figure 19: Comprobación de Nuevo Campo en Usuarios

2.16 Crear una consulta en la que inactive a 2 usuarios

- Realice una consulta para cambiar el valor de actividad a inactivo para 2 usuarios específicos.

```
db.usuarios.updateOne({ nombre: "Mathew" }, { $set: { actividad: false } });
```

```
db.usuarios.updateOne({ nombre: "Abby" }, { $set: { actividad: false } });
```

- Comprobación de cambios.

```
db.usuarios.find({}, { nombre: 1, actividad: 1, _id: 0 });
```

```

> db.usuarios.find({}, { nombre: 1, actividad: 1, _id: 0 });
< {
  nombre: 'Juan Pérez',
  actividad: true
}
{
  nombre: 'Ana Gómez',
  actividad: true
}
{
  nombre: 'Carlos Rodríguez',
  actividad: true
}
{
  nombre: 'Laura Martínez',
  actividad: true
}
{
  nombre: 'Miguel Torres',
  actividad: true
}
{
  nombre: 'Josue',
  actividad: true
}
{
  nombre: 'Mathew',
  actividad: false
}
{
  nombre: 'Abby',
  actividad: false
}

```

Figure 20: Comprobación de Cambio de Actividad

2.17 Crear una consulta en la que cambie la unidad de medida de todas las recetas que tienen lb a kg

- Realice una consulta para actualizar todas las recetas que tienen la unidad de medida lb y cambiarla a kg. Ya que no hay recetas con cantidad lb se hará el procedimiento inverso, se convertirá los kg a lb.
- Buscar cuales son dichos documentos.

```
db.recetas.find({ "ingredientes.cantidad": /kg/ });
```

```

db.recetas.find({ "ingredientes.cantidad": /kg/ })
{
  "_id": "61",
  "nombre": "Paseo de Carillitas",
  "descripcion": "Paseo tradicional mexicano con carne de cerdo cocida lentamente hasta quedar dorada y jugosa.",
  "tiempo total minutos": 180,
  "porciones": 8,
  "calorias": 500,
  "dificultad": "Medio",
  "vegetariano": false,
  "ingredientes": [
    {
      "nombre": "carne de cerdo (pierna o lomo)",
      "cantidad": "1 kg"
    },
    {
      "nombre": "marraja",
      "cantidad": "1 unidad (jugosa)"
    },
    {
      "nombre": "ají",
      "cantidad": "15 dientes"
    },
    {
      "nombre": "hoja de laurel",
      "cantidad": "1 unidad"
    },
    {
      "nombre": "sal",
      "cantidad": "1 cucharada"
    },
    {
      "nombre": "pistacho negro",
      "cantidad": "1 cucharadita"
    },
    {
      "nombre": "coconut",
      "cantidad": "1 cucharadita"
    },
    {
      "nombre": "agua",
      "cantidad": "500 ml"
    },
    {
      "nombre": "tortillas de maíz",
      "cantidad": "12 unidades"
    },
    {
      "nombre": "salsa verde o roja",
      "cantidad": "al gusto"
    }
  ]
}

```

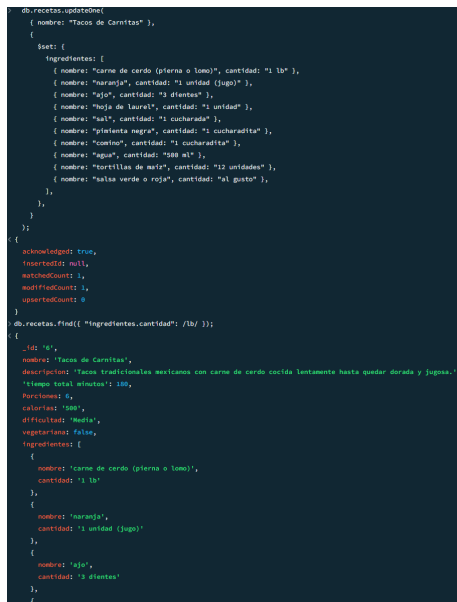
Figure 21: Ver Recetas en Kg

- Aplicar el cambio.


```

db.recetas.updateOne(
  { nombre: "Tacos de Carnitas" },
  {
    $set: {
      ingredientes: [
        { nombre: "carne de cerdo (pierna o lomo)", cantidad: "1 lb" },
        { nombre: "naranja", cantidad: "1 unidad (jugo)" },
        { nombre: "ajo", cantidad: "3 dientes" },
        { nombre: "hoja de laurel", cantidad: "1 unidad" },
        { nombre: "sal", cantidad: "1 cucharada" },
        { nombre: "pimienta negra", cantidad: "1 cucharadita" },
        { nombre: "comino", cantidad: "1 cucharadita" },
        { nombre: "agua", cantidad: "500 ml" },
        { nombre: "tortillas de maíz", cantidad: "12 unidades" },
        { nombre: "salsa verde o roja", cantidad: "al gusto" },
      ],
    },
  }
);

```



```

db.recetas.updateOne(
  { nombre: "Tacos de Carnitas" },
  {
    $set: {
      ingredientes: [
        { nombre: "carne de cerdo (pierna o lomo)", cantidad: "1 lb" },
        { nombre: "naranja", cantidad: "1 unidad (jugo)" },
        { nombre: "ajo", cantidad: "3 dientes" },
        { nombre: "hoja de laurel", cantidad: "1 unidad" },
        { nombre: "sal", cantidad: "1 cucharada" },
        { nombre: "pimienta negra", cantidad: "1 cucharadita" },
        { nombre: "comino", cantidad: "1 cucharadita" },
        { nombre: "agua", cantidad: "500 ml" },
        { nombre: "tortillas de maíz", cantidad: "12 unidades" },
        { nombre: "salsa verde o roja", cantidad: "al gusto" },
      ],
    },
  }
);
{
  "acknowledged": true,
  "insertedId": null,
  "matchedCount": 1,
  "modifiedCount": 1,
  "upsertedCount": 0
}
db.recetas.find( { "ingredientes.cantidad": /lb/ } );
{
  "_id": "55",
  "nombre": "Tacos de Carnitas",
  "descripcion": "Tacos tradicionales mexicanos con carne de cerdo cocida lentamente hasta quedar dorada y jugosa.",
  "tiempo total minutos": 180,
  "porciones": 4,
  "calorias": 1800,
  "dificultad": "Medio",
  "vegetariano": false,
  "ingredientes": [
    {
      "nombre": "carne de cerdo (pierna o lomo)",
      "cantidad": "1 lb"
    },
    {
      "nombre": "naranja",
      "cantidad": "1 unidad (jugo)"
    },
    {
      "nombre": "ajo",
      "cantidad": "3 dientes"
    }
  ]
}

```

Figure 22: Comprobación de Cambios

2.18 Crear una consulta en la que elimine a los usuarios inactivos

- Realice una consulta para eliminar todos los usuarios que tengan el campo actividad establecido en inactivo.

```
db.usuarios.deleteMany({ actividad: false });
```

- Comprobación de cambios.

```
db.usuarios.find({}, { nombre: 1, actividad: 1, _id: 0 });
```

```
> db.usuarios.find({}, { nombre: 1, actividad: 1, _id: 0 });
< {
  nombre: 'Juan Pérez',
  actividad: true
}
{
  nombre: 'Ana Gómez',
  actividad: true
}
{
  nombre: 'Carlos Rodríguez',
  actividad: true
}
{
  nombre: 'Laura Martínez',
  actividad: true
}
{
  nombre: 'Miguel Torres',
  actividad: true
}
{
  nombre: 'Josue',
  actividad: true
}
```

Figure 23: Comprobación de Eliminación de Usuarios Inactivos

Nota: Enlace consultado para actualizaciones en grupo.