

Nombre:

Apellidos:

Fecha:

26/07/21

- Todas las preguntas tienen el mismo valor.
- Las preguntas incorrectas no restan puntuación.
- En general las preguntas poseen una única respuesta, si alguna de ellas tiene respuestas múltiples se indica en el enunciado.

1. ¿Cómo se organizan los datos dentro de MongoDB?

- a. MongoDB mantiene una única base de datos que contiene múltiples colecciones, donde los documentos son almacenados.
- b. MongoDB es capaz de manejar múltiples bases de datos con múltiples colecciones que almacenan documentos en formato BSON.
- c. MongoDB no utiliza colecciones.
- d. MongoDB almacena los datos en tablas.

2. La shell de MongoDB utiliza documentos JSON de forma intensiva para cumplir diferentes tipos de tareas como las operaciones CRUD, manejo de configuraciones de administración, etc. ¿cuál de las siguientes afirmaciones sobre JSON es correcta?

- a. El formato JSON es también el utilizado para almacenar datos en el disco de forma persistente.
- b. El formato JSON soporta el tipo date.
- c. MongoDB obliga a definir un esquema de datos específico para cada colección y que debe cumplir todos los documentos de esa colección.
- d. El formato JSON no soporta nativamente el tipo decimal de BSON.

3. Elige la sentencia correcta:

- a. Si un documento vacío es insertado en MongoDB, entonces el documento es almacenado como un documento vacío.
- b. Si un documento vacío es insertado en MongoDB, entonces el documento es almacenado con un único campo: el `_id` auto-generado.
- c. La inserción de un documento vacío en MongoDB fallará siempre.
- d. Ninguna de las anteriores es correcta.

4. Considera las siguientes sentencias ejecutadas desde la consola de MongoDB y elige la opción correcta:

```
use gimnasio;  
db.clientes.drop();  
db.clientes.insert({nombre: "Juan", apellidos: "Pérez"});  
db.clientes.insert({nombre: "Juan", apellidos: "Pérez"});  
db.clientes.insert({nombre: "Carlos"});
```

- a. La segunda operación de inserción fallará porque MongoDB no permite tener dos documentos en la misma colección con los mismos campos y valores.
- b. Ninguna de las operaciones fallará, pero la segunda operación sobrescribirá el primer documento y solamente dos documentos persistirán.
- c. Todos los documentos serán añadidos y, por tanto, la colección contará con tres documentos.
- d. Todas las inserciones fallarán porque es necesario especificar un campo `_id` para todos los documentos a insertar.

5. Una inserción fallará si:

- a. Ya existe un documento en la colección con el mismo valor para el campo `_id`.
- b. Se intenta asignar un array al campo `_id`.
- c. El documento insertado posee una sintaxis inválida.
- d. Todas las respuestas anteriores son correctas.

6. Basándose en la siguiente inserción mostrada, ¿cuál de las siguientes afirmaciones son correctas?

```
db.vehiculos.insert( [  
  { _id: "6748FHJ", brand: "Open", model: "Corsa" },  
  { _id: "6749FHJ", brand: "Open", model: "Corsa" },  
  { _id: 4389, companyName: "MongoDB" },  
  { _id: [1,2], CPU: "Intel", brand: "Dell" },  
  { _id: "6750FHJ", brand: "Open", model: "Corsa" }  
, { ordered: true } )
```

- a. Los documentos son ordenados por `_id` antes de ser insertados.
 - b. La inserción del tercer y cuarto documento fallará porque sus valores para el campo `_id` poseen tipos distintos.
 - c. Después de realizar la operación, tres documentos habrán sido insertados.
 - d. Después de realizar la operación, cinco documentos habrán sido insertados.
7. ¿Qué comandos están disponibles en MongoDB para eliminar documentos?
- a. MongoDB prohíbe eliminar documentos porque el paradigma NoSQL no lo permite.
 - b. Los documentos solamente pueden ser eliminados uno a uno con `deleteOneByOne`.
 - c. Los documentos pueden ser eliminados mediante `deleteOne` y `deleteMany`.
 - d. El comando `deleteOne` es utilizado para eliminar documentos de una instancia standalone en MongoDB, mientras que `deleteMany` se emplea únicamente en clústers replica set.
8. ¿Qué sucede cuando se borra totalmente una colección con `drop()`, en lugar de realizar la operación de eliminación de todos documentos?
- a. Son eliminados todos los datos, índices y metadatos asociados a la colección.
 - b. Son eliminados los índices, pero no los datos.
 - c. Son eliminados los datos, pero no los índices.
 - d. Son operaciones similares.

9. Dada la siguiente consulta, ¿cuál es la respuesta esperada?

```
db.movies.find( { "year" : 1989, "title" : "Batman" } )
```

- a. Devuelve todos los documentos de la colección movies que poseen el campo year igual a 1989 y el campo title igual a Batman.
- b. Devuelve todos los documentos de la colección movies que poseen el campo year igual a 1989 o el campo title igual a Batman.
- c. Devuelve todos los documentos de todas las colecciones que poseen el campo year igual a 1989 o el campo title igual a Batman.
- d. Ninguna de las anteriores es correcta.

10. ¿Cuál de las siguientes afirmaciones es correcta en relación a los documentos embebidos?

- a. No es posible filtrar por campos que se encuentran dentro de documentos embebidos.
- b. Puede utilizarse la notación del punto (.) para acceder a los campos de los documentos embebidos.
- c. Los documentos embebidos no son soportados por MongoDB.
- d. Debe utilizarse el operador \$within_embedded_document para filtrar dentro de documentos embebidos.

11. Los resultados de las consultas pueden retornar el documento completo o parte del mismo mediante proyecciones. Considerando las siguientes instrucciones ejecutadas en la consola de MongoDB en la colección `movies` vacía, ¿cuál de las siguientes afirmaciones es correcta?

```
db.movies.insert(
{
  "title" : "Forrest Gump",
  "category" : [ "drama", "romance" ],
  "imdb_rating" : 8.8,
  "filming_locations" : [
    { "city" : "Savannah", "state" : "GA", "country" : "USA" },
    { "city" : "Monument Valley", "state" : "UT", "country" : "USA" },
    { "city" : "Los Angeles", "state" : "CA", "country" : "USA" }
  ],
  "box_office" : {
    "gross" : 557,
    "opening_weekend" : 24,
    "budget" : 55
  }
})
```

```
db.movies.find( { "title" : "Forrest Gump" },
                { "title" : 1, "imdb_rating" : 1 } )
```

- a. `find` no permite operaciones de proyección.
- b. El siguiente documento es devuelto:

```
{
  "_id" : ObjectId("5515942d31117f52a5122353"),
  "title" : "Forrest Gump",
  "imdb_rating" : 8.8
}
```

- c. El siguiente documento es devuelto:

```
{
  "title" : "Forrest Gump",
  "imdb_rating" : 8.8
}
```

- d. El siguiente documento es devuelto:

```
{
  "category" : [ "drama", "romance" ],
}
```

```
"filming_locations" : [
  { "city" : "Savannah", "state" : "GA", "country" : "USA" },
  { "city" : "Monument Valley", "state" : "UT", "country" : "USA" },
  { "city" : "Los Angeles", "state" : "CA", "country" : "USA" }
],
"box_office" : {
  "gross" : 557,
  "opening_weekend" : 24,
  "budget" : 55
}
}
```

12. A partir de las siguientes instrucciones ejecutadas en la consola de MongoDB, ¿cuál de las afirmaciones es correcta?

```
db.sensordata.insertOne(
{
  "_id" : "building1_floor1_room1_sensor1",
  "manufacturer" : "Siemens",
  ...
  "last_10_measures" : [ 10.5, 12, 14, 17, 18, 25, 18, 15, 12, 10],
  ...
})
```

```
db.sensordata.findOne( { "last_10_measures" : 25 } )
```

- MongoDB itera el array `last_10_measures` y comprueba cada elemento contra el valor del criterio especificado en el filtro. Si hay coincidencia, retorna el elemento encontrado.
- MongoDB itera el array `last_10_measures` y comprueba cada elemento contra el valor del criterio especificado en el filtro. Si hay coincidencia, retorna el documento completo.
- MongoDB no permite este tipo de consultas.
- La consulta no es correcta porque debe utilizarse el operador `$text`.

13. ¿Cuántos documentos serán actualizados después de ejecutar las siguientes instrucciones de MongoDB?

```
db.movies.insert( [
  {
    "title" : "Batman",
    "category" : [ "action", "adventure" ],
    "imdb_rating" : 7.6,
    "budget" : 35
  },
  {
    "title" : "Godzilla",
    "category" : [ "action", "adventure", "sci-fi" ],
    "imdb_rating" : 6.6
  },
  {
    "title" : "Home Alone",
    "category" : [ "family", "comedy" ],
    "imdb_rating" : 7.4
  }
] )

db.movies.updateMany( { "category": "action", },
                      { $set: { "category.$" : "action-adventure" } } )
```

- a. Dos.
 - b. Todos.
 - c. Ninguno.
 - d. Solamente el último.
14. ¿Cuántos tipos de índice son soportados por MongoDB?
- a. MongoDB no soporta índices.
 - b. Solamente el índice del campo `_id`, que es un índice de tipo único.
 - c. Existen muchos tipos de índices soportados por MongoDB: simples, compuestos, múltiples, únicos, parciales, de texto, etc.
 - d. Solamente tres: de campo único, de campo compuesto y simples.

15. ¿Qué impacto tienen los índices sobre el rendimiento?

- a. Los índices mejoran el rendimiento de las lecturas que son cubiertas, pero las inserciones serán más lentas siempre.
- b. Los índices mejoran el rendimiento de las lecturas que son cubiertas, pero las inserciones serán más lentas cuando los índices tengan que ser actualizados.
- c. Los índices siempre mejorarán las operaciones CRUD de cualquier tipo.
- d. Los índices no tienen nunca implicaciones sobre el rendimiento de lectura/escritura.

16. Basándose en las siguientes instrucciones ejecutadas en la consola de MongoDB, ¿cuál de las afirmaciones es correcta?

```
db.testcol.drop()
db.testcol.createIndex( { x : 1, y : 1 } )
db.testcol.insertOne( { _id : 1, x : 1, y : 1 } )
db.testcol.insertOne( { _id : 2, x : [ 1, 2 ], y : 1 } )
db.testcol.insertOne( { _id : 3, x : 1, y : [ 1, 2 ] } )
db.testcol.insertOne( { _id : 4, x : [ 1, 2 ], y : [ 1, 2 ] } )
```

- a. Todos los documentos serán insertados correctamente.
- b. La creación del índice fallará.
- c. Serán insertados dos documentos.
- d. Serán insertados tres documentos.

17. ¿Qué mecanismo existe en MongoDB para eliminar documentos después de haber transcurrido un determinado periodo de tiempo?

- a. Los índices TTL eliminan documentos automáticamente basándose en umbrales de expiración o fechas.
- b. Los índices de campo único y compuesto poseen una opción para establecer el tiempo de vida de los documentos.
- c. En los índices se puede establecer un umbral de operación lenta para eliminar documentos pesados.
- d. No existe ningún mecanismo de ese tipo en MongoDB.

18. ¿Cuál de las siguientes respuestas es afirmativa acerca de la salida proporcionada por el método `explain`? (Varias respuestas afirmativas)

```
...
"executionStats" : {
  "executionSuccess" : true,
  "nReturned" : 8,
  "executionTimeMillis" : 107,
  "totalKeysExamined" : 0,
  "totalDocsExamined" : 51428,
  "executionStages" : {
    "stage" : "COLLSCAN",
    "filter" : {
      "user.followers_count" : {
        "$eq" : 1000
      }
    }
  },
}
```

- a. No es una consulta óptima porque la fase empleada es COLLSCAN.
 - b. No es una consulta óptima, pero es muy rápida porque ronda los 100ms de tiempo.
 - c. Se recorren 51428 documentos y son devueltos 8.
 - d. Se recorren 8 documentos y son retornados 51428.
19. Elige una posible salida devuelta tras ejecutar la siguiente operación de agregación.

```
db.tweets.aggregate( [
  { "$match" : { "user.time_zone" : "Brasilia",
                 "user.statuses_count" : { "$gte" : 100 } } },
  { "$project" : { "followers" : "$user.followers_count",
                  "tweets" : "$user.statuses_count",
                  "screen_name" : "$user.screen_name" } },
  { "$sort" : { "followers" : -1 } },
  { "$limit" : 1 } ] )
```

- a. { `_id` : ObjectId('52fd2490bac3fa1975477702'),
 `followers` : 2597,
 `screen_name`: 'marbles',
 `tweets` : 12334
 }
- b. { `followers` : 2597,
 `screen_name`: 'marbles',
 `tweets` : 12334
 }

- c.

```
{ screen_name: 'marbles',  
  tweets      : 12334,  
  followers   : 2597  
}
```
- d.

```
{ _id      : ObjectId('52fd2490bac3fa1975477702'),  
  followers : 2597,  
  screen_name: 'marbles',  
  tweets    : 5  
}
```

20. Dado los siguientes documentos insertados y la operación de agregación asociada, elige la salida devuelta.

```
db.sales.insertMany( [  
  { "_id" : 1, "item" : "abc", "price" : 10,  
    "quantity" : 2, "date" : ISODate("2014-03-01T08:00:00Z") },  
  { "_id" : 2, "item" : "jkl", "price" : 20,  
    "quantity" : 1, "date" : ISODate("2014-03-01T09:00:00Z") },  
  { "_id" : 3, "item" : "xyz", "price" : 5,  
    "quantity" : 10, "date" : ISODate("2014-03-15T09:00:00Z") },  
  { "_id" : 4, "item" : "xyz", "price" : 5,  
    "quantity" : 20, "date" : ISODate("2014-04-04T11:21:39.736Z") },  
  { "_id" : 5, "item" : "abc", "price" : 10,  
    "quantity" : 10, "date" : ISODate("2014-04-04T21:23:13.331Z") } ] )
```

```
db.sales.aggregate( [  
  { $group : {  
    _id : null,  
    totalPrice: { $sum: { $multiply: [ "$price", "$quantity" ] } },  
    averageQuantity: { $avg: "$quantity" },  
    count: { $sum: 1 }  
  } }  
] )
```

- a.

```
{ "_id" : null, "totalPrice" : 250, "averageQuantity" : 6.6, "count" : 6 }
```
- b.

```
{ "_id" : null, "totalPrice" : 190, "averageQuantity" : 8.9, "count" : 15  
}
```
- c.

```
{ "_id" : null, "totalPrice" : 290, "averageQuantity" : 8.6, "count" : 5 }
```
- d.

```
{ "_id" : null, "totalPrice" : 90, "averageQuantity" : 8, "count" : 1 }
```

21. ¿Cuál de las siguientes afirmaciones es una buena práctica en el diseño del modelo de datos en MongoDB?

- a. El esquema de datos es independiente de los patrones de acceso de la aplicación según la filosofía de MongoDB.
- b. Solamente es necesario considerar relaciones 1-N y N-N.
- c. Es conveniente siempre evitar los documentos embebidos.
- d. Ninguna de las anteriores es correcta.

22. ¿Cuál de las siguientes afirmaciones sobre los documentos embebidos es correcta?

- a. Embeber documentos es siempre una mala idea.
- b. Los documentos embebidos vuelven más complejo el modelo de datos con respecto a disponer de dos colecciones.
- c. Embeber documentos es una buena idea cuando la relación es 1-N y N son muchísimos documentos.
- d. El rendimiento de lectura puede mejorar si se utilizan documentos embebidos porque toda la información se encuentra en un único documento.

23. ¿Cuál de las siguientes afirmaciones sobre la denormalización es correcta?

- a. La desnormalización no posee ninguna ventaja y debería ser evitada siempre.
- b. La idea de la denormalización es agregar datos redundantes donde se estime oportuno para facilitar y mejorar el rendimiento de algunas operaciones de lectura.
- c. Si un campo de un documento se actualiza con frecuencia, entonces es buen candidato para ser denormalizarlo.
- d. La denormalización no existe. Solamente existe el concepto de normalización.

24. ¿Cuál de las siguientes afirmaciones sobre la replicación es correcta?

- a. Los árbitros almacenan datos.
- b. Un miembro oculto no almacena datos.
- c. Los árbitros no pueden convertirse en miembros primarios.
- d. Los miembros retrasados mantienen los datos del primario actualizados en tiempo real.

25. ¿Qué característica fundamental ofrece la replicación en MongoDB?:

- a. Alta disponibilidad
- b. Operaciones CRUD más rápidas.
- c. Escalabilidad vertical.
- d. Escalabilidad horizontal.

26. ¿Qué tipo de replicación utiliza MongoDB?

- a. Replicación binaria.
- b. Replicación basada en árboles binarios.
- c. Replicación yuxtapuesta.
- d. Ninguna de las anteriores.

27. ¿Qué sucede en la siguiente reconfiguración de un replica set?

```
cfg = rs.conf();  
cfg.members[3].votes = 0;  
cfg.members[3].priority = 0;  
rs.reconfig(cfg);
```

- a. Se configura una prioridad de 0 al tercer miembro del replica set.
- b. El miembro reconfigurado se convertirá en miembro oculto.
- c. El miembro reconfigurado no podrá convertirse en primario.
- d. Ninguna de las anteriores.

28. Dada la siguiente configuración de un replica set, elige la opción correcta.

```
MongoDB Enterprise drplan:PRIMARY> rs.conf()  
{ "_id" : "drplan",  
  "version" : 4,  
  "protocolVersion" : NumberLong(1),  
  "members" : [ { "_id" : 0,  
                  "host" : "node1:27017",  
                  "arbiterOnly" : false,  
                  "buildIndexes" : true,  
                  "hidden" : false,  
                  "priority" : 10,  
                  "tags" : { },  
                  "slaveDelay" : NumberLong(0),  
                  "votes" : 1 },  
                { "_id" : 1,  
                  "host" : "node1:27018",  
                  "arbiterOnly" : false,
```

```
"buildIndexes" : true,
"hidden" : false,
"priority" : 5,
"tags" : { },
"slaveDelay" : NumberLong(0),
"votes" : 1 },
{ "_id" : 2,
"host" : "node2:27019",
"arbiterOnly" : false,
"buildIndexes" : true,
"hidden" : false,
"priority" : 1,
"tags" : { },
"slaveDelay" : NumberLong(0),
"votes" : 1 } ],
"settings" : { "chainingAllowed" : true,
"heartbeatIntervalMillis" : 2000,
"heartbeatTimeoutSecs" : 10,
"electionTimeoutMillis" : 10000,
"getLastErrorModes" : { },
"getLastErrorDefaults" : { w" : 1, "wtimeout" : 0 },
"replicaSetId" : ObjectId("57fb70585ddfaced502de56c8") }
}
```

- a. El node1:27017 será siempre el miembro primario y si se pierde conectividad con él, ninguno del resto de miembros puede convertirse en primario porque poseen una prioridad menor.
- b. Hay un miembro que toma el rol de árbitro.
- c. Si dos miembros caen, todavía se mantendría la mayoría y existiría la posibilidad de escribir en el miembro primario.
- d. Ninguna de las anteriores es correcta.

29. ¿Cómo son replicadas las operaciones de escritura en los miembros de un replica set?

- a. Manteniendo un seguimiento de los archivos de datos a nivel de bit.
- b. Utilizando la herramienta rsync de los sistemas Unix.
- c. Las operaciones son almacenadas en chunks y distribuidas por el balanceador de MongoDB.
- d. Almacenando las operaciones en una colección capped denominada oplog, que es mantenida por el miembro primario y replicada entre los miembros secundarios.



30. ¿Cuál es el tamaño por defecto de un chunk en un clúster sharding?
- a. 64MB
 - b. 32MB
 - c. 16MB
 - d. 128MB
31. ¿Cuál de los siguientes criterios es recomendable que tenga una shard key?
- a. Inserciones monotónicas o predecibles.
 - b. Alta frecuencia de escritura con mucha repetición de valores.
 - c. Alta cardinalidad.
 - d. Es indiferente.
32. ¿Dónde se almacenan las colecciones no fragmentadas en un clúster sharding?
- a. En el mongos.
 - b. En un shard aleatorio.
 - c. En el shard primario.
 - d. En el shard primario, pero solamente en el miembro primario del replica set asociado.
33. ¿Qué afirmación es correcta sobre la autenticación en MongoDB?
- a. MongoDB no provee ningún mecanismo de autenticación.
 - b. MongoDB ofrece un mecanismo simple de autenticación basado en usuario/contraseña.
 - c. MongoDB ofrece en su última versión los siguientes mecanismos de autenticación: SCRAM, certificados X509, LDAP y Kerberos.
 - d. MongoDB solamente ofrece mecanismos de autenticación en la versión Enterprise.
34. ¿Qué afirmación es correcta respecto a los roles?
- a. Un rol define un tipo personal de dato al que es accedido por el programador backend.
 - b. Los roles son un conjunto de privilegios que los usuarios necesitan poseer para poder manejar y acceder a algún recurso (base de datos, colección o clúster) de MongoDB.
 - c. Un rol es un conjunto de usuarios que comparten las mismas credenciales para acceder a las aplicaciones de MongoDB.
 - d. Todas las opciones anteriores son correctas.