

## INTRODUCCION

Para empeorar el panorama, Internet está lleno de amenazas contra la integridad de sus datos. Los virus y los troyanos no solo roban sus datos. En algunos casos, los borran. También debe considerar el ransomware, un fenómeno que está en crecimiento. En este caso, un hacker introduce un virus en su equipo, el cual cifra sus datos. Luego, le pedirán que pague cientos de dólares para que descifren sus datos. Si generó una copia de seguridad de sus datos, no debería afectarle tanto estas amenazas. Tan solo deberá limpiar el disco duro y usar su copia de seguridad más reciente.

No importa si se debe a un error mecánico, a un desastre natural o un crimen con malicia, sus datos se perdieron. Pero eso no significa que no pueda recuperarlos. Puede hacer copia de seguridad de ellos.

En el siguiente documento se creará un plan de copia de seguridad y un respaldo para una base de datos NoSQL, en este caso se trabajará con la base de datos no relaciona MongoDB. Siendo esta una de las bases de datos mas utilizadas en la actualidad por muchas organizaciones.

# PLAN DE COPIA DE SEGURIDAD

## OBJETIVOS

¿Que esperamos lograr con la implementación del respaldo de una BD en mongodb?

El resultado esperado en la implementación de respaldos en esta base de datos es el poder guardar una base de datos para previamente restaurarla cada que sea necesaria, manteniendo así su integridad ante cualquier percance realizado, evitando pérdidas ante diversas vulnerabilidades y creando un ambiente seguro dentro del servidor.

¿Qué logramos obtener?

Crear copias de seguridad de una base de datos no relacional (NoSQL)

## PERSONAS INVOLUCRADAS

- Diana Karen Ortiz Chávez – Administradora de la base de datos NoSQL

## PLANEACION

¿Qué?

Se realizará una copia de seguridad de una base de datos en mongodb, para entenderlo mejor, un respaldo de base de datos se refiere a las diversas estrategias y procedimientos involucrados en la protección de la base de datos contra la pérdida de información, y la reconstrucción después de cualquier tipo de pérdida. La copia de seguridad implementada en este plan será completa.

¿Para qué?

**Proteger los datos.**

**Administrar las bases de datos**

**Controlar la información**

¿Por qué?

Es necesario mantener copias de seguridad de diversas bases de datos porque así mantenemos una mejor integridad de nuestras bases de datos, es un requisito muy necesario para la mayoría de las organizaciones.

¿Cómo?

Implementaremos el respaldo y la restauración de la base de datos con el uso de comandos:

```
mongodump /host:localhost /port:27017 /db:nombre
```

**Mongodump:** Mongodump es la aplicación por defecto para realizar copias de seguridad del contenido de una instancia mongodb, el formato utilizado es el bson que presenta de forma binaria las estructuras de datos y sus mapas. Los backups se pueden ejecutar en modo consistente o inconsistente

**/host:localhost:** Dentro del host (justo después de los dos puntos) se colocará el localhost SOLO si nuestro equipo tiene ese nombre. Localhost es el nombre comúnmente utilizado para designar el equipo que estamos utilizando. Hace referencia a la computadora o dispositivo que estamos usando en un momento concreto.

**/port:27017:** Aquí colocaremos el puerto definido dentro de mongodb, el puerto por defecto es el 27017.

**/db:nombre :** Se colocará la base de datos seleccionada, pondremos su nombre DESPUES del db y los dos puntos

Todo este punto se explicará mejor en la evidencia solicitada.

¿Cuándo?

La copia de seguridad se hará cada cierto tiempo.

En este caso será:

Cada 7 días en una primera instancia de la creación y uso de la base de datos (empleada actualmente)

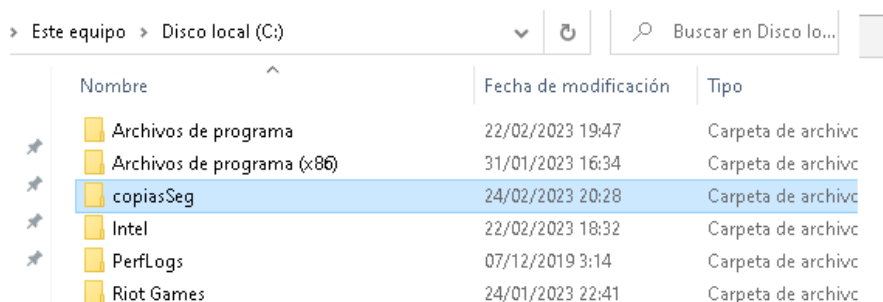
Cada 3 días, si es que la base de datos almacena grandes números de registros al día

.

¿Dónde?

Los respaldos se encontrarán dentro de un servidor local (PC).

Se almacenará en la siguiente carpeta

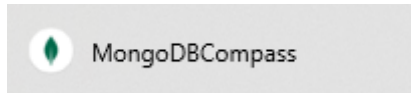


Este equipo > Disco local (C:)			
Buscar en Disco lo...			
	Nombre	Fecha de modificación	Tipo
	Archivos de programa	22/02/2023 19:47	Carpeta de archiv
	Archivos de programa (x86)	31/01/2023 16:34	Carpeta de archiv
	copiasSeg	24/02/2023 20:28	Carpeta de archiv
	Intel	22/02/2023 18:32	Carpeta de archiv
	PerfLogs	07/12/2019 3:14	Carpeta de archiv
	Riot Games	24/01/2023 22:41	Carpeta de archiv

## PROCEDIMIENTO DE COPIA DE SEGURIDAD

Para comenzar con la realización de una copia de seguridad, debemos tener considerado lo siguiente.

- Contar con el gestor de base de datos mongodb



- Tener instalada la herramienta mongoshell

```
mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\vortiz>mongosh
Current Mongosh Log ID: 63f9a3d3c18d396a679156cb
Connecting to:  mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000&appName=mongosh+1
.7.1
Using MongoDB:      6.0.4
Using Mongosh:       1.7.1

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

-----
The server generated these startup warnings when booting
2023-02-22T19:06:46.364-06:00: Access control is not enabled for the database. Read and write access to data and conf
iguration is unrestricted
-----
-----
Enable MongoDB's free cloud-based monitoring service, which will then receive and display
metrics about your deployment (disk utilization, CPU, operation statistics, etc).

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

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

- Tener instaladas las herramientas de mongodb

### TOOLS

## MongoDB Command Line Database Tools Download

The MongoDB Database Tools are a collection of command-line utilities for working with a MongoDB deployment. These tools release independently from the MongoDB Server schedule enabling you to receive more frequent updates and leverage new features as soon as they are available. See the [MongoDB Database Tools](#) documentation for more information.

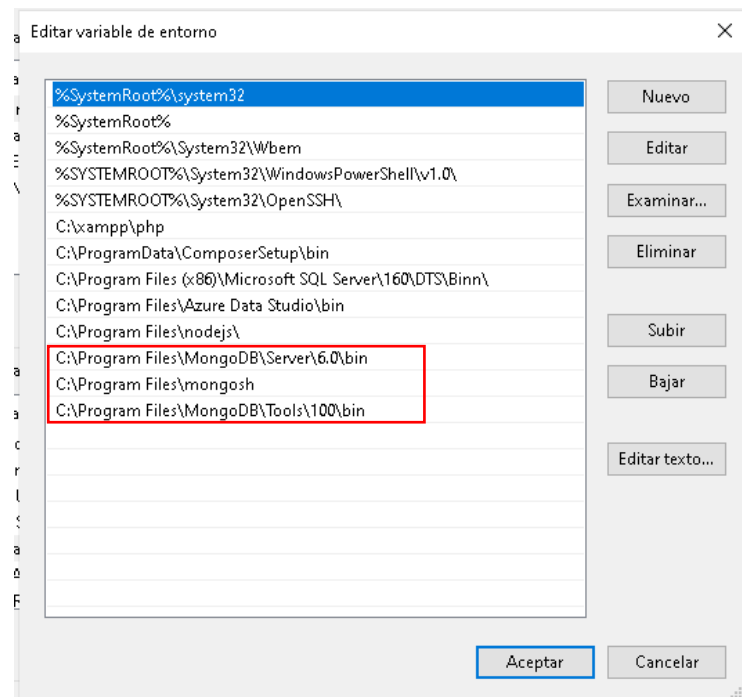
Version

100.6.1



Platform

Además, es bueno tener en cuenta que debemos tener nuestras variables de entorno actualizadas con cada herramienta utilizada para mongodb.



Ahora, después de esto ya podremos configurar, crear y realizar respaldos de nuestras bases de datos en mongodb

1. Creamos nuestra base de datos a trabajar (podemos crearla o utilizar cualquier base de datos ya previamente creada)
2. Para crear una base de datos en mongo solo es necesario colocar el comando 'use' seguido del nombre de nuestra base de datos a utilizar.

```
test> use trabajoSeg
switched to db trabajoSeg
```

3. Tomemos en cuenta que para ejecutar mongoshell debemos tener el servidor de mongo arriba
4. Esto se hace con el comando mongod

```
Microsoft Windows [Versión 10.0.19045.2604]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\ortiz>mongod

{"t":{"sdate":"2023-02-24T20:11:21.561-06:00"},"s":"I", "c":"NETWORK", "id":4915701, "ctx":"-", "msg":"Initialized wire specification", "attr":{"spec":{"incomingExternalClient":{"minWireVersion":0,"maxWireVersion":17},"incomingInternalClient":{"minWireVersion":0,"maxWireVersion":17},"outgoing":{"minWireVersion":6,"maxWireVersion":17},"isInternalClient":true}}}}
{"t":{"sdate":"2023-02-24T20:11:21.566-06:00"},"s":"I", "c":"CONTROL", "id":23285, "ctx":"thread1", "msg":"Automatically disabling TLS 1.0, to force-enable TLS 1.0 specify --sslDisabledProtocols 'none'"}
{"t":{"sdate":"2023-02-24T20:11:21.567-06:00"},"s":"I", "c":"NETWORK", "id":4648602, "ctx":"thread1", "msg":"Implicit TCP FastOpen in use."}
{"t":{"sdate":"2023-02-24T20:11:21.571-06:00"},"s":"I", "c":"REPL", "id":5123008, "ctx":"thread1", "msg":"Successfully registered PrimaryOnlyService", "attr":{"service":"TenantMigrationDonorService", "namespace":"config.tenantMigrationDonors"}}
{"t":{"sdate":"2023-02-24T20:11:21.572-06:00"},"s":"I", "c":"REPL", "id":5123008, "ctx":"thread1", "msg":"Successfully registered PrimaryOnlyService", "attr":{"service":"TenantMigrationRecipientService", "namespace":"config.tenantMigrationRecipients"}}
{"t":{"sdate":"2023-02-24T20:11:21.572-06:00"},"s":"I", "c":"REPL", "id":5123008, "ctx":"thread1", "msg":"Successfully registered PrimaryOnlyService", "attr":{"service":"ShardSplitDonorService", "namespace":"config.tenantSplitDonors"}}
{"t":{"sdate":"2023-02-24T20:11:21.572-06:00"},"s":"I", "c":"CONTROL", "id":5945603, "ctx":"thread1", "msg":"Multi threading initialized"}
{"t":{"sdate":"2023-02-24T20:11:21.625-06:00"},"s":"I", "c":"CONTROL", "id":4615611, "ctx":"initandlisten", "msg":"MongoDB starting", "attr":{"pid":13188, "port":27017, "dbPath":"c:/data/db/", "architecture":"64-bit", "host":"DESKTOP-UJEU17U"}}
{"t":{"sdate":"2023-02-24T20:11:21.631-06:00"},"s":"I", "c":"CONTROL", "id":23398, "ctx":"initandlisten", "msg":"Target operating system minimum version", "attr":{"targetMinOS":"Windows 7/Windows Server 2008 R2"}}
{"t":{"sdate":"2023-02-24T20:11:21.656-06:00"},"s":"I", "c":"CONTROL", "id":23403, "ctx":"initandlisten", "msg":"Build Info", "attr":{"buildInfo":{"version":"6.0.4", "gitVersion":"44ff59461c1353638a71e710f385a566bcd2f547", "modules":[], "allocator":"tcmalloc", "environment":{"distmod":"windows", "distarch":"x86_64", "target_arch":"x86_64"}}}}
```

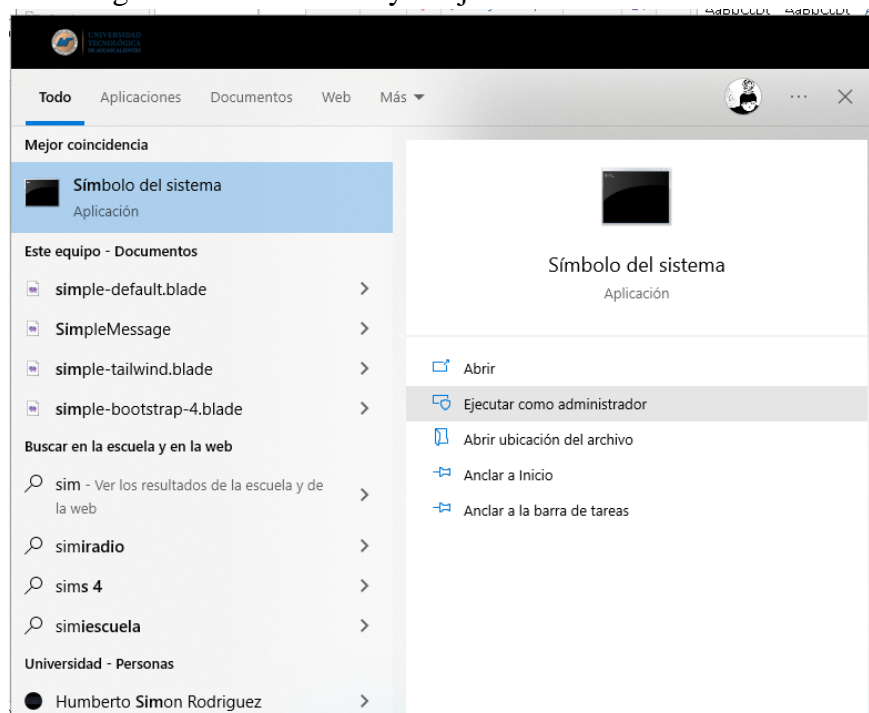
5. Utilizaremos la base de datos trabajoSeg

```
trabajoSeg> show dbs
admin          40.00 KiB
config         108.00 KiB
local          72.00 KiB
prueba1        72.00 KiB
trabajoSeg     72.00 KiB
trabajoSeg>
```

6. Para poder utilizar esta base de datos o documento que es como se les llama dentro de mongo, debemos crearles algunas colecciones.
7. Crearemos la colección user e insertaremos varios registros para respaldarlos luego.

```
trabajoSeg> db.user.insert({name: "Luis", age: 15})
DeprecationWarning: Collection.insert() is deprecated. Use insertOne, insertMany, or bulkWrite.
{ acknowledged: true,
  insertedIds: { '0': ObjectId("63f9a9136b2348dbbdcac6ff") } }
trabajoSeg> db.user.insert({name: "Andrea", age: 19})
{ acknowledged: true,
  insertedIds: { '0': ObjectId("63f9a9226b2348dbbdcac700") } }
trabajoSeg> db.user.insert({name: "Valentina", age: 17})
{ acknowledged: true,
  insertedIds: { '0': ObjectId("63f9a92f6b2348dbbdcac701") } }
trabajoSeg> db.user.insert({name: "Aaron", age: 22})
{ acknowledged: true,
  insertedIds: { '0': ObjectId("63f9a93c6b2348dbbdcac702") } }
trabajoSeg> db.user.insert({name: "Miguel", age: 25})
{ acknowledged: true,
  insertedIds: { '0': ObjectId("63f9a9476b2348dbbdcac703") } }
```

8. Listo, nuestra base de datos(documento) ya esta creada y tiene la colección user. Podemos proceder con la copia de seguridad.
9. Para respaldar nuestra base de datos haremos uso del comando mongodump
- 9.1 Nos dirigimos a nuestro cmd y lo ejecutamos como administrador.



9.2 Dentro de este nos movemos entre carpetas gracias al comando cd solo para dirigirnos a copiasSeg

```
C:\WINDOWS\system32>cd C:\copiasSeg  
C:\copiasSeg>
```

9.3 Colocamos el comando siguiente para crear la copia de seguridad

```
C:\copiasSeg>mongodump /host:localhost /port:27017 /db:trabajoSeg  
2023-02-25T00:42:52.766-0600   writing trabajoSeg.user to dump\trabajoSeg\user.bson  
2023-02-25T00:42:52.831-0600   done dumping trabajoSeg.user (5 documents)  
C:\copiasSeg>
```

9.4 Y listo, nuestra base de datos a sido respaldada dentro de la carpeta que elegimos previamente

10. Supongamos que la base de datos sufrió un percance y por accidente se ha estropeado, bueno, siempre es posible restaurar la base de datos SOLO si a esta se le ha realizado una copia de seguridad.

11. Simularemos lo siguiente

12. La base de datos prueba1 ha sido eliminada.

Antes

```
test> show dbs  
admin          40.00 KiB  
config         108.00 KiB  
local          72.00 KiB  
prueba1        72.00 KiB  
trabajoSeg     72.00 KiB
```

Después

```
prueba1> show dbs  
admin          40.00 KiB  
config         108.00 KiB  
local          72.00 KiB  
trabajoSeg     72.00 KiB
```

13. Entonces, para poder restaurar esa base de datos es necesario utilizar el comando **mongorestore**

14. Nos dirigimos a nuestro cmd, lo ejecutamos como administrador y gracias al cd nos movemos a nuestra carpeta copiasSeg

Administrador: Símbolo del sistema

```
C:\copiasSeg> _
```

## 15. Colocamos el comando

```
Administrador: Símbolo del sistema

C:\copiasSeg>mongorestore /host:localhost /port:27017 /db:prueba1
2023-02-25T01:01:55.359-0600 error connecting to host: could not connect to server: server selection error: server selection timeout, current topology: { Type: Single, Servers: [{ Addr: localhost:27017, Type: Unknown, Last error: connection() error occurred during connection handshake: dial tcp: lookup localhost: no such host }, ] }

C:\copiasSeg>mongorestore /host:localhost /port:27017 /db:prueba1
2023-02-25T01:03:47.958-0600 The --db and --collection flags are deprecated for this use-case; please use --nsInclude instead, i.e. with --nsInclude=${DATABASE}.${COLLECTION}
2023-02-25T01:03:47.972-0600 using default 'dump' directory
2023-02-25T01:03:47.972-0600 building a list of collections to restore from dump dir
2023-02-25T01:03:47.973-0600 don't know what to do with subdirectory "dump\prueba1", skipping...
2023-02-25T01:03:47.973-0600 don't know what to do with subdirectory "dump\trabajoSeg", skipping...
2023-02-25T01:03:47.973-0600 0 document(s) restored successfully. 0 document(s) failed to restore.

C:\copiasSeg>mongorestore
2023-02-25T01:06:08.275-0600 using default 'dump' directory
2023-02-25T01:06:08.369-0600 preparing collections to restore from
2023-02-25T01:06:08.384-0600 reading metadata for prueba1.user from dump\prueba1\user.metadata.json
2023-02-25T01:06:08.406-0600 reading metadata for trabajoSeg.user from dump\trabajoSeg\user.metadata.json
2023-02-25T01:06:08.437-0600 restoring to existing collection trabajoSeg.user without dropping
2023-02-25T01:06:08.477-0600 restoring trabajoSeg.user from dump\trabajoSeg\user.bson
2023-02-25T01:06:09.960-0600 restoring prueba1.user from dump\prueba1\user.bson
2023-02-25T01:06:09.999-0600 finished restoring prueba1.user (2 documents, 0 failures)
2023-02-25T01:06:09.142-0600 continuing through error: E11000 duplicate key error collection: trabajoSeg.user index: _id_dup key: { _id: ObjectId('63f9a9136b2348dbbdcac6ff') }
2023-02-25T01:06:09.143-0600 continuing through error: E11000 duplicate key error collection: trabajoSeg.user index: _id_dup key: { _id: ObjectId('63f9a9226b2348dbbdcac700') }
2023-02-25T01:06:09.143-0600 continuing through error: E11000 duplicate key error collection: trabajoSeg.user index: _id_dup key: { _id: ObjectId('63f9a92f6b2348dbbdcac701') }
2023-02-25T01:06:09.144-0600 continuing through error: E11000 duplicate key error collection: trabajoSeg.user index: _id_dup key: { _id: ObjectId('63f9a93c6b2348dbbdcac702') }
2023-02-25T01:06:09.145-0600 continuing through error: E11000 duplicate key error collection: trabajoSeg.user index: _id_dup key: { _id: ObjectId('63f9a9476b2348dbbdcac703') }
2023-02-25T01:06:09.146-0600 finished restoring trabajoSeg.user (0 documents, 5 failures)
2023-02-25T01:06:09.146-0600 no indexes to restore for collection prueba1.user
2023-02-25T01:06:09.147-0600 no indexes to restore for collection trabajoSeg.user
2023-02-25T01:06:09.148-0600 2 document(s) restored successfully. 5 document(s) failed to restore.

C:\copiasSeg>
```

16. Y listo, nuestra base de datos prueba1 ha sido restaurada y podemos seguir trabajando con ella desde la ultima instancia en la que la dejamos.

```
test> show dbs
admin          40.00 KiB
config         108.00 KiB
local          72.00 KiB
prueba1        8.00 KiB
trabajoSeg     72.00 KiB
test>

prueba1> db.user.find()
[
  {
    _id: ObjectId("63f801ee021c029bef885a5a"),
    name: 'Ada Lovelace',
    age: 205
  },
  { _id: ObjectId("63f802e2021c029bef885a5b"), name: 'Luis', age: 15 }
]
prueba1>
```



# COMPROBACIÓN DE LA COPIA DE SEGURIDAD Y DE LA RESTAURACION

Copia de seguridad creada.

<< Disco local (C:) > copiasSeg > dump			Buscar en dump
Nombre	Fecha de modificación	Tipo	
prueba1	24/02/2023 20:28	Carpeta de archivo	
trabajoSeg	25/02/2023 0:42	Carpeta de archivo	

Contenido

```
user.metadata.json - Visual Studio Code
C:\> copiasSeg > dump > trabajoSeg > user.metadata.json
1  "indexes":[{"v":{"numberInt":"2"},"key":{"_id":{"numberInt":"1"},"name":"_id"},"uid":"7000720c5f054517bed65f77657df375","collectionN...
```

Restauración realizada

Antes

```
prueba1> show dbs
admin          40.00 KiB
config         108.00 KiB
local          72.00 KiB
trabajoSeg     72.00 KiB
```

Después

```
test> show dbs
admin          40.00 KiB
config         108.00 KiB
local          72.00 KiB
prueba1       72.00 KiB
trabajoSeg     72.00 KiB
```

## CONCLUSIONES

No estamos seguros de todos los peligros a los que estamos expuestos en cuanto a seguridad de la información, cualquier descuido puede ocasionar perdidas masivas de datos, es por eso, que implementar copias de seguridad en nuestros proyectos y/o bases de datos es de gran importancia en nuestra actualidad (si estas están automatizadas seria mil veces mejor) ya que protege al usuario, protege a la organización y protege a toda esa cadena de personas y entidades relacionadas.

## REFERENCIAS BIBLIOGRAFICAS

(APA)

*La importancia de hacer copia de seguridad de los datos.* (s. f.).

<https://mx.norton.com/blog/how-to/the-importance-of-data-back-up>

*mongorestore — MongoDB Database Tools.* (s. f.).

<https://www.mongodb.com/docs/database-tools/mongorestore/>

Alexander Calderon. (2020, 23 mayo). *mongodb mongodump crear backup.* YouTube.

<https://www.youtube.com/watch?v=Dcc8JjD7Tos>

*Installing the Database Tools on Windows — MongoDB Database Tools.* (s. f.).

<https://www.mongodb.com/docs/database-tools/installation/installation-windows/>

*mongodump — MongoDB Database Tools.* (s. f.).

<https://www.mongodb.com/docs/database-tools/mongodump/>

MongoDB. (s. f.). *Cree Una Base De Datos En.*

<https://www.mongodb.com/es/basics/create-database>