



Universidad de Guadalajara

Alumno: Sánchez Gómez Edgardo Enrique

Código: 218401479

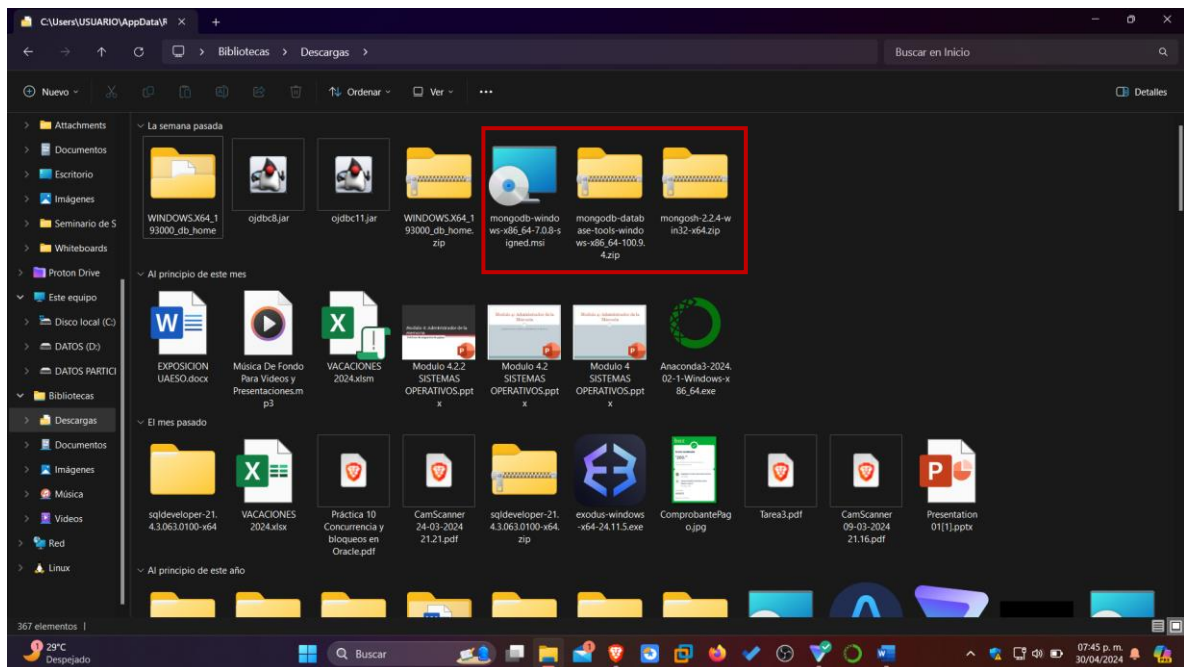
Asignatura: Administración de Bases de Datos

Práctica 15: Instalación y creación de usuarios en MongoDB

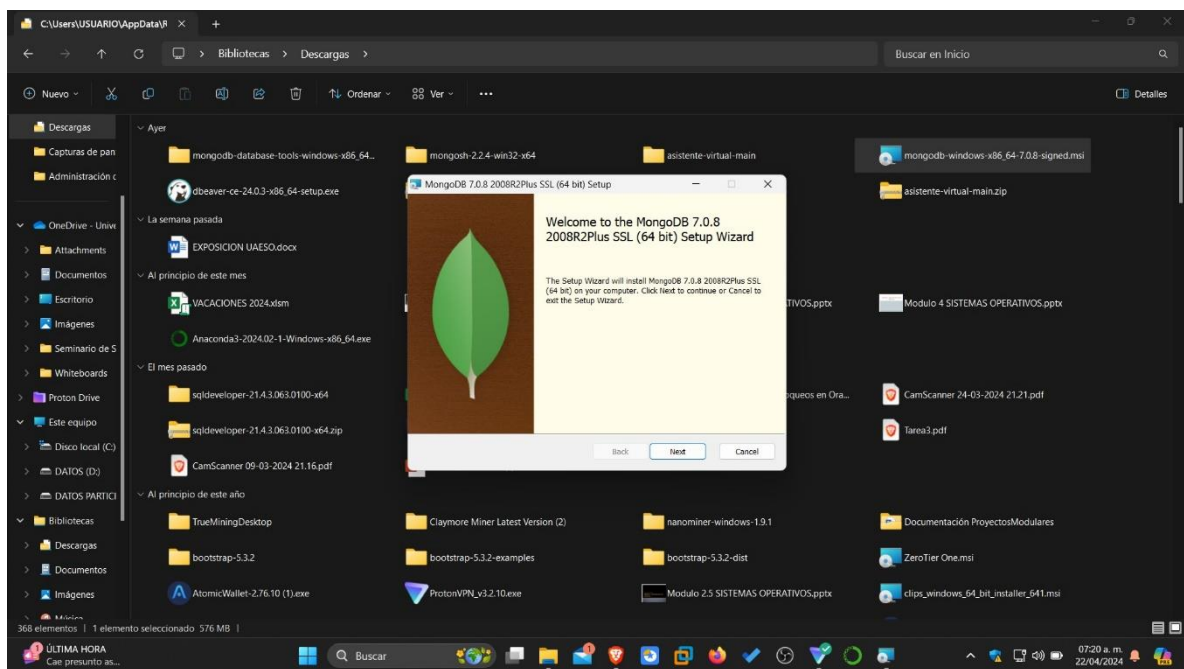
Fecha de entrega: 05/Mayo/2024

a) Instalación de MongoDB

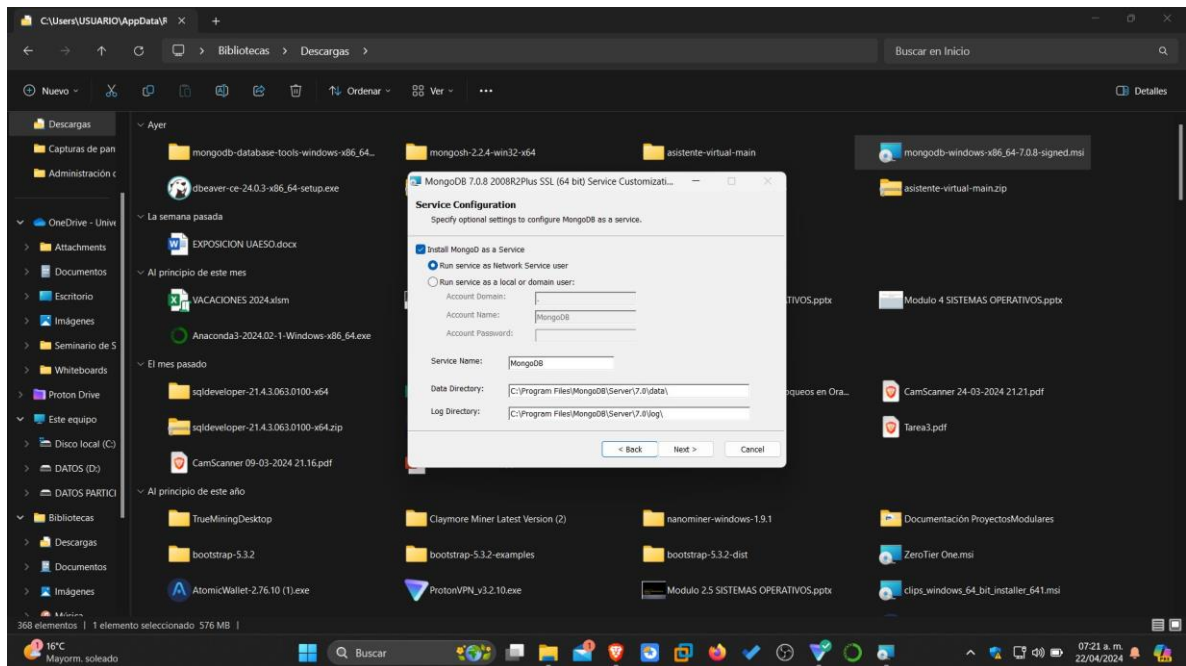
1. Descargamos MongoDB Community Edition, MongoDB Shell, y MongoDB Database Tools.



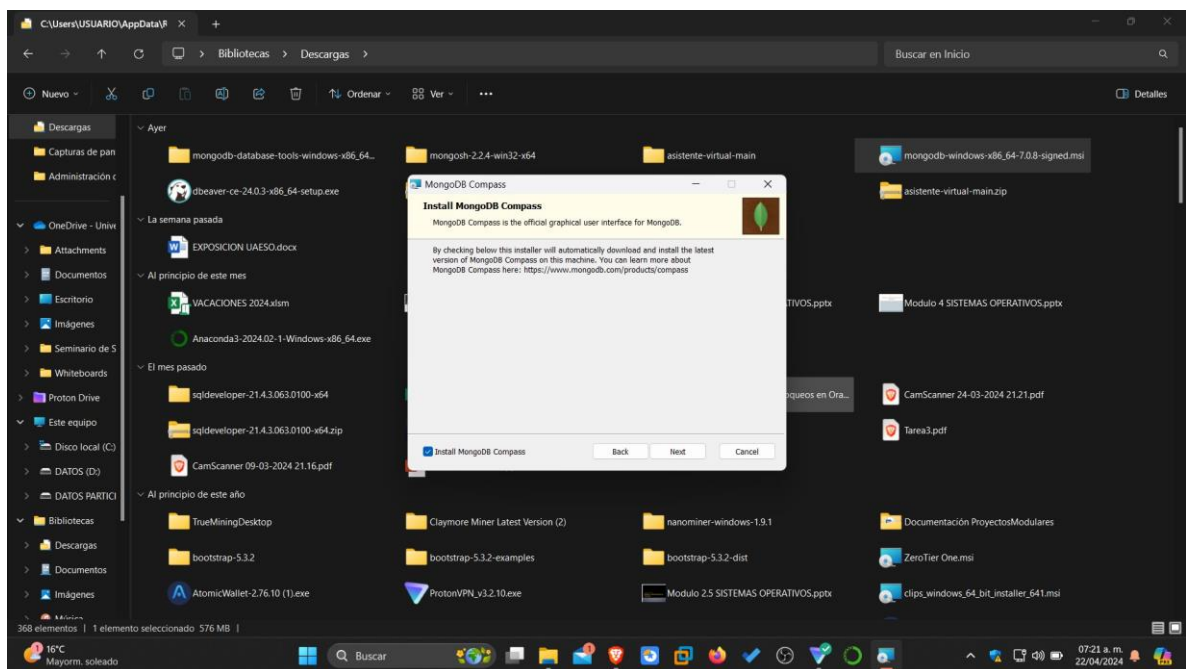
2. Instalamos MongoDB Community Edition dando clic en el archivo “.msi”, y damos clic en Next.



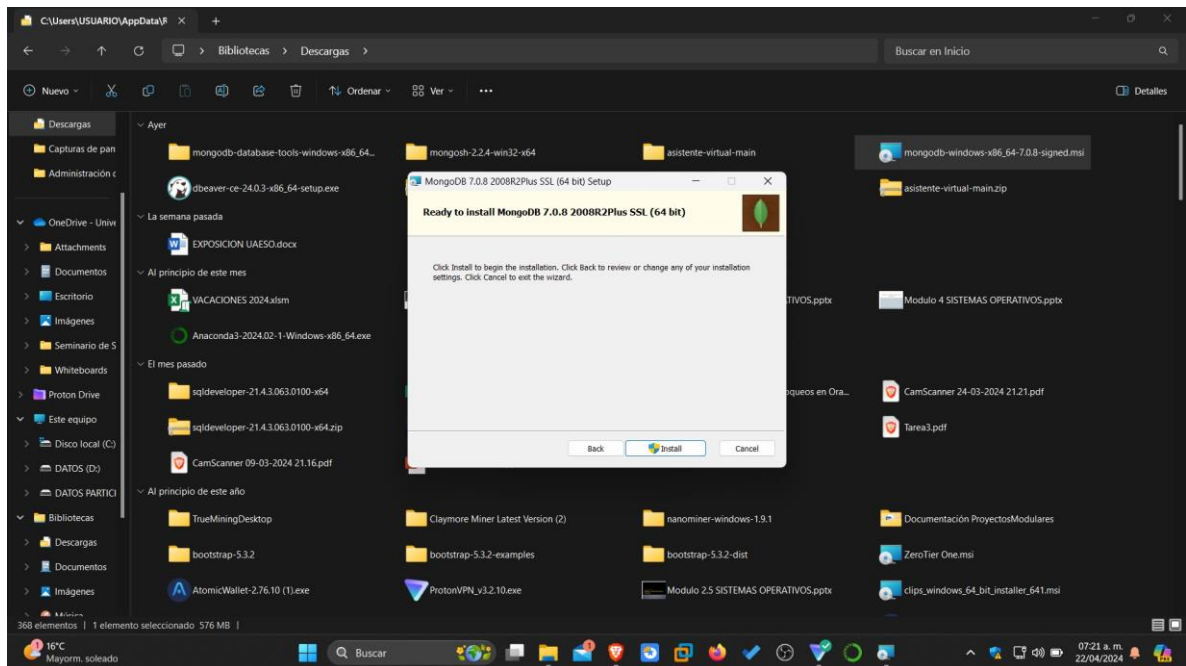
3. Aceptamos los términos y condiciones y damos clic en Next.



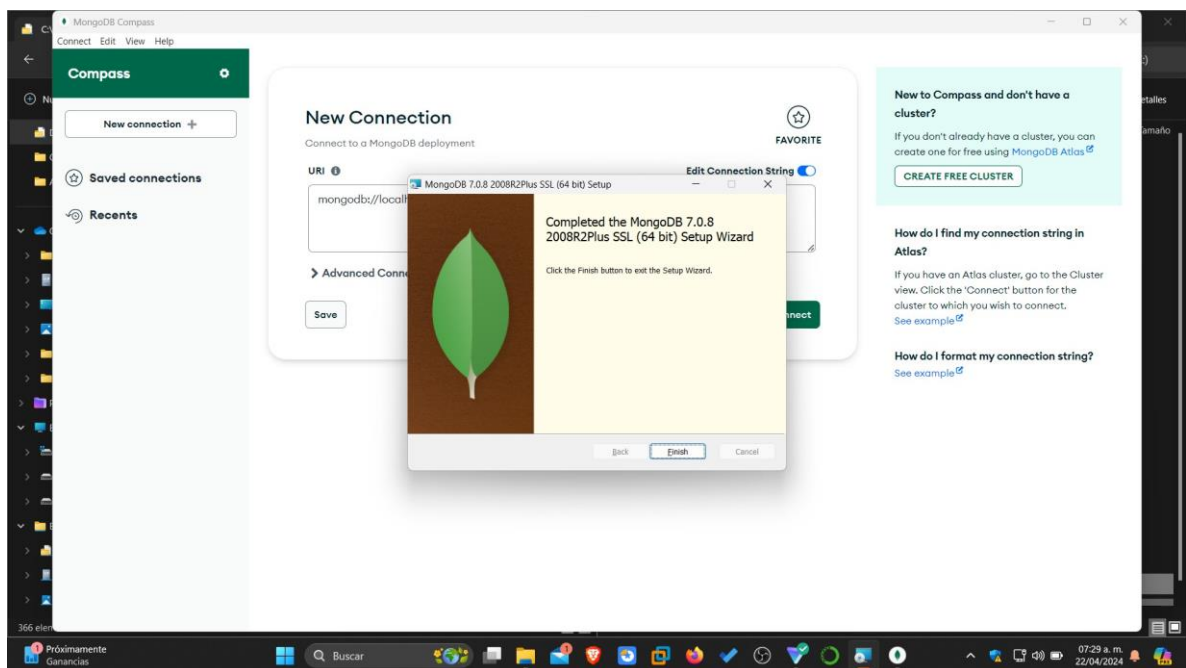
6. Activamos la casilla de instalar MongoDB Compass para la interfaz gráfica oficial, y damos en Next.



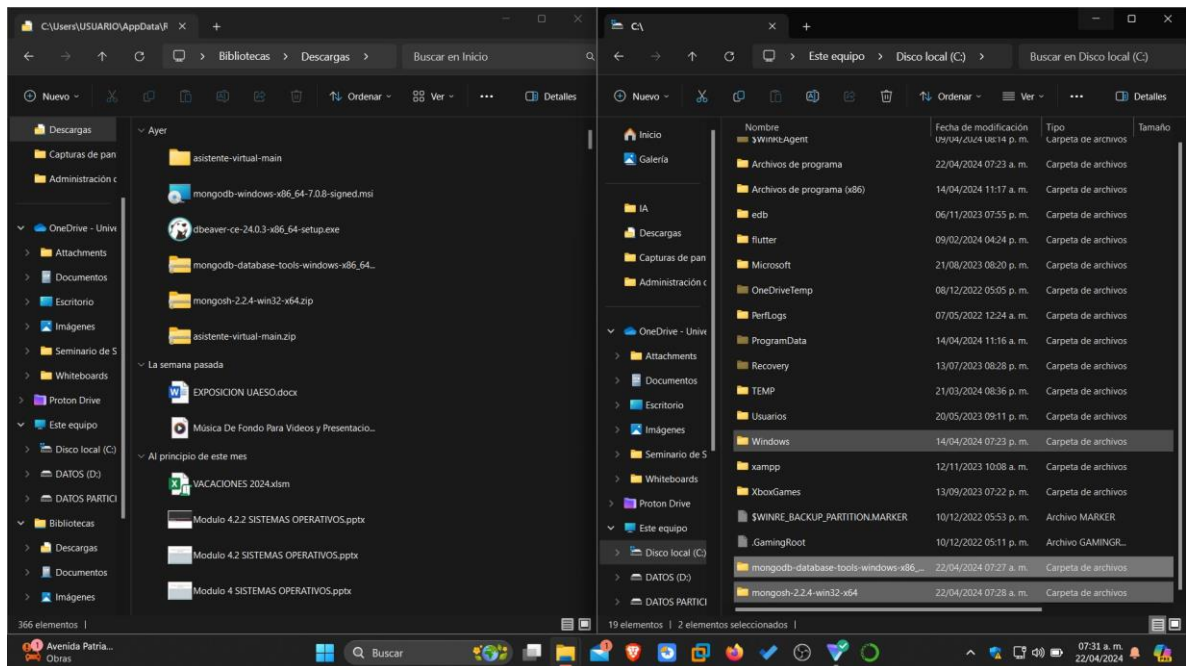
7. Damos clic en instalar



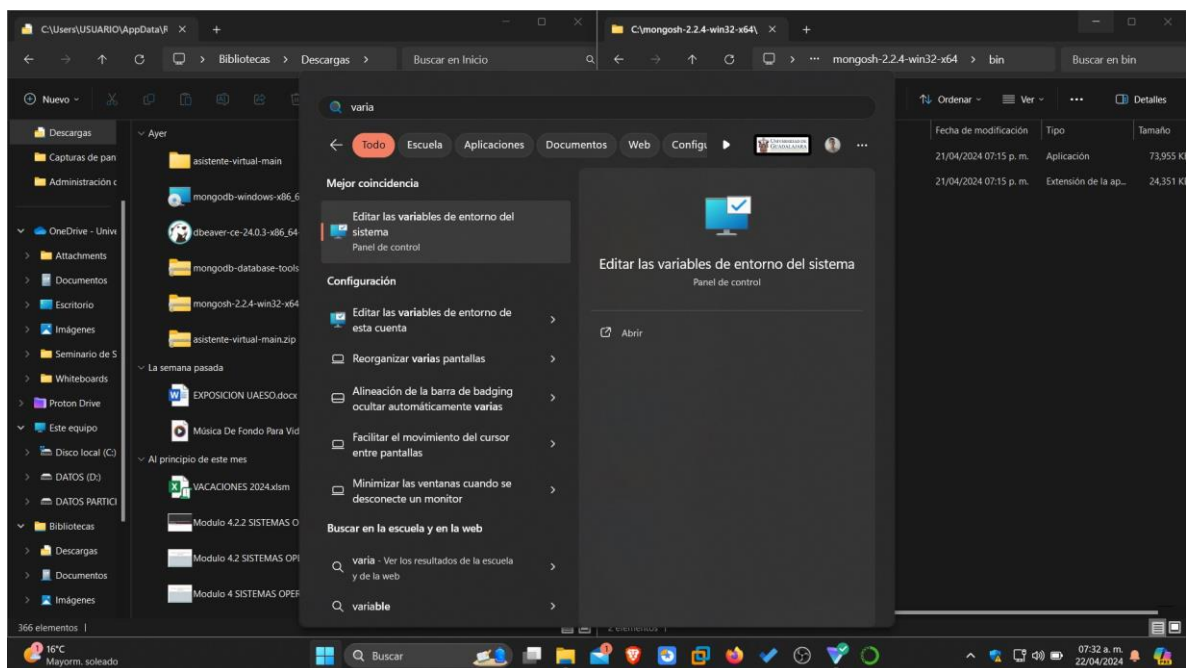
8. Damos clic en Finish



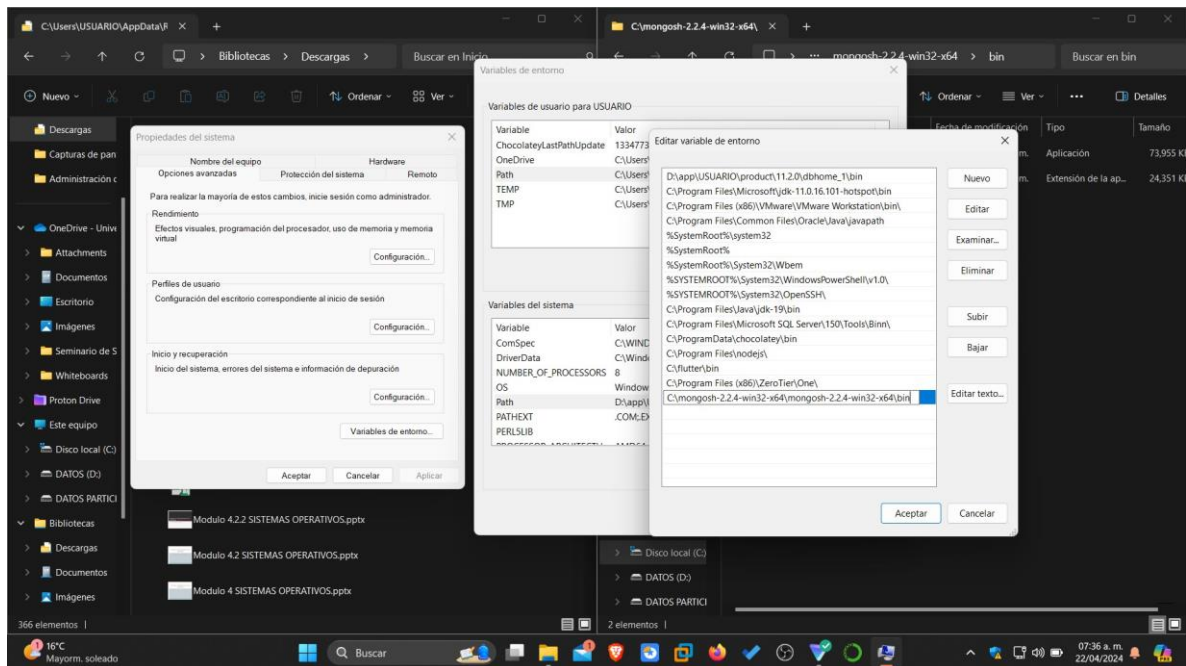
9. Extraemos las carpetas de mongosh y mongodb database tools en la raíz del disco C.



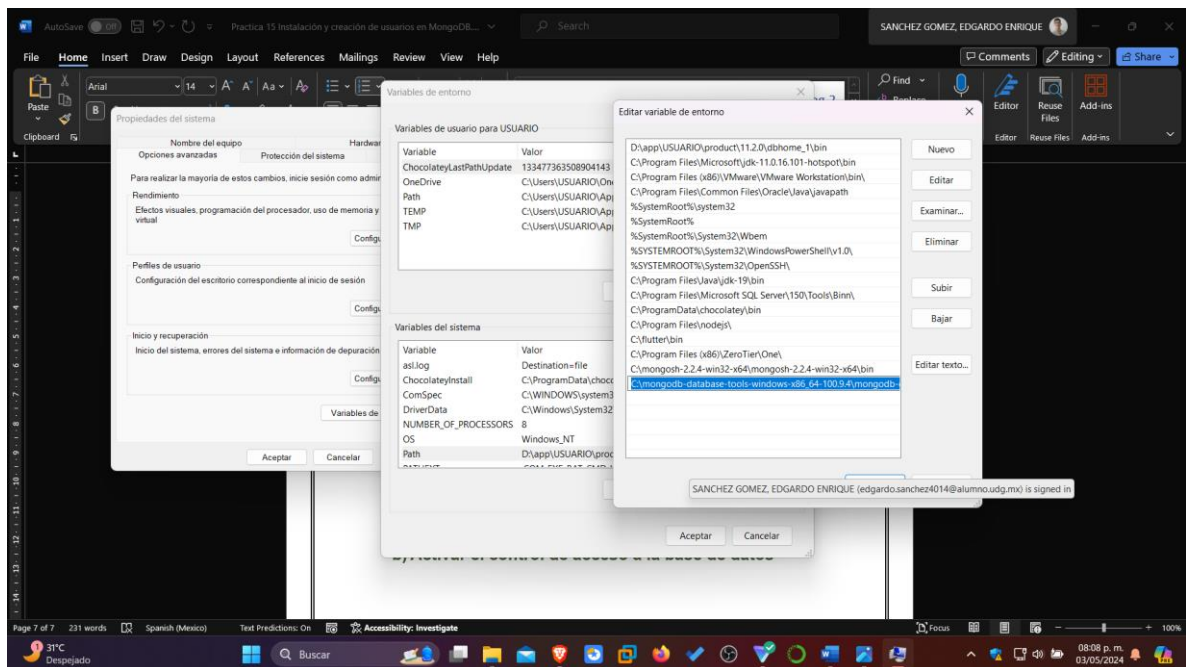
10. Abrimos la función de editar las variables del entorno



11. Abrimos la opción de variables de entorno, damos doble clic en la variable Path de variables del sistema, damos en nuevo para añadir una nueva variable, e ingresamos la dirección de la carpeta bin dentro de mongosh.



12. Repetimos el paso anterior, pero ahora agregamos la ruta bin de MongoDB database tools.



b) Activar el control de acceso a la base de datos

1. Abrimos un bloc de notas como administrador, y abrimos el archivo “mongod.cfg” ubicado en la ruta “C:\Program Files\MongoDB\Server\7.0\bin”, y des comentamos la línea de “security:”, además de añadir debajo la línea “

authorization: enabled” (Considera los 2 espacios al inicio de la línea, son importantes), y guardamos el archivo.

```

destination: file
logAppend: true
path: C:\Program Files\MongoDB\Server\7.0\log\mongod.log

# network interfaces
net:
  port: 27017
  bindIp: 127.0.0.1

#processManagement:

security:
  authorization: enabled

#operationProfiling:

#replication:

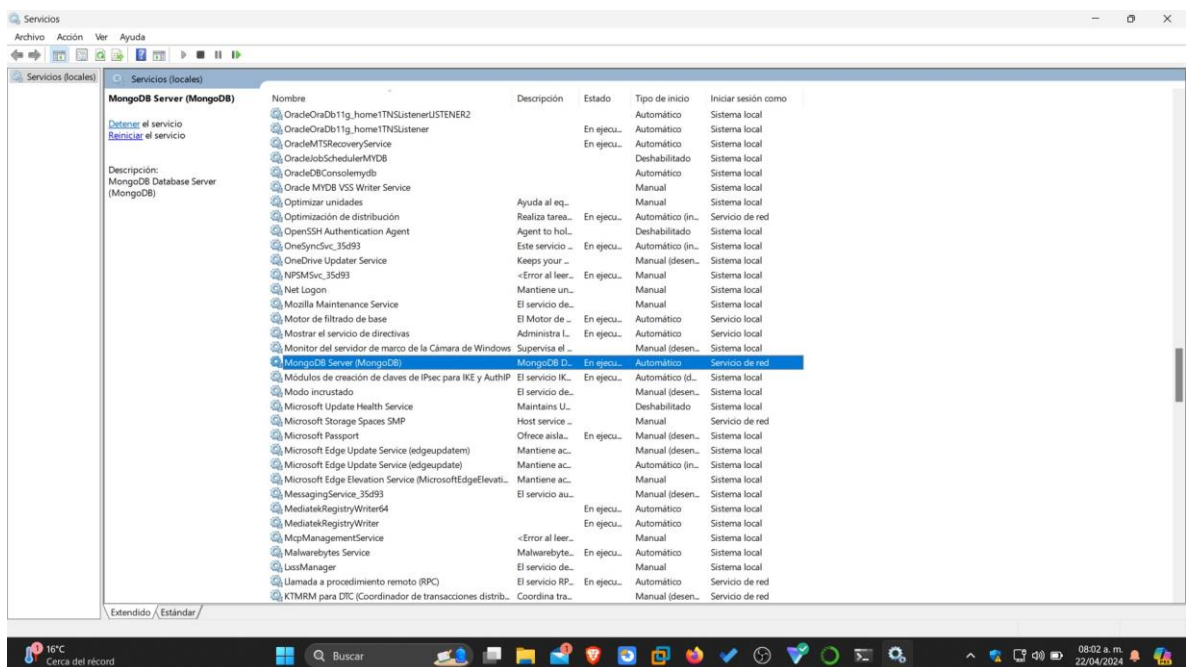
#sharding:

## Enterprise-Only Options:

#auditLog:

```

2. Abrimos los servicios del sistema y reiniciamos el servicio de “MongoDB Server (MongoDB)”.



3. Accedemos a MongoDB Shell desde una terminal con “mongosh”.


```
PS C:\Users\USUARIO> mongosh
Current Mongosh Log ID: 6635bc542fffe7d8e7117b7a
Connecting to:      mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000&appName=mongosh+2.2.4
Using MongoDB:      7.0.8
Using Mongosh:       2.2.4
mongosh 2.2.5 is available for download: https://www.mongodb.com/try/download/shell

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

test> use admin
switched to db admin
admin> db.createUser({
...   user: "admin",
...   pwd: passwordPrompt(),
...   roles: [
...     {role: "userAdminAnyDatabase", db: "admin"},
...     {role: "readWriteAnyDatabase", db: "admin"}
...   ]
... })
Enter password
*****{ ok: 1 }
admin>
```

4. Cambiamos (creamos) a una base de datos llamada “admin” con “use admin”.

```
PS C:\Users\USUARIO> mongosh
Current Mongosh Log ID: 6635bc542fffe7d8e7117b7a
Connecting to:      mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000&appName=mongosh+2.2.4
Using MongoDB:      7.0.8
Using Mongosh:       2.2.4
mongosh 2.2.5 is available for download: https://www.mongodb.com/try/download/shell

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

test> use admin
switched to db admin
admin> db.createUser({
...   user: "admin",
...   pwd: passwordPrompt(),
...   roles: [
...     {role: "userAdminAnyDatabase", db: "admin"},
...     {role: "readWriteAnyDatabase", db: "admin"}
...   ]
... })
Enter password
*****{ ok: 1 }
admin>
```

5. Ejecutamos el script correspondiente para crear un usuario admin, con los roles correspondientes para la base de datos admin, utilizando “passwordPromp()” para que después de ejecutarlo, solicite la contraseña, y esta no se muestre al escribirla.

```
mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000&appName=mongosh+2.2.4
PS C:\Users\USUARIO> mongosh
Current Mongosh Log ID: 6635bc542fffe7d8e7117b7a
Connecting to:      mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000&appName=mongosh+2.2.4
Using MongoDB:      7.0.8
Using Mongosh:       2.2.4
mongosh 2.2.5 is available for download: https://www.mongodb.com/try/download/shell

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

test> use admin
switched to db admin
admin> db.createUser({
...   user: "admin",
...   pwd: passwordPrompt(),
...   roles: [
...     {role: "userAdminAnyDatabase", db: "admin"},
...     {role: "readWriteAnyDatabase", db: "admin"}
...   ]
... })
Enter password
*****{ ok: 1 }
admin>
```

6. Accedemos desde otra terminal al usuario que acabamos de crear con “mongosh -u admin --authenticationDatabase admin”.

```
mongosh mongodb://<credentials>@127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000&authSource=admin&appName=mongosh+2.2.4
PS C:\Users\USUARIO> mongosh -u admin --authenticationDatabase admin
Enter password: *****
Current Mongosh Log ID: 6635c12837cf7d7a58117b7a
Connecting to:      mongodb://<credentials>@127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000&authSource=admin&appName=mongosh+2.2.4
Using MongoDB:      7.0.8
Using Mongosh:       2.2.4
mongosh 2.2.5 is available for download: https://www.mongodb.com/try/download/shell

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

test> db.createUser({
...   user: "edgardon",
...   pwd: passwordPrompt(),
...   roles: [
...     {role: "readWrite", db: "tienda"},
...     {role: "read", db: "escuela"}
...   ]
... })
Enter password
*****{ ok: 1 }
test> |
```

7. Con el script adecuado creamos un usuario mortal con permisos de lectura y escritura para una base de datos, y de solo lectura para otra base de datos.

```
mongosh mongodb://<creds> x + v
PS C:\Users\USUARIO> mongosh -u admin --authenticationDatabase admin
Enter password: *****
Current Mongosh Log ID: 6635c12837cf7d7a58117b7a
Connecting to:      mongodb://<credentials>@127.0.0.1:27017/?directConnection=true&serverSelectionT
imeoutMS=2000&authSource=admin&appName=mongosh+2.2.4
Using MongoDB:      7.0.8
Using Mongosh:      2.2.4
mongosh 2.2.5 is available for download: https://www.mongodb.com/try/download/shell

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

test> db.createUser({
...   user: "edgardom",
...   pwd: passwordPrompt(),
...   roles: [
...     {role: "readWrite", db: "tienda"},
...     {role: "read", db: "escuela"}
...   ]
... })
Enter password
*****{ ok: 1 }
test> |
```

8. Desde una nueva terminal accedemos al usuario mortal que acabamos de crear, recordando que en el parámetro “--authenticationDatabase” se coloca el nombre de la base de datos en la que se creó (En este caso en test).

```
mongosh mongodb://<creds> x + v
PS C:\Users\USUARIO> mongosh -u edgardom --authenticationDatabase test
Enter password: *****
Current Mongosh Log ID: 6636549b76f230b84b117b7a
Connecting to:      mongodb://<credentials>@127.0.0.1:27017/?directConnection=true&serverSelec
tionTimeoutMS=2000&authSource=test&appName=mongosh+2.2.4
Using MongoDB:      7.0.8
Using Mongosh:      2.2.4
mongosh 2.2.5 is available for download: https://www.mongodb.com/try/download/shell

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

test>
```

9. Para comprobar que tenemos asignados correctamente los permisos, accedemos a la base de datos “tienda” con “use tienda”, creamos una colección, e insertamos algunos documentos en la colección creada para corroborar los permisos de escritura.

```
mongosh mongodb://c-rede x + v
For mongosh info see: https://docs.mongodb.com/mongodb-shell/

test> use tienda
switched to db tienda
tienda> db.createCollection("productos")
{ ok: 1 }
tienda> db.productos.insertMany([
...   {
...     nombre: "Refresco",
...     precio: 15.24
...   },
...   {
...     nombre: "Papas",
...     precio: 12
...   }
... ])
{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId('66365b4976f230b84b117b7b'),
    '1': ObjectId('66365b4976f230b84b117b7c')
  }
}
tienda> |
```

10. Ejecutamos “db.productos.find()” para ver los documentos registrados en la colección de productos, para corroborar el permiso de lectura en la base de datos.

```
mongosh mongodb://c-rede x + v
...   }
... ])
{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId('66365b4976f230b84b117b7b'),
    '1': ObjectId('66365b4976f230b84b117b7c')
  }
}
tienda> db.tienda.find()

tienda> db.productos.find()
[
  {
    _id: ObjectId('66365b4976f230b84b117b7b'),
    nombre: 'Refresco',
    precio: 15.24
  },
  {
    _id: ObjectId('66365b4976f230b84b117b7c'),
    nombre: 'Papas',
    precio: 12
  }
]
tienda> |
```

11. Notamos que, si queremos hacer lo mismo para la base de datos de escuela, no nos lo permite, ya que en esa base de datos solo tenemos permisos de lectura.

```
mongosh mongodb://<cred> x + v
  '0': ObjectId('66365b4976f230b84b117b7b'),
  '1': ObjectId('66365b4976f230b84b117b7c')
}
}
tienda> db.tienda.find()

tienda> db.productos.find()
[
  {
    _id: ObjectId('66365b4976f230b84b117b7b'),
    nombre: 'Refresco',
    precio: 15.24
  },
  {
    _id: ObjectId('66365b4976f230b84b117b7c'),
    nombre: 'Papas',
    precio: 12
  }
]

tienda> use escuela
switched to db escuela
escuela> db.createCollection("alumnos")
MongoServerError[Unauthorized]: not authorized on escuela to execute command { create: "alumnos",
lsid: { id: UUID("aabb5f85-5178-46ff-90cc-651db8604df3") }, $db: "escuela" }
escuela>
```

12. Desde otro usuario con permisos de escritura en la base de datos escuela (Como el usuario admin), creamos una colección con documentos.

```
PS C:\Users\USUARIO> mongosh -u admin --authenticationDatabase admin
Enter password: *****
Current Mongosh Log ID: 66366564fcad757782117b7a
Connecting to: mongod://<credentials>@127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000&authSource=admin&appName=mongosh+2.2.4
Using MongoDB: 7.0.8
Using Mongosh: 2.2.4
mongosh 2.2.5 is available for download: https://www.mongodb.com/try/download/shell

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

test> use escuela
switched to db escuela
escuela> db.createCollection("alumnos")
{ ok: 1 }
escuela> db.alumnos.insertMany([
... {
...   codigo: "1215659F",
...   nombre: "Pedro Salazar",
...   edad: 20
... },
... {
...   codigo: "46416RTFA3",
...   nombre: "Juan Gonzalez",
...   edad: 21
... }
... ])
{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId('6636661cfcad757782117b7b'),
    '1': ObjectId('6636661cfcad757782117b7c')
  }
}
escuela>
```

13. Desde el usuario mortal ejecutamos una consulta para ver las colecciones en la base de datos “escuela”, y ejecutamos otra consulta para ver los documentos en dicha colección; con esto comprobamos que tiene asignado correctamente el permiso de lectura en esa base de datos.


```
mongosh mongodb://c-crede x mongosh mongodb://c-crede x + v
}
]
tienda> use escuela
switched to db escuela
escuela> db.createCollection("alumnos")
MongoServerError[Unauthorized]: not authorized on escuela to execute command { create: "alumnos",
1c-id: { _id: UUID("3abb5f85-5178-465f-80cc-651db8604df3") }, $db: "escuela" }
escuela> show collections
alumnos
escuela> db.alumnos.find()
[
  {
    _id: ObjectId('6636661cfcad757782117b7b'),
    codigo: '12156S9F',
    nombre: 'Pedro Salazar',
    edad: 20
  },
  {
    _id: ObjectId('6636661cfcad757782117b7c'),
    codigo: '46416RTFA3',
    nombre: 'Juan Gonzalez',
    edad: 21
  }
]
escuela> |
```

28°C
Soleado

Buscar

10:48 a. m.
04/05/2024