



Administración de bases de datos

D07

M-J 7-9 AM



Práctica 15: Instalación y creación de usuarios en MongoDB

Centro Universitario de Ciencias Exactas e Ingenierías

Universidad de Guadalajara

2024^a

a) Reporte de instalación de MongoDB

Se abrirán esos 3 links para descargar los componentes a instalar.

The screenshot shows a Google Classroom announcement titled "Administración de Bases de Datos". The announcement contains three links:

- MongoDB Community Server: <https://www.mongodb.com/try/download/community>
- MongoDB Shell: <https://www.mongodb.com/try/download/shell>
- MongoDB Database Tools: <https://www.mongodb.com/try/download/database-tools>

Se descarga el mongoDB command line database.

The screenshot shows the MongoDB Database Tools download page. The "MongoDB Command Line Database Tools Download" section is highlighted. It includes dropdown menus for Version (set to 100.9.4), Platform (Windows x86_64), and Package (zip). A "Download" button is visible at the bottom.

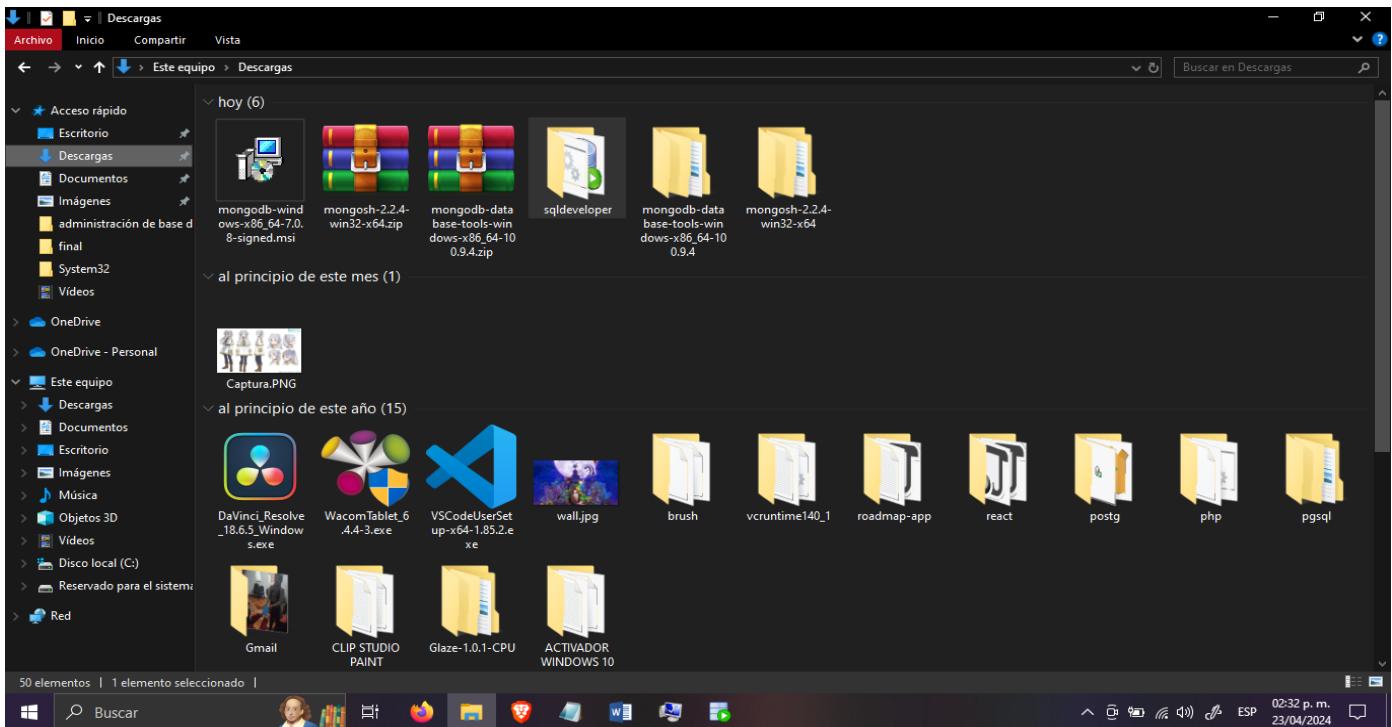
Se descarga el community server.

The screenshot shows a web browser window with the URL [mongodb.com/try/download/community](https://www.mongodb.com/try/download/community). The page is titled "MongoDB Community Server Download". On the left, there's a sidebar with links to various MongoDB products: MongoDB Atlas, MongoDB Enterprise Advanced, MongoDB Community Edition, MongoDB Community Server (which is highlighted), MongoDB Community Kubernetes Operator, Tools, Atlas SQL Interface, and Mobile & Edge. The main content area features a "MONGOOSE COMMUNITY SERVER" logo and a section titled "MongoDB Community Server Download". It describes the database as offering a flexible document data model with support for ad-hoc queries, secondary indexing, and real-time aggregations. Below this, it mentions MongoDB Atlas as a fully-managed service with advanced functionality like auto-scaling and full-text search. A "Give it a try with a free, highly-available 512 MB cluster" button is present. A terminal command box shows "\$ brew install mongodb-atlas" and "\$ atlas setup". At the bottom, there's a dropdown menu for "Version" set to "7.0.9 (current)". The browser's address bar shows "Buscar" and the taskbar at the bottom includes icons for File Explorer, Task View, Start, and other pinned apps.

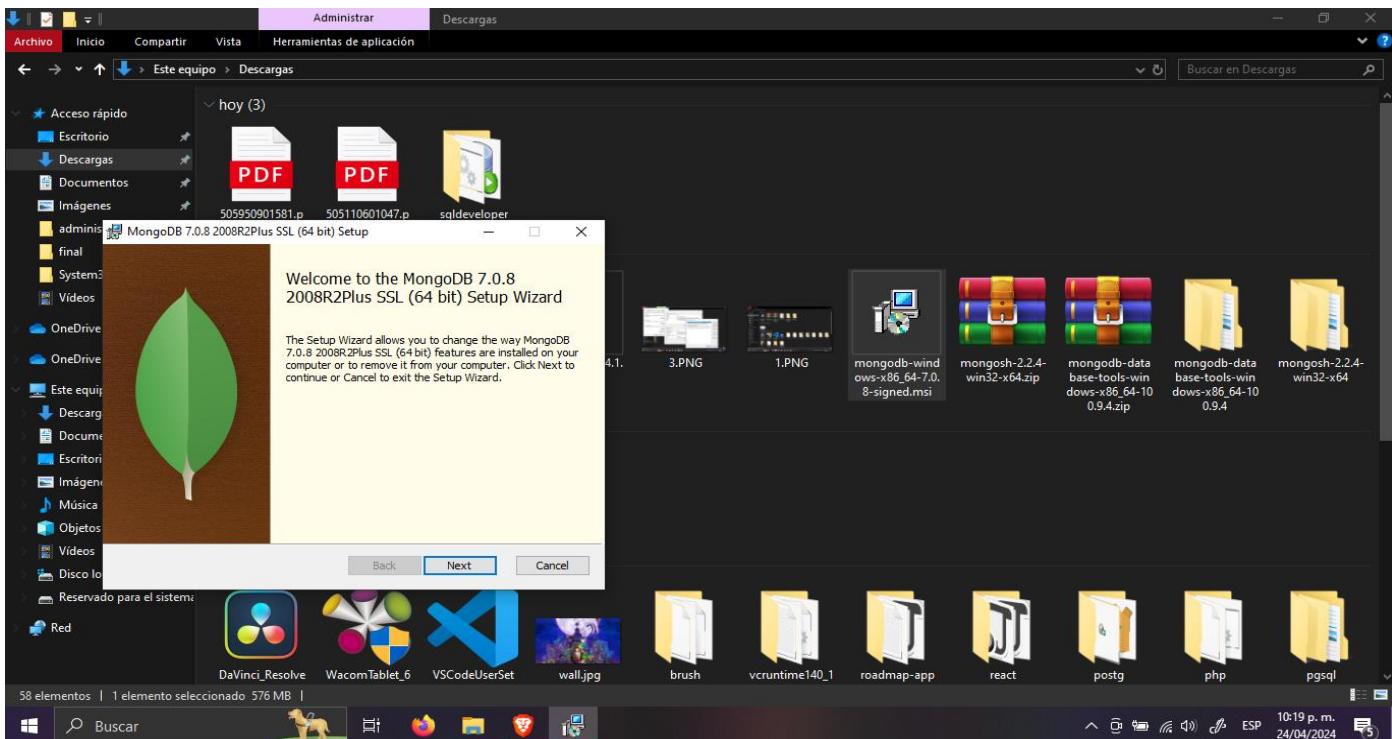
Y se instala el mongoDB Shell download.

The screenshot shows a web browser window with the URL [mongodb.com/try/download/shell](https://www.mongodb.com/try/download/shell). The page is titled "MongoDB Shell Download". The sidebar on the left lists: MongoDB Atlas, MongoDB Enterprise Advanced, MongoDB Community Edition, Tools (highlighted), MongoDB Shell (which is also highlighted), MongoDB Compass (GUI), Atlas CLI, Atlas Kubernetes Operator, MongoDB CLI for Cloud Manager and Ops Manager, MongoDB Cluster-to-Cluster Sync, Relational Migrator, MongoDB Database Tools, MongoDB Connector for BI, App Services CLI, and Atlas SQL Interface. The main content area features a "TOOLS" logo and a section titled "MongoDB Shell Download". It describes MongoDB Shell as the quickest way to connect to MongoDB, mentioning its modern, extensible command-line interface with syntax highlighting, intelligent autocomplete, contextual help, and error messages. A compatibility note states that Red Hat Enterprise Linux (RHEL) 7, Amazon Linux 2, SUSE Linux Enterprise Server (SLES) 12, and Ubuntu 18.04 support is deprecated and might be removed in a later mongosh release. A note also states that MongoDB Shell is an open source product developed separately from the MongoDB Server. A "Learn more" link is provided. At the bottom, there are dropdown menus for "Version" set to "2.2.5", "Platform" set to "Windows x64 (10+)", and "Package" set to "zip". The browser's address bar shows "Buscar" and the taskbar at the bottom includes icons for File Explorer, Task View, Start, and other pinned apps.

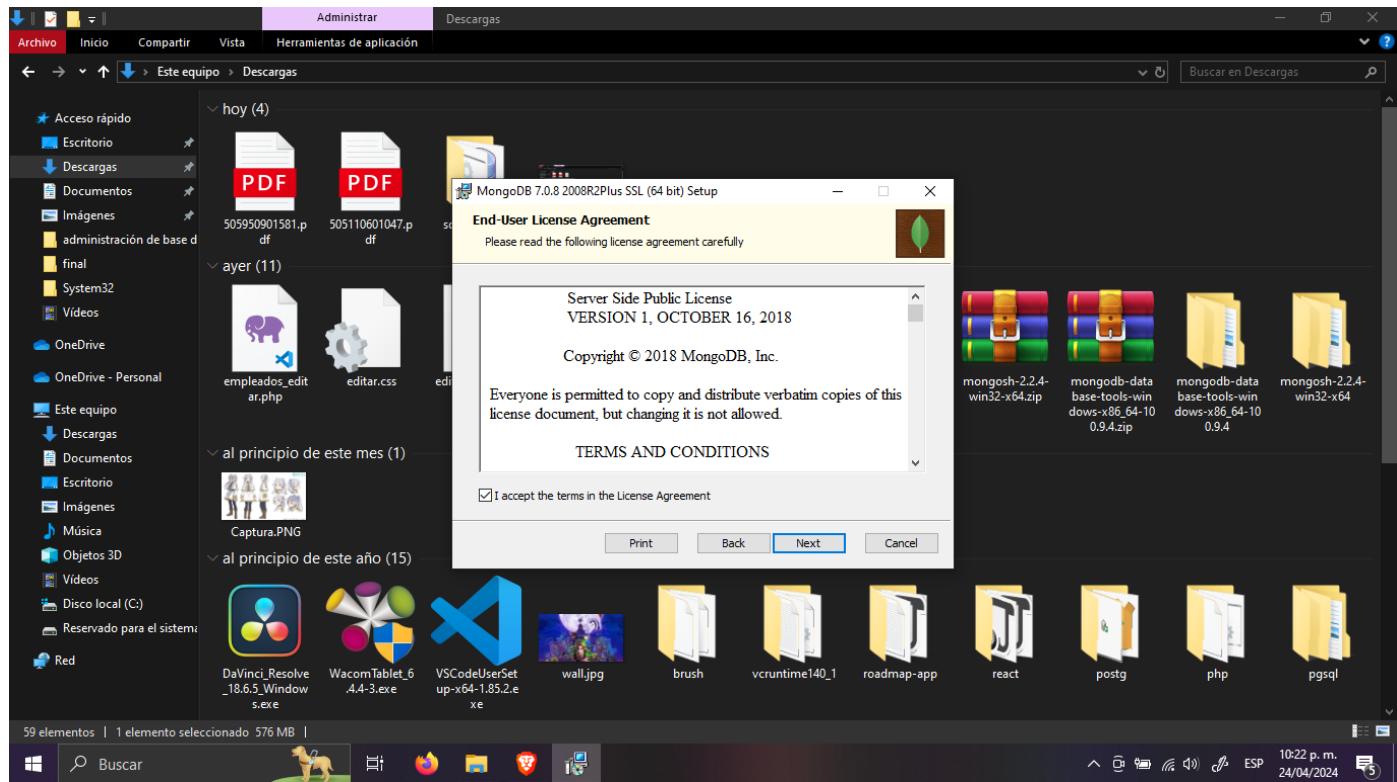
Para empezar, se tienen que tener descargados los elementos esenciales para instalar mongoDB:



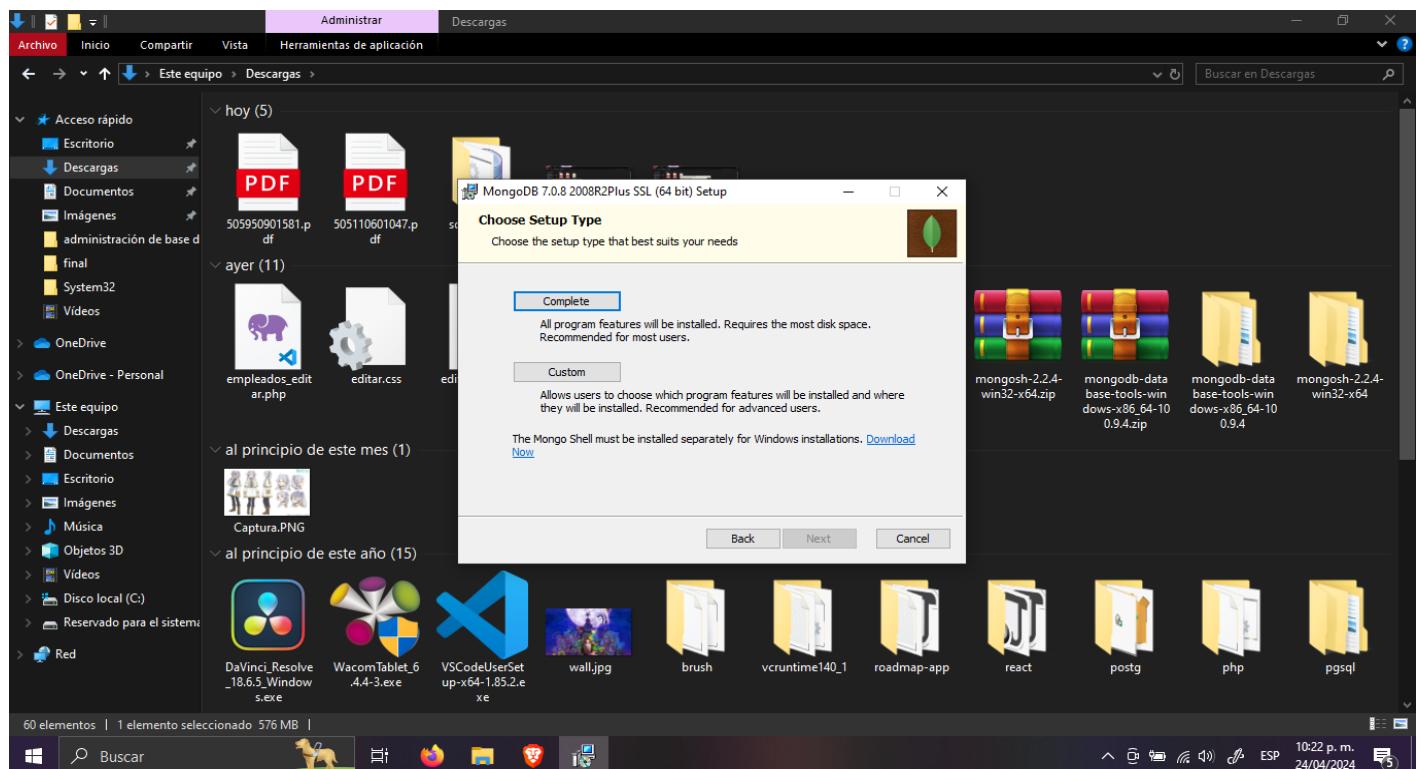
Se ejecuta el archivo msi para iniciar la instalación de mongodb. El cual abrirá esta pestaña que nos permitirá iniciar la instalación.



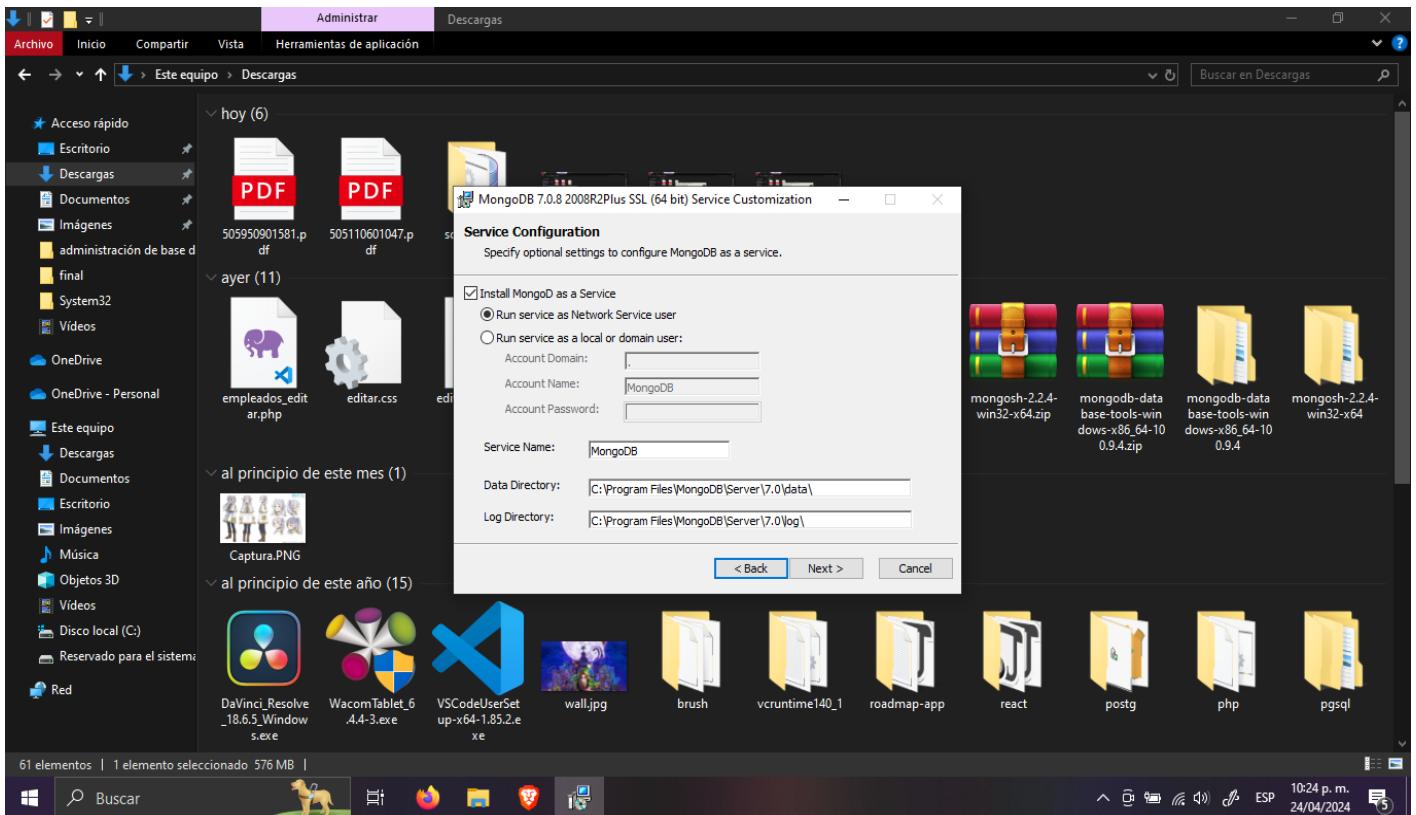
Aceptamos las condiciones de uso y sus términos que si leímos.



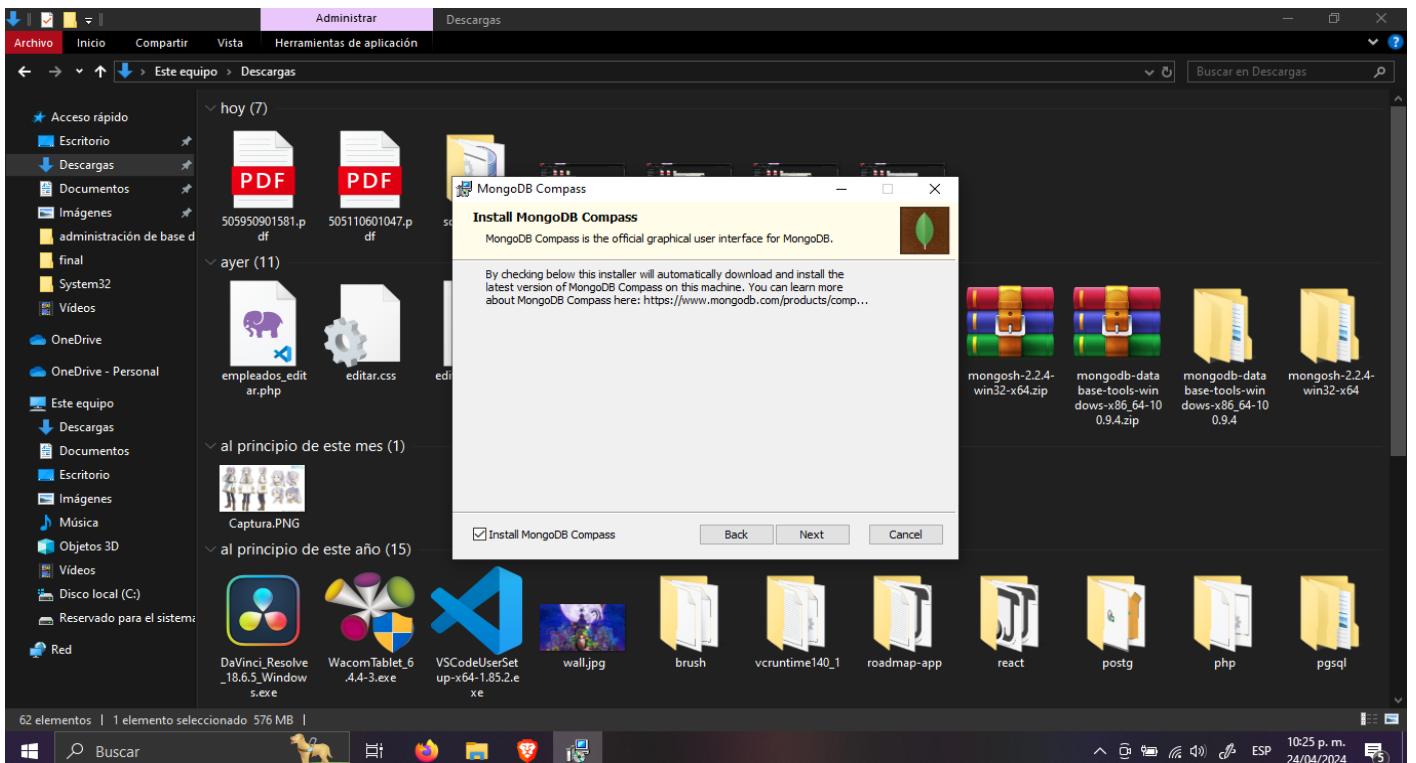
Seleccionamos el setup completo.



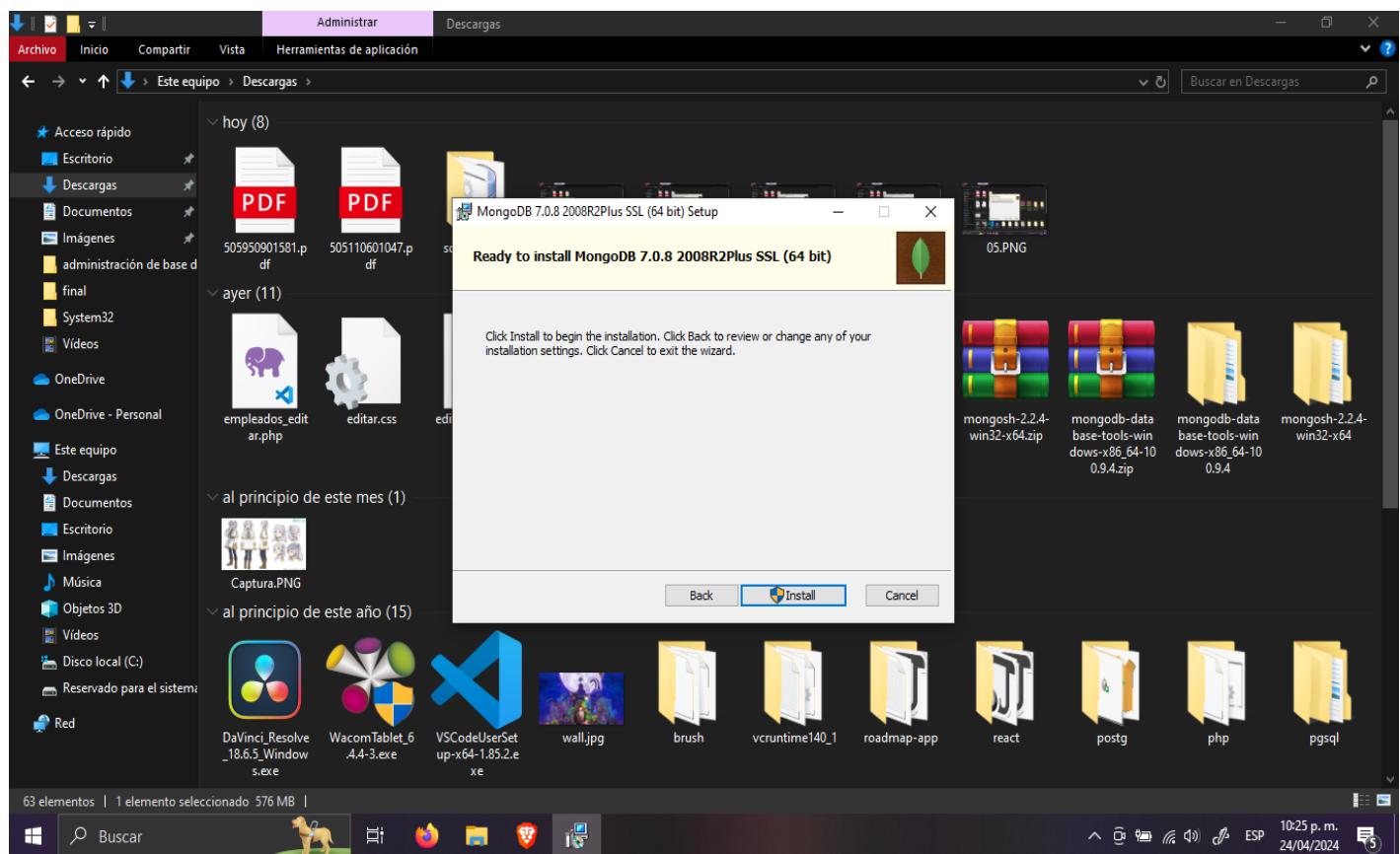
El software se ejecuta como servicio.



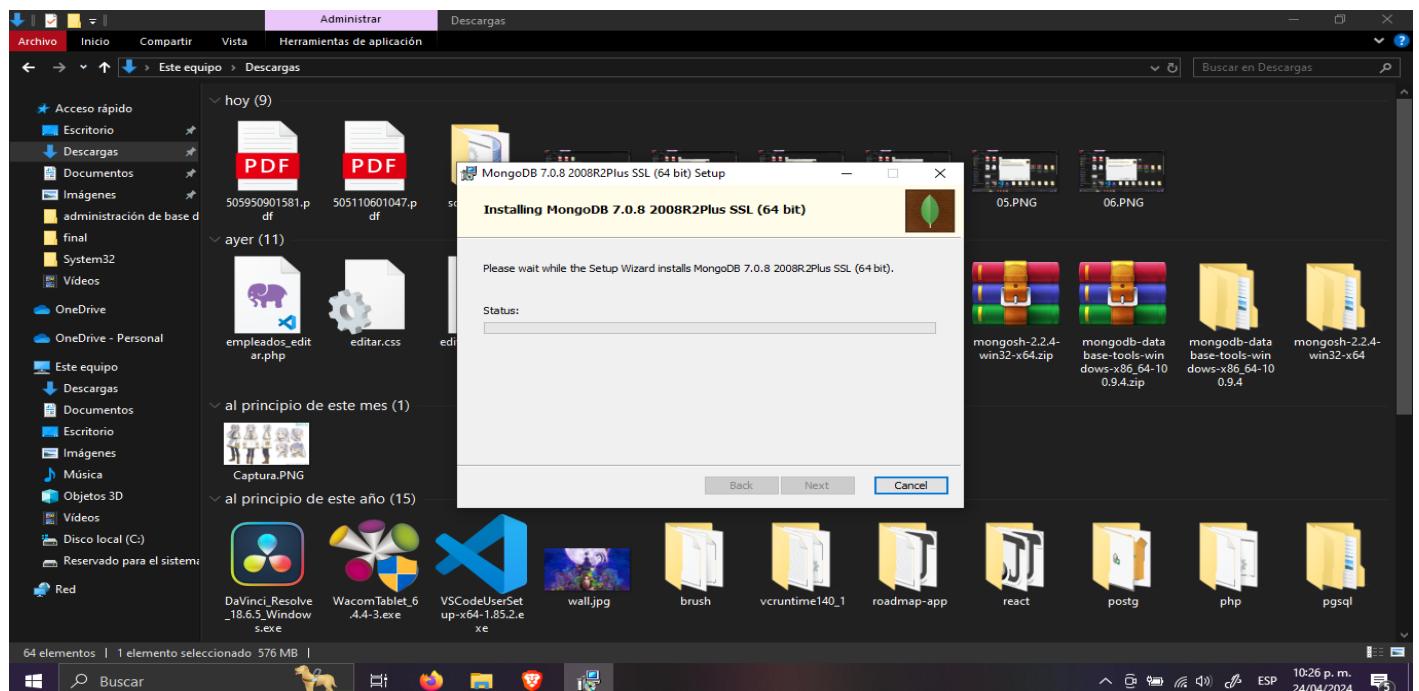
Se le da siguiente y se instala el compass por si acaso.



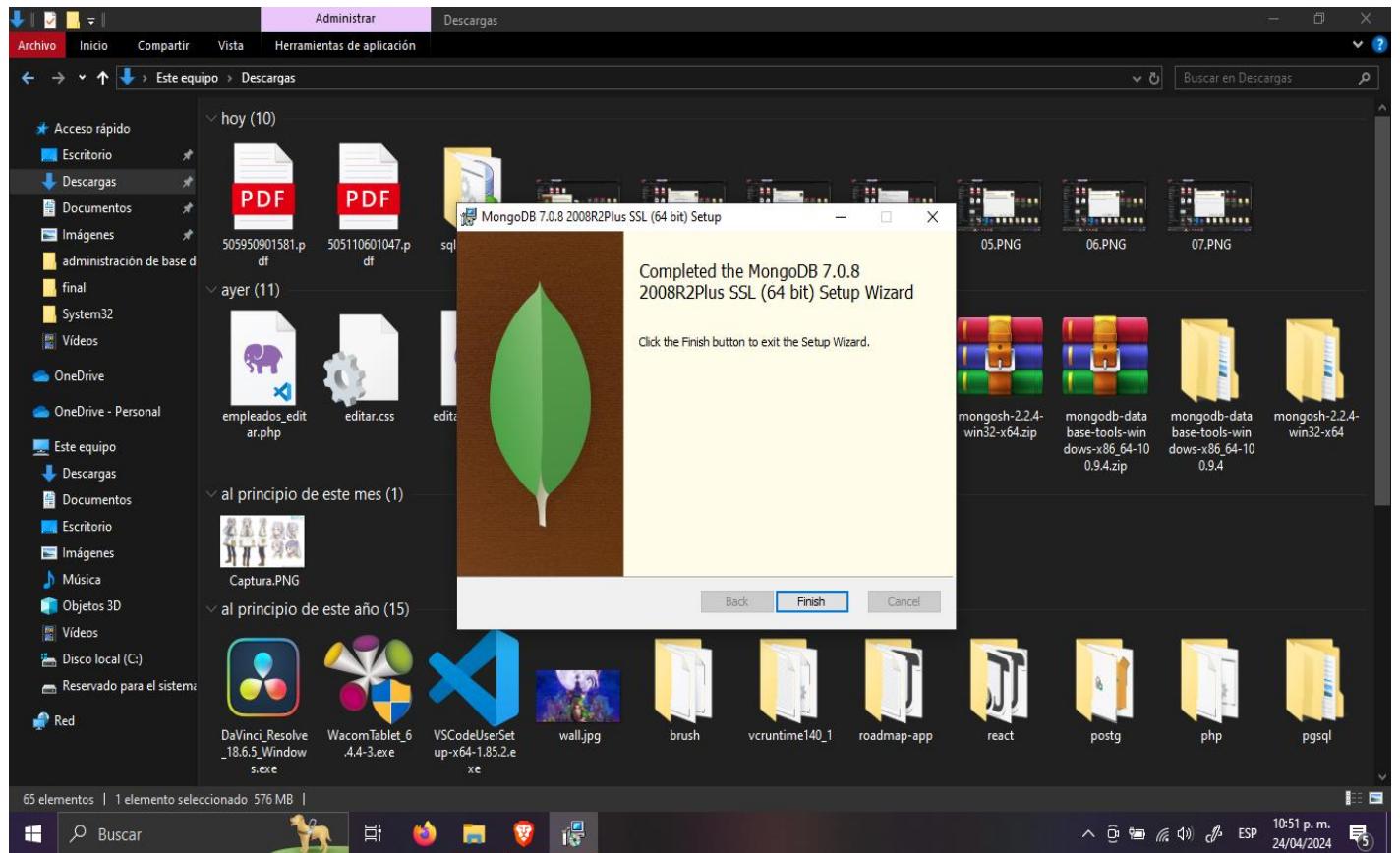
Se da click para iniciar la instalación.



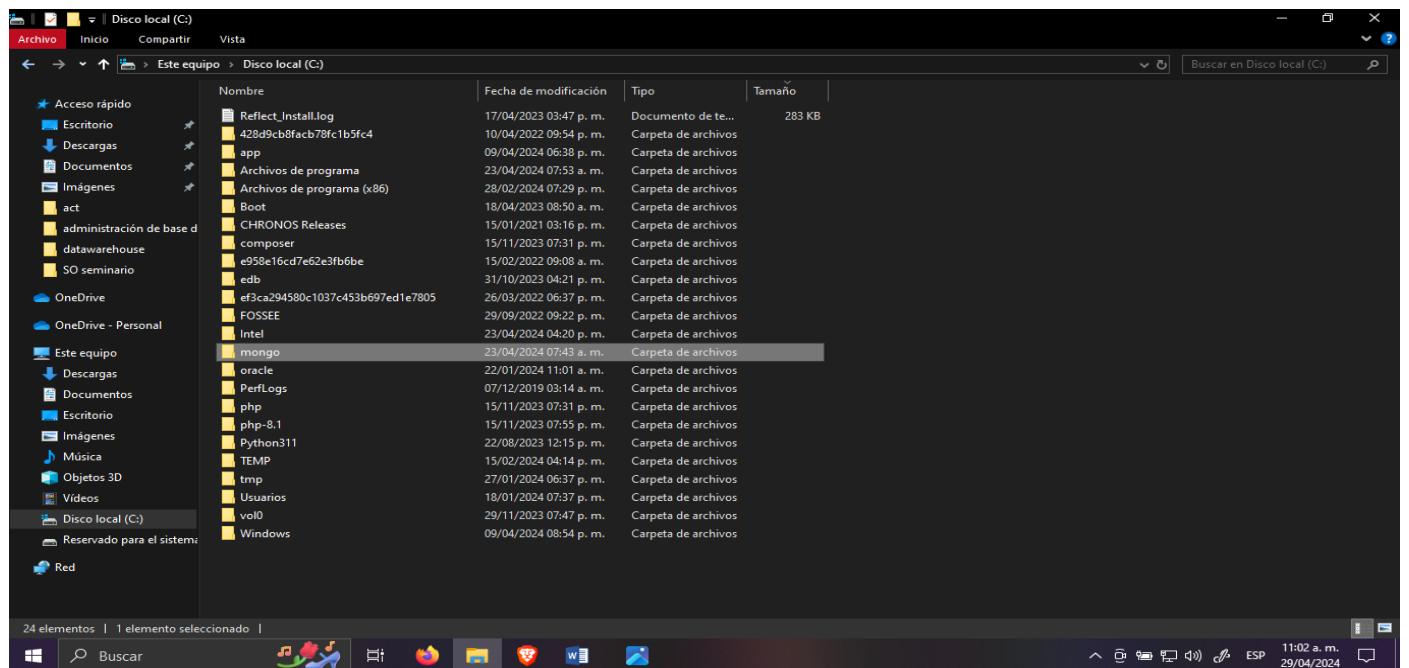
Se instalan los paquetes.



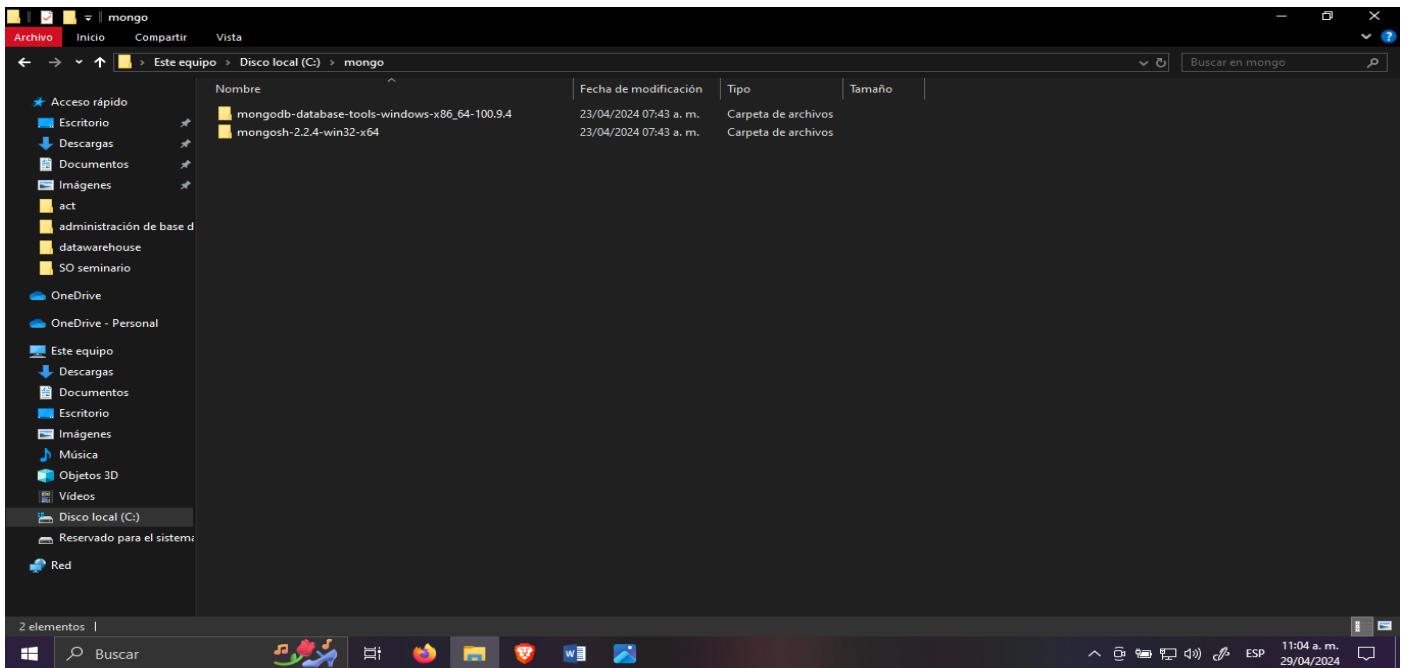
Se le da click en finalizar.



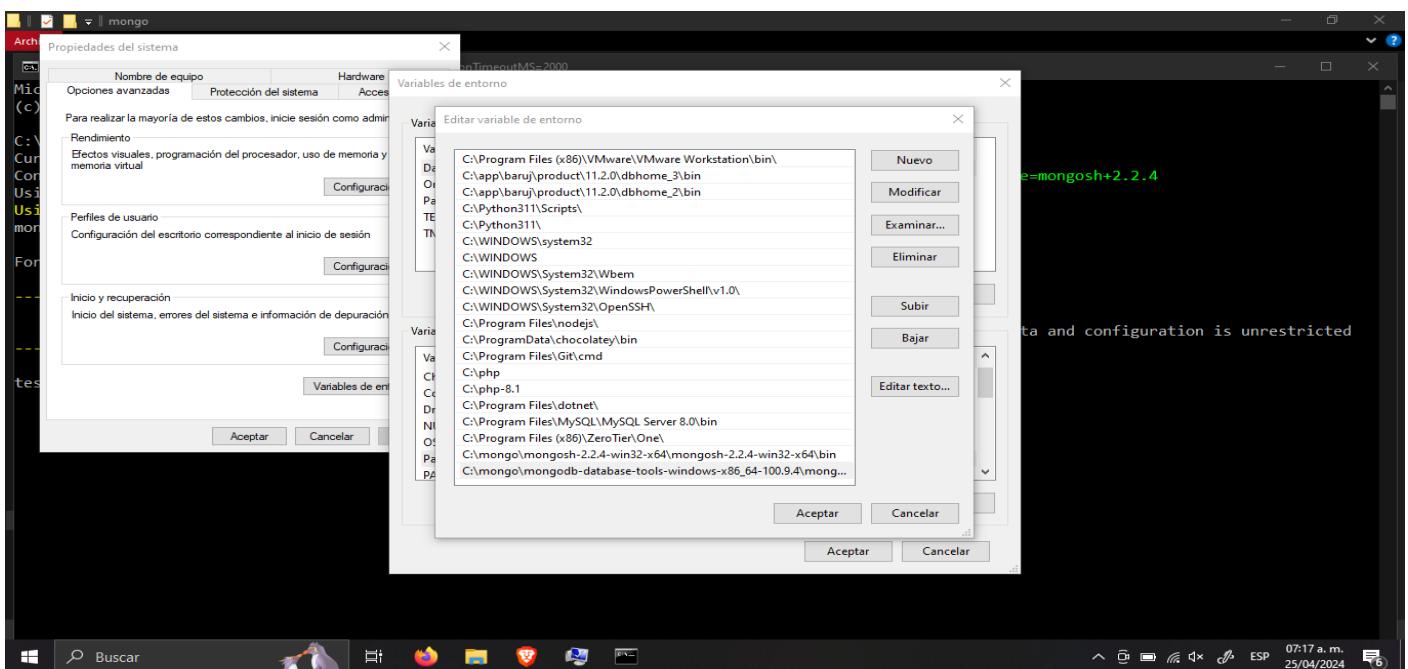
Se crea una carpeta que servirá para organizar los archivos que se han de descomprimir.



Se descomprimen los archivos de tolos y Shell, para posteriormente ponerlos en este directorio.

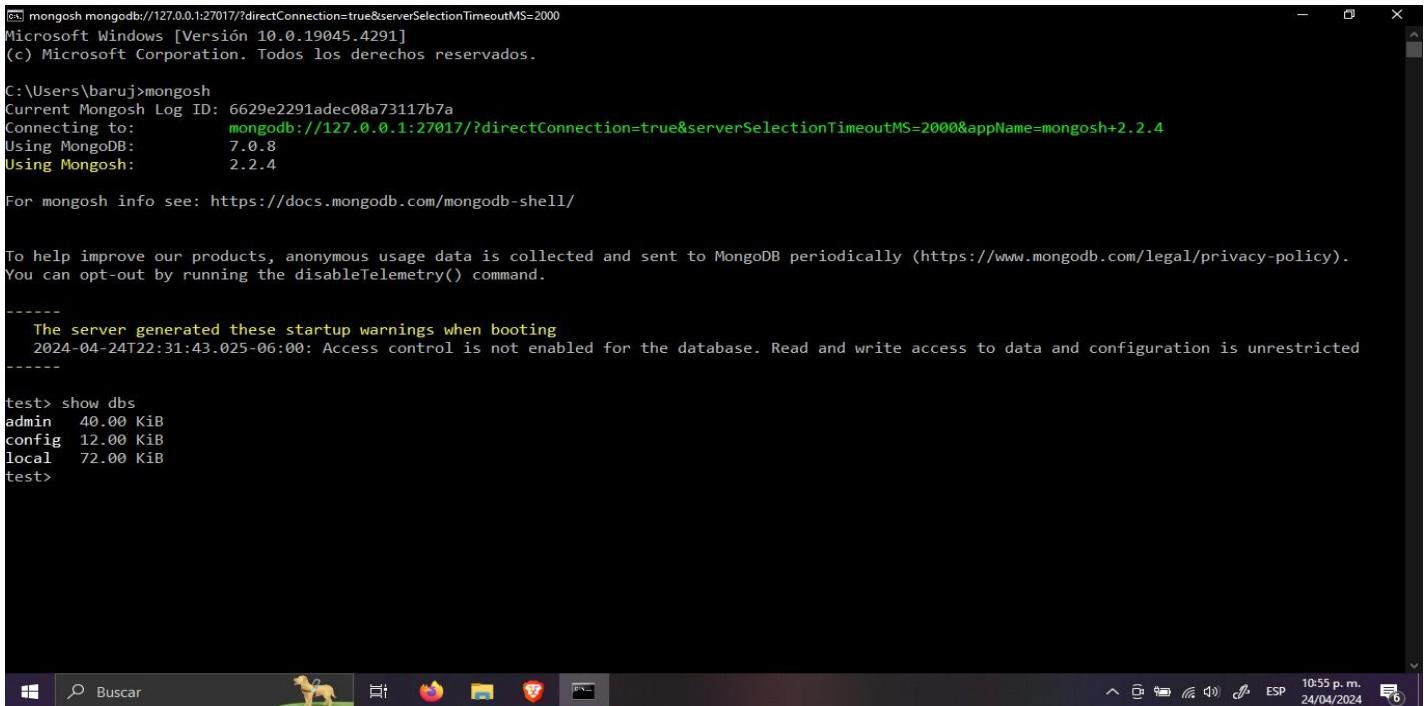


En propiedades de sistema, en variables de entorno y en PATH, se añadirán las direcciones de los BIN de ambos archivos que se descomprimieron.



Con esto ya podemos iniciar mongodb, utilizando mongosh en el CMD.

Por lo que se pudo conectar y además se pueden visualizar las bases de datos del sistema.



```
mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000
Microsoft Windows [Versión 10.0.19045.4291]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\baruj>mongosh
Current Mongosh Log ID: 6629e2291a08a73117b7a
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

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

To help improve our products, anonymous usage data is collected and sent to MongoDB periodically (https://www.mongodb.com/legal/privacy-policy).
You can opt-out by running the disableTelemetry() command.

-----
The server generated these startup warnings when booting
2024-04-24T22:31:43.025-06:00: Access control is not enabled for the database. Read and write access to data and configuration is unrestricted
-----
test> show dbs
admin   40.00 KiB
config  12.00 KiB
local   72.00 KiB
test>
```

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

Crear un usuario administrador con sus respectivos roles.

En la línea de comandos, se ingresa el siguiente comando para crear un usuario con los privilegios de administrador.

```
db.createUser({
  user: "admon",
  pwd: passwordPrompt(),
  roles: [
    {role: "userAdminAnyDatabase", db: "admin"},
    {role: "readWriteAnyDatabase", db: "admin"}
  ]
})
```

```
mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000
...
})
Enter password
TypeError: db.CreateUser is not a function
admin> db.createUser({
...   user: "admon",
...   pwd: passwordPrompt(),
...   roles: [
...     {role: "userAdminAnyDatabase", db: "admin"},
...     {role: "readWriteAnyDatabase", db: "admin"}
...   ]
... })
Enter password
TypeError: db.createUser is not a function
admin> db.createUser({
...   user: "admon",
...   pwd: passwordPrompt(),
...   roles: [
...     {role: "userAdminAnyDatabase", db: "admin"},
...     {role: "readWriteAnyDatabase", db: "admin"}
...   ]
... })
Enter password
*****{ ok: 1 }
admin>
admin>
admin>
admin>
admin>
admin>
```

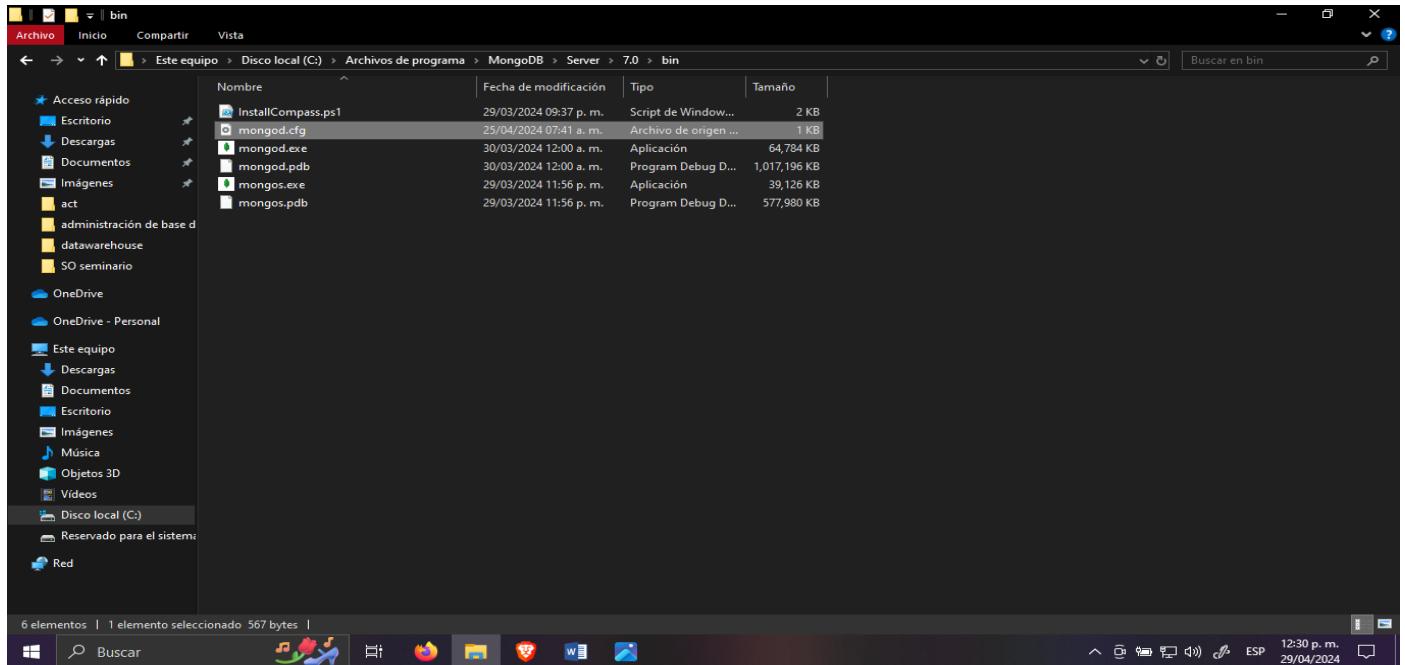
Con “show users”, para ver la información de los usuarios existentes y los permisos que estos tienen.

```
mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000
admin> db.createUser({
...   user: "admon",
...   pwd: passwordPrompt(),
...   roles: [
...     {role: "userAdminAnyDatabase", db: "admin"},
...     {role: "readWriteAnyDatabase", db: "admin"}
...   ]
... })
Enter password
*****{ ok: 1 }
admin>
admin>
admin>
admin>
admin>
admin>
admin> show users
[
  {
    _id: 'admin.admon',
    userId: UUID('94027db5-74c7-49b5-8408-857b7565d986'),
    user: 'admon',
    db: 'admin',
    roles: [
      { role: 'readWriteAnyDatabase', db: 'admin' },
      { role: 'userAdminAnyDatabase', db: 'admin' }
    ],
    mechanisms: [ 'SCRAM-SHA-1', 'SCRAM-SHA-256' ]
  }
]
admin>
```

Configurar el archivo de configuración de MongoDB para activar la autenticación.

Para activar la autenticación en el gestor MongoDB.

Nos vamos al directorio bin de mongo db, esto para ubicar el archivo mongod.cfg que será modificado.



Se tiene que abrir como administrador para que se apliquen los cambios. Aquí se debe quitar el comentario a seguridad y habilitarla.

```
# for documentation of all options, see:
#   http://docs.mongodb.org/manual/reference/configuration-options/

# Where and how to store data.
storage:
  dbPath: C:\Program Files\MongoDB\Server\7.0\data

# where to write logging data.
systemLog:
  destination: file
  logAppend: true
  path: C:\Program Files\MongoDB\Server\7.0\log\mongod.log

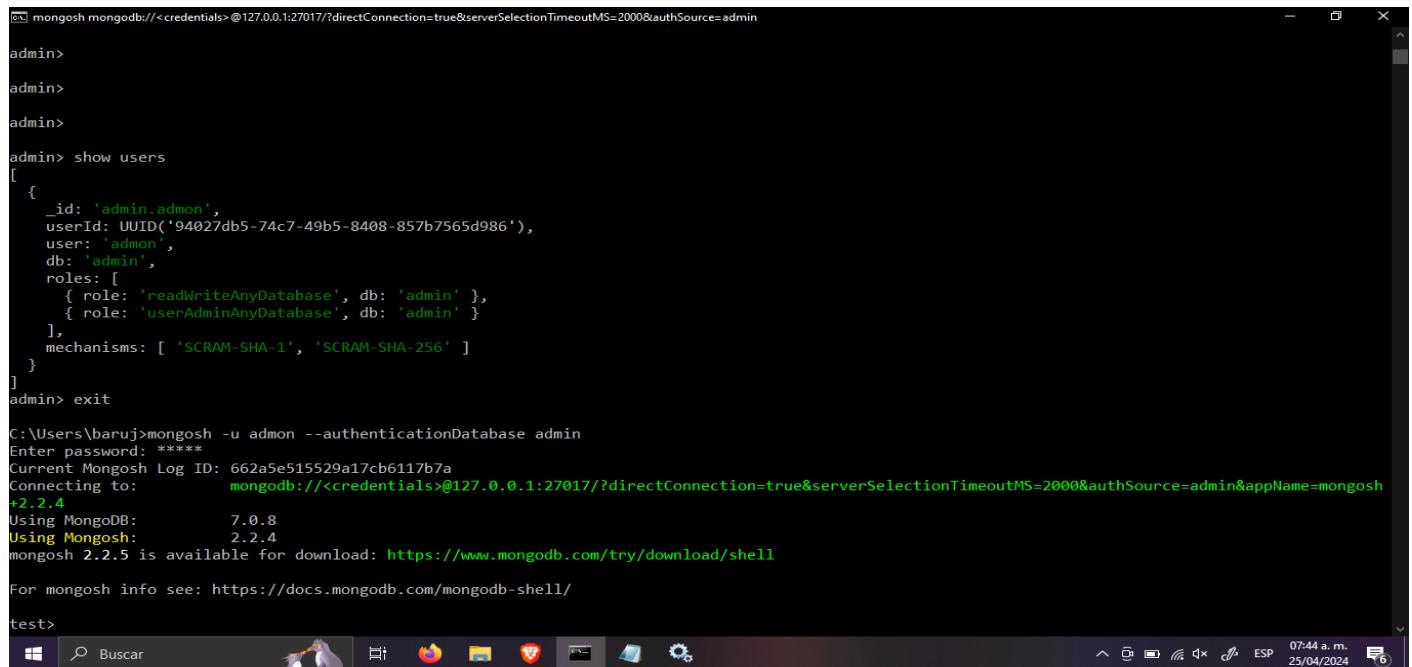
# network interfaces
net:
  port: 27017
  bindIp: 127.0.0.1

#processManagement:
security:
  authorization: enabled |

#operationProfiling:
#replication:
#sharding:
## Enterprise-Only Options:
#auditLog:
```

Con esto podemos que el inicio de sesión cambia y en este caso se tiene que acceder con el usuario y sus credenciales.

```
mongosh -u admon --authenticationDatabase admin
```



```
mongosh mongodb://<credentials>@127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000&authSource=admin
admin>
admin>
admin>
admin> show users
[
  {
    _id: 'admin.admon',
    userId: UUID('94027db5-74c7-49b5-8408-857b7565d986'),
    user: 'admon',
    db: 'admin',
    roles: [
      { role: 'readWriteAnyDatabase', db: 'admin' },
      { role: 'userAdminAnyDatabase', db: 'admin' }
    ],
    mechanisms: [ 'SCRAM-SHA-1', 'SCRAM-SHA-256' ]
  }
]
admin> exit
C:\Users\barujj>mongosh -u admon --authenticationDatabase admin
Enter password: *****
Current Mongosh Log ID: 662a5e515529a17cb6117b7a
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/mongodb-shell/
test>
```

Crear un usuario mortal con permisos de lectura y escritura en una base de datos y con permiso de sólo lectura en otra base de datos. Mostrar el funcionamiento de cada rol realizando una consulta o inserción en alguna colección.

Utilizando la conexión admin, creamos al usuario mortal, de la misma manera que se creó al usuario admin, sin embargo, los privilegios cambian, en este caso solo puede leer y escribir en la base de datos "nuevabase" y solo tiene el privilegio de lectura en la base de datos "baselectura".

Añadiendo que se utiliza el comando "show users" para verificar que si se haya creado el usuario mortal con sus permisos.

```
mongosh mongodb://<credentials>@127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000&authSource=admin

test> use admin
switched to db admin
admin> db.createUser({
...   user: "mortal_baruj",
...   pwd: passwordPrompt(),
...   roles: [
...     {role: "readWrite", db: "nuevabase"},
...     {role: "read", db: "baselectura"}
...   ]
... })
Enter password
*****{ ok: 1 }
admin> show users
[
  {
    _id: 'admin.admon',
    userId: UUID('94027db5-74c7-49b5-8408-857b7565d986'),
    user: 'admon',
    db: 'admin',
    roles: [
      { role: 'readWriteAnyDatabase', db: 'admin' },
      { role: 'userAdminAnyDatabase', db: 'admin' }
    ],
    mechanisms: [ 'SCRAM-SHA-1', 'SCRAM-SHA-256' ]
  },
  {
    _id: 'admin.mortal_baruj',
    userId: UUID('ee5abb7d-e9b1-479e-963e-06d5506b5791'),
    user: 'mortal_baruj',
    db: 'admin',
    roles: [
      { role: 'readWrite', db: 'nuevabase' },
      { role: 'read', db: 'baselectura' }
    ],
  }
]
MongoDB shell version: 5.2.2
Copyright (c) 2023, MongoDB, Inc.
All rights reserved.
Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
Details available at https://www.mongodb.com/licensing/oss-2.0.
Server version: 5.2.2
Connection id: 1
Current database: admin

```

Para empezar, se crean las colecciones en las bases de datos establecidas para el usuario mortal, la primera colección será frutas en “nuevabase” donde el usuario puede leer y escribir.

```
use nuevabase
```

```
db.frutas.insertMany([
  { nombre: "manzana", tipo: "fruta" },
  { nombre: "uva", tipo: "fruta" },
  { nombre: "pera", tipo: "fruta" }
])
```

Y se visualizan los datos de la colección con find()

```
mongosh mongodb://<credentials>@127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000&authSource=admin
test>
nuevabase>
switched to db nuevabase
nuevabase>
nuevabase> db.frutas.insertMany([
...   { nombre: "manzana", tipo: "fruta" },
...   { nombre: "uva", tipo: "fruta" },
...   { nombre: "pera", tipo: "fruta" }
... ])
{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId('66306e2f55621e1873117b7b'),
    '1': ObjectId('66306e2f55621e1873117b7c'),
    '2': ObjectId('66306e2f55621e1873117b7d')
  }
}
nuevabase> db.frutas.find()
[ {
  _id: ObjectId('66306e2f55621e1873117b7b'),
  nombre: 'manzana',
  tipo: 'fruta'
},
{
  _id: ObjectId('66306e2f55621e1873117b7c'),
  nombre: 'uva',
  tipo: 'fruta'
},
{
  _id: ObjectId('66306e2f55621e1873117b7d'),
  nombre: 'pera',
  tipo: 'fruta'
} ]
```

La siguiente colección será juegos y está en “baselectura” donde el usuario solo puede realizar la lectura.

use baselectura

```
db.juegos.insertMany([
  { nombre: "super mario", tipo: "juegos" },
  { nombre: "mother", tipo: "juegos" },
  { nombre: "zelda", tipo: "juegos" }
])
```

```
mongosh mongodb://<credentials>@127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000&authSource=admin
baselectura>
switched to db baselectura
baselectura>
baselectura>
...   { nombre: "super mario", tipo: "juegos" },
...   { nombre: "mother", tipo: "juegos" },
...   { nombre: "zelda", tipo: "juegos" }
...
...
...
...
...
...
...
...
...
...
baselectura>
baselectura> db.juegos.insertMany([
...   { nombre: "super mario", tipo: "juegos" },
...   { nombre: "mother", tipo: "juegos" },
...   { nombre: "zelda", tipo: "juegos" }
... ])
{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId('6630703955621e1873117b7e'),
    '1': ObjectId('6630703955621e1873117b7f'),
    '2': ObjectId('6630703955621e1873117b80')
  }
}
baselectura>
```

Y se visualizan los datos de la colección con find()

```
mongosh mongodb://<credentials>@127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000&authSource=admin
...
baselectura>
baselectura> db.juegos.insertMany([
...   { nombre: "super mario", tipo: "juegos" },
...   { nombre: "mother", tipo: "juegos" },
...   { nombre: "zelda", tipo: "juegos" }
... ])
{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId('6630703955621e1873117b7e'),
    '1': ObjectId('6630703955621e1873117b7f'),
    '2': ObjectId('6630703955621e1873117b80')
  }
}
baselectura> db.juegos.find()
[
  {
    _id: ObjectId('6630703955621e1873117b7e'),
    nombre: 'super mario',
    tipo: 'juegos'
  },
  {
    _id: ObjectId('6630703955621e1873117b7f'),
    nombre: 'mother',
    tipo: 'juegos'
  },
  {
    _id: ObjectId('6630703955621e1873117b80'),
    nombre: 'zelda',
    tipo: 'juegos'
  }
]
baselectura>
```

Ya con las bases de datos creadas y sus respectivas colecciones, entramos con el usuario mortal con el siguiente comando:

```
mongosh -u mortal_baruj --authenticationDatabase admin
```

cambiamos a la base de datos “nuevabase” y hacemos lectura de una colección con find().

```
mongosh mongodb://<credentials>@127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000&authSource=admin
]
admin> exit

C:\Users\baruj>mongosh -u mortal_baruj --authenticationDatabase admin
Enter password: *****
Current Mongosh Log ID: 66310223b5ada338f3117b7a
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/mongodb-shell/

nuevabase>
switched to db nuevabase
nuevabase> db.frutas.find()
[
  {
    _id: ObjectId('66306e2f55621e1873117b7b'),
    nombre: 'manzana',
    tipo: 'fruta'
  },
  {
    _id: ObjectId('66306e2f55621e1873117b7c'),
    nombre: 'uva',
    tipo: 'fruta'
  },
  {
    _id: ObjectId('66306e2f55621e1873117b7d'),
    nombre: 'pera',
    tipo: 'fruta'
  }
]
nuevabase>
```

Se añade un nuevo elemento para la colección con insertOne().

```
mongosh mongodb://<credentials>@127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000&authSource=admin
]
nuevabase> db.frutas.insertOne({ nombre: "naranja", tipo: "frutas" })
```

Si volvemos a realizar lectura de los elementos con find() podemos observar que al final se agregó el nuevo elemento.

```
mongosh mongodb://<credentials>@127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000&authSource=admin
}
]
nuevabase> db.frutas.find()
[
  {
    acknowledged: true,
    _id: ObjectId('66306e2f55621e1873117b7b'),
    nombre: 'manzana',
    tipo: 'fruta'
  },
  {
    _id: ObjectId('66306e2f55621e1873117b7c'),
    nombre: 'uva',
    tipo: 'fruta'
  },
  {
    _id: ObjectId('66306e2f55621e1873117b7d'),
    nombre: 'pera',
    tipo: 'fruta'
  },
  {
    _id: ObjectId('663103bab5ada338f3117b7b'),
    nombre: 'naranja',
    tipo: 'frutas'
  }
]
nuevabase>
```

Además, podemos eliminar un nuevo elemento con deleteOne()

```
mongosh mongodb://<credentials>@127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000&authSource=admin
  nombre: 'uva',
  tipo: 'fruta'
},
{
  _id: ObjectId('66306e2f55621e1873117b7d'),
  nombre: 'pera',
  tipo: 'fruta'
},
{
  _id: ObjectId('663103bab5ada338f3117b7b'),
  nombre: 'naranja',
  tipo: 'frutas'
}
]
nuevabase>
nuevabase> db.frutas.deleteOne({nombre: "pera"})
```

Si hacemos lectura con find() podemos observar que el elemento fue eliminado.

```
mongosh mongodb://<credentials>@127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000&authSource=admin
{
  _id: ObjectId('663103bab5ada338f3117b7b'),
  nombre: 'naranja',
  tipo: 'frutas'
}
]
nuevabase>
nuevabase> db.frutas.find()
[
  {
    _id: ObjectId('66306e2f55621e1873117b7b'),
    nombre: 'manzana',
    tipo: 'fruta'
  },
  {
    _id: ObjectId('66306e2f55621e1873117b7c'),
    nombre: 'uva',
    tipo: 'fruta'
  },
  {
    _id: ObjectId('663103bab5ada338f3117b7b'),
    nombre: 'naranja',
    tipo: 'frutas'
  }
]
nuevabase>
```



8:47 a.m.
30/04/2024

Si nos conectamos a la base de datos “baselectura” podemos hacer una lectura con find() de la colección juegos, y observamos los elementos insertados.

```
mongosh mongodb://<credentials>@127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000&authSource=admin
{
  _id: ObjectId('663103bab5ada338f3117b7b'),
  nombre: 'naranja',
  tipo: 'frutas'
}
]
nuevabase> use baselectura
baselectura> db.juegos.find()
[
  {
    _id: ObjectId('6630703955621e1873117b7e'),
    nombre: 'super mario',
    tipo: 'juegos'
  },
  {
    _id: ObjectId('6630703955621e1873117b7f'),
    nombre: 'mother',
    tipo: 'juegos'
  },
  {
    _id: ObjectId('6630703955621e1873117b80'),
    nombre: 'zelda',
    tipo: 'juegos'
  }
]
baselectura>
```



8:48 a.m.
30/04/2024

Sin embargo, no se puede insertar ningún nuevo elemento con insertOne()

```
mongosh mongodb://<credentials>@127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000&authSource=admin
    tipo: 'juegos'
},
{
  _id: ObjectId('6630703955621e1873117b7f'),
  nombre: 'mother',
  tipo: 'juegos'
},
{
  _id: ObjectId('6630703955621e1873117b80'),
  nombre: 'zelda',
  tipo: 'juegos'
}
]
baselectura> db.juegos.insertOne({ nombre: "metroid", tipo: "juegos" })
MongoServerError[Unauthorized]: not authorized on baselectura to execute command { insert: "juegos", documents: [ { nombre: "metroid", tipo: "juegos" }, _id: ObjectId('663104dcb5ada338f3117b7c') ] }, ordered: true, lsid: { id: UUID("68fdb0cf-425b-4458-95a5-07a27b491f94") }, $db: "baselectura" }
```

Tampoco se podrá eliminar nada con deleteOne()

```
mongosh mongodb://<credentials>@127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000&authSource=admin
baselectura> db.juegos.in
baselectura.juegos.inhlorized]: not authorized on baselectura to execute command { insert: "juegos", documents: [ { nombre: "metroid", tipo: "juegos" } ]
baselectura>
baselectura> db.juegos.deleteOne({ nombre: "mother" })
MongoServerError[Unauthorized]: not authorized on baselectura to execute command { delete: "juegos", deletes: [ { q: { nombre: "mother" }, limit: 1 } ], ordered: true, lsid: { id: UUID("68fdb0cf-425b-4458-95a5-07a27b491f94") }, $db: "baselectura" }
```

Por lo que se puede apreciar de que los permisos otorgados cumplieron la función de otorgar mayor seguridad a la base de datos.