

2024



# Big Data Aplicado

Profesor/a: José Manuel González Rodríguez

ACTIVIDAD EVALUABLE 1.4 (MONGODB)  
28/11/2024

## Índice

<b>1.- Introducción</b>	<b>4</b>
1.1.- Objetivo del Trabajo	4
1.2.- Herramientas Necesarias	4
<b>2.- Preparación del Entorno</b>	<b>5</b>
2.1.- Instalación de MongoDB	5
2.1.1.- Descargar MongoDB	5
2.1.2.- Descargar MongoDB Compass (GUI)	10
2.1.3.- Descargar Mongosh	12
2.1.4.- Descargar Tools	15
<b>3.- Ejercicios</b>	<b>21</b>
<b>Ejercicio 1: Creación de la Base de Datos citibike.</b>	<b>21</b>
Paso 1: Abrir la shell de MongoDB	21
Paso 2: Cambiar al contexto de la base de datos	21
Paso 3: Verificar la creación de la base de datos	22
<b>Ejercicio 2: Crear una Colección con las Iniciales del Nombre y Apellidos.</b>	<b>22</b>
Paso 1: Moverse a la base de datos citibike	22
Paso 2: Crear la Colección	23
Paso 3: Verificar la creación de la colección	23
<b>Ejercicio 3: Importación del Archivo citibike.csv.</b>	<b>23</b>
Paso 1: Ubicar el archivo CSV	23
Paso 2: Ejecutar el comando mongoimport	24
<b>Ejercicio 4: Mostrar el Número de Viajes en la Colección.</b>	<b>24</b>
<b>Ejercicio 5: Inserción de un Nuevo Viaje.</b>	<b>24</b>
Comando para Insertar el Viaje	25
<b>Ejercicio 6: Obtener los 5 Primeros Viajes para la Bicicleta con Identificador 24556.</b>	<b>26</b>
<b>Ejercicio 7: Obtener los Viajes cuya Estación de Salida o Llegada sea City Hall.</b>	<b>27</b>
<b>Ejercicio 8: Obtener los Viajes Ordenados por Duración.</b>	<b>28</b>
<b>Ejercicio 9: Actualización de la Denominación de la Estación con Identificador 3203.</b>	<b>29</b>
<b>Ejercicio 10: Determinar quién ha hecho más viajes: Usuarios Masculinos o Femeninos.</b>	<b>30</b>
<b>Ejercicio 11: Mostrar los Viajes de los Usuarios Nacidos en la Década de los 80.</b>	<b>31</b>
<b>Ejercicio 12: Crear una Nueva Colección "genderless" con Viajes sin Género Conocido.</b>	<b>31</b>
<b>Ejercicio 13: Eliminar Viajes con Género Desconocido y Mostrar el Número de Viajes Restantes.</b>	<b>32</b>
<b>Ejercicio 14: Calcular la Duración Promedio de los Viajes.</b>	<b>32</b>
<b>Ejercicio 15: Contar el Número de Viajes por Tipo de Usuario.</b>	<b>33</b>
<b>Ejercicio 16: Mostrar el Nombre de la Estación más Popular como Estación de Salida.</b>	<b>33</b>
<b>Ejercicio 17: Obtener los Identificadores de las 4 Bicicletas Más Usadas.</b>	<b>34</b>
<b>Ejercicio 18: Obtener el Número de Viajes Realizados el 30 de septiembre de 2015.</b>	<b>35</b>

<b>4.- Exportación de la base de datos a formato csv.</b>	<b>35</b>
4.1.- Explotación de la base de datos en un archivo PMGA_viajes.csv.	35
4.2.- Exportación de la base de datos en un archivo genderless_viajes.csv.	36
<b>5.- Conclusiones</b>	<b>36</b>
5.1.- Análisis del Proceso y Resultados	36
5.2.- Limitaciones y Recomendaciones.	37
5.2.1.- Recomendaciones	38
5.2.2.- Resumen Final	38
<b>6.- Bibliografía</b>	<b>39</b>
6.1.- Libros y Textos Académicos	39
6.2.- Documentación Técnica	39
6.3.- Artículos y Recursos en Línea	39
6.4.- Sitios Web y Blogs Técnicos	40
<b>7.- Mapa mental Actividad Evaluable 1.4 (MongoDB)</b>	<b>41</b>
<b>8.- Anexos</b>	<b>41</b>
8.1.- PMGA_viajes.csv	41
8.2.- genderless_viajes.csv	42
8.3.- Actividad Evaluable 1.4 (MongoDB).pdf	42
8.4.- Mapa Mental Actividad Evaluable 1.4 (MongoDB).pdf	42

## 1.- Introducción

### 1.1.- Objetivo del Trabajo

El objetivo principal de este trabajo es analizar, gestionar y extraer información relevante a partir de un conjunto de datos del servicio de alquiler de bicicletas **Citibike** de la ciudad de Nueva York, utilizando la base de datos NoSQL **MongoDB**. A través de diferentes ejercicios, se busca realizar tareas de creación, manipulación, consulta y actualización de datos en una base de datos estructurada para manejar grandes volúmenes de información. Este trabajo permitirá desarrollar habilidades en el uso de MongoDB para trabajar con datos reales, identificar patrones, calcular estadísticas, y realizar análisis de datos que puedan ser aplicables en entornos de **Big Data**.

En particular, se persiguen los siguientes objetivos específicos:

- Crear una base de datos y una colección para almacenar los datos de los viajes.
- Importar datos desde un archivo CSV utilizando herramientas de MongoDB.
- Ejecutar consultas y comandos específicos para analizar los datos.
- Actualizar y gestionar datos, demostrando la flexibilidad y capacidad de las bases de datos NoSQL.
- Interpretar los resultados obtenidos y utilizarlos para tomar decisiones basadas en datos.

### 1.2.- Herramientas Necesarias

Para completar las actividades planteadas, se requiere el uso de las siguientes herramientas y recursos:

- **MongoDB Community Server:** Instalación de la versión comunitaria de MongoDB como servidor local para gestionar la base de datos.
- **MongoDB Shell:** Herramienta de línea de comandos para ejecutar comandos e interactuar con las bases de datos.
- **MongoDB Tools:** Conjunto de utilidades para importar y exportar datos, incluyendo el comando `mongoimport`, necesario para cargar el dataset en formato CSV.

- **Consola del sistema operativo:** Se utilizará la consola (cmd.exe en Windows o terminal en otros sistemas) para ejecutar comandos de MongoDB y mostrar los resultados.
- **Archivo CSV del Dataset de Citibike:** Contiene información sobre los viajes realizados, incluyendo detalles como la duración del viaje, estaciones de origen y destino, identificadores de bicicletas, tipos de usuarios, género y año de nacimiento.

Estas herramientas proporcionan un entorno completo para desarrollar las habilidades requeridas en el manejo de bases de datos NoSQL y trabajar con grandes volúmenes de datos estructurados de forma eficiente.

## 2.- Preparación del Entorno

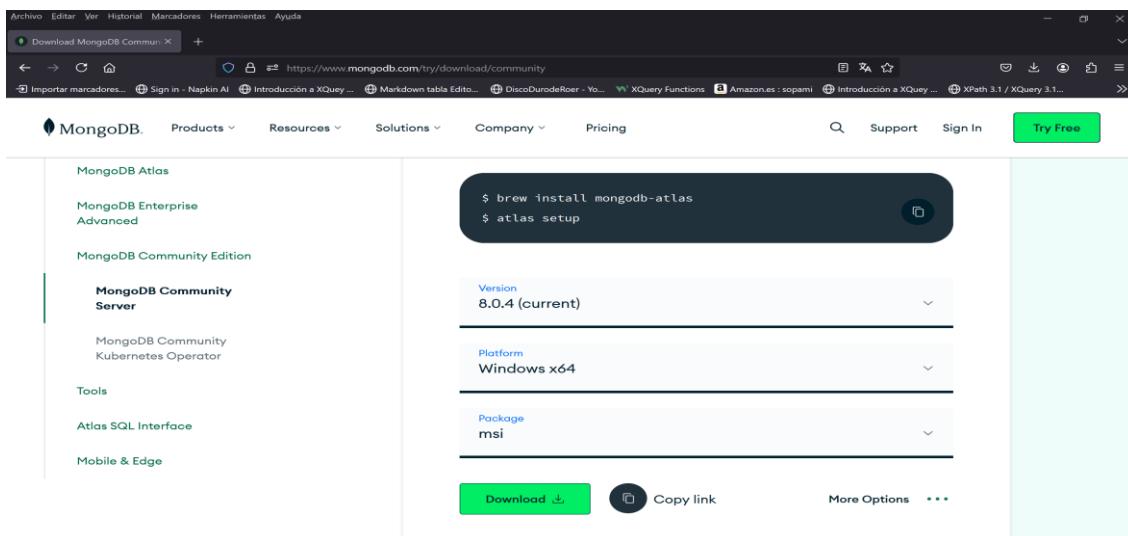
Para realizar las actividades propuestas, es fundamental preparar el entorno de trabajo instalando y configurando las herramientas necesarias. A continuación, se describe el proceso para la instalación de MongoDB y los pasos iniciales para configurarlo correctamente.

### 2.1.- Instalación de MongoDB

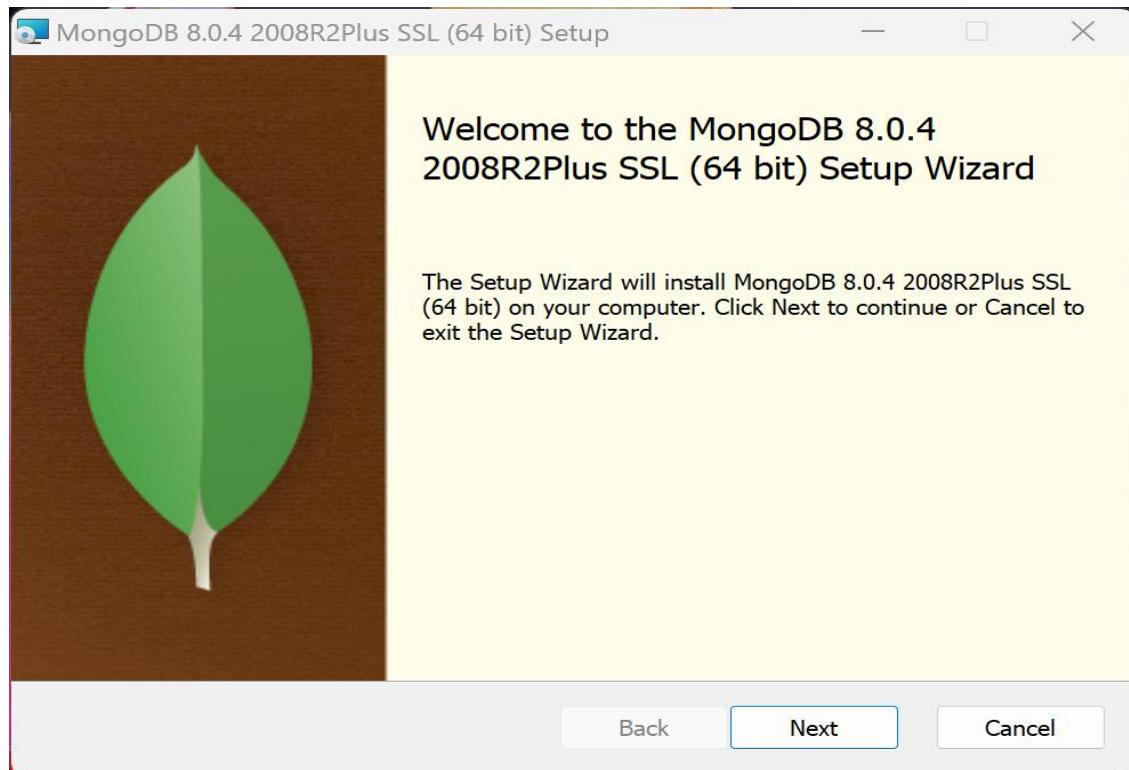
La instalación de MongoDB se realiza en su versión comunitaria, la cual está diseñada para un uso general y es completamente gratuita. El proceso varía ligeramente dependiendo del sistema operativo, pero a continuación se proporciona una guía general:

#### 2.1.1.- Descargar MongoDB

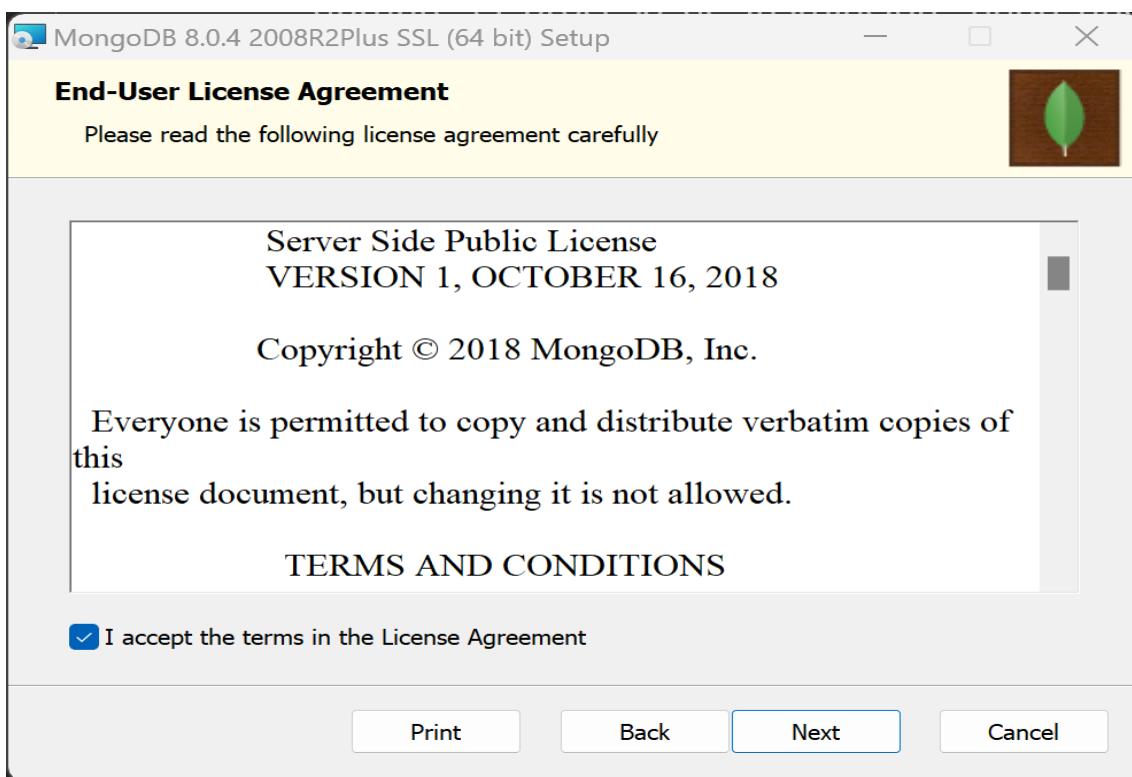
Accede a la página oficial de MongoDB: <https://www.mongodb.com/try/download/community>. Selecciona tu sistema operativo (Windows, macOS o Linux) y la versión más reciente del servidor comunitario.



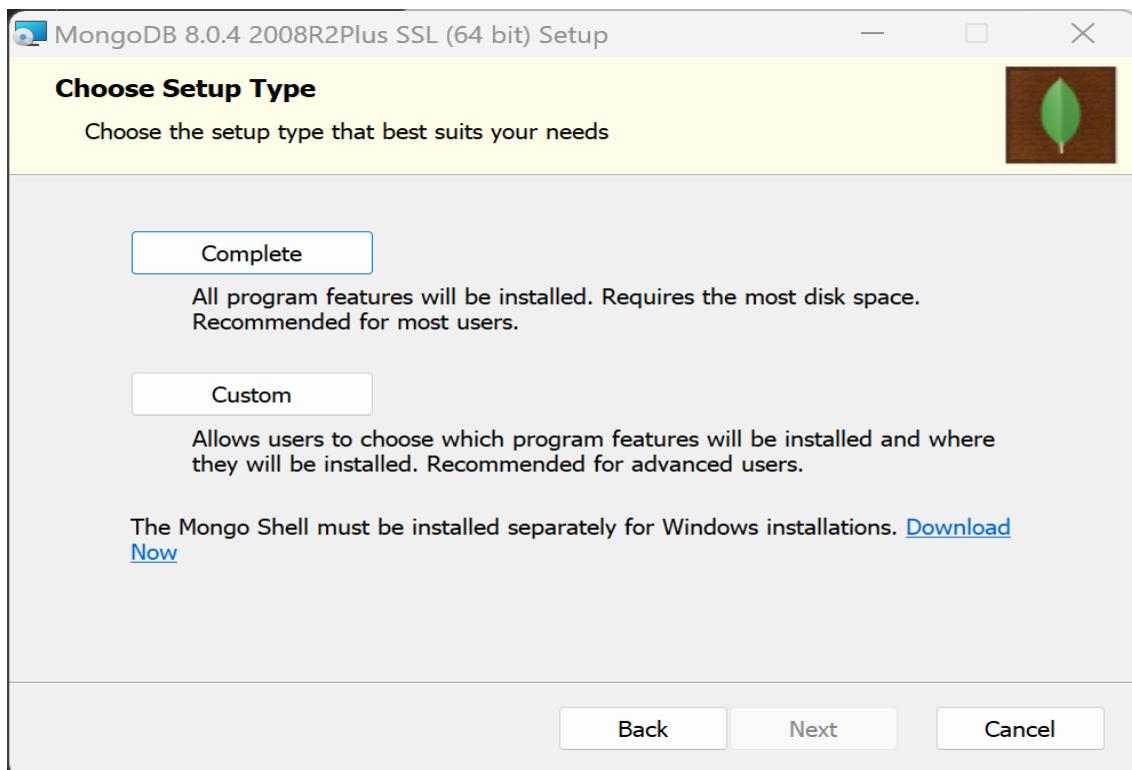
En esta interfaz, optamos por la versión 8.0.4 de MongoDB para Windows x64 en formato MSI, iniciamos su descarga mediante el botón Download y nos cercioramos de incluir MongoDB Shell (mongosh) durante el proceso de instalación, dado que es fundamental para la ejecución de comandos interactivos en el sistema gestor de bases de datos.



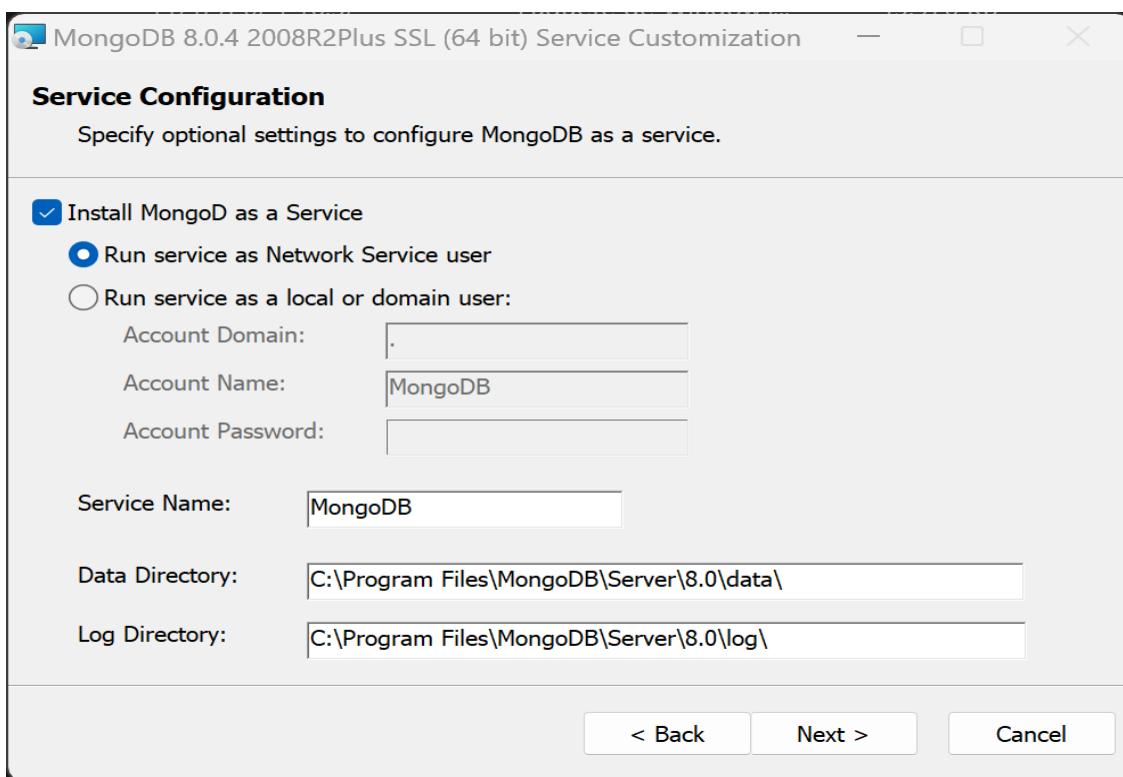
Haga doble clic en el archivo `mongodb-windows-x86_64-8.0.4-signed.msi` para iniciar la instalación. Luego, presione "Next" para continua.



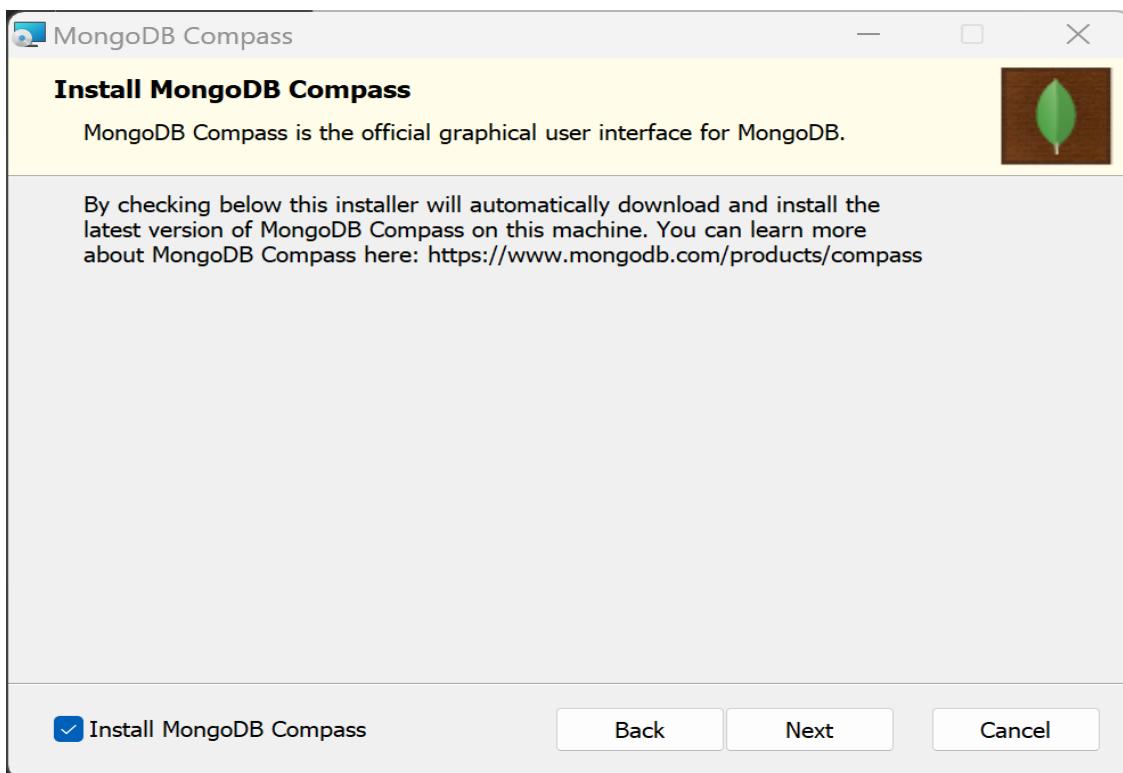
En esta pantalla, procedemos a marcar la opción de aceptación de la licencia y pulsamos el botón "Next" para continuar.



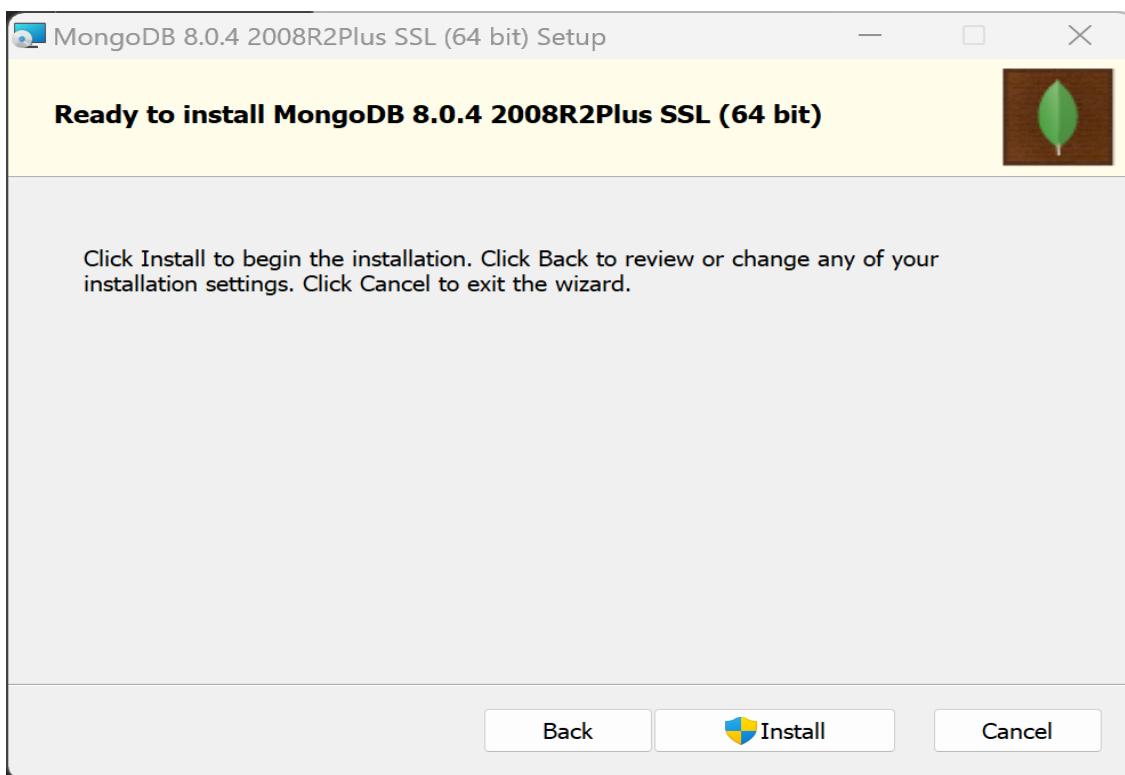
En esta pantalla, se debe hacer clic en el botón "Complete", tras lo cual no aparecerá otra ventana.



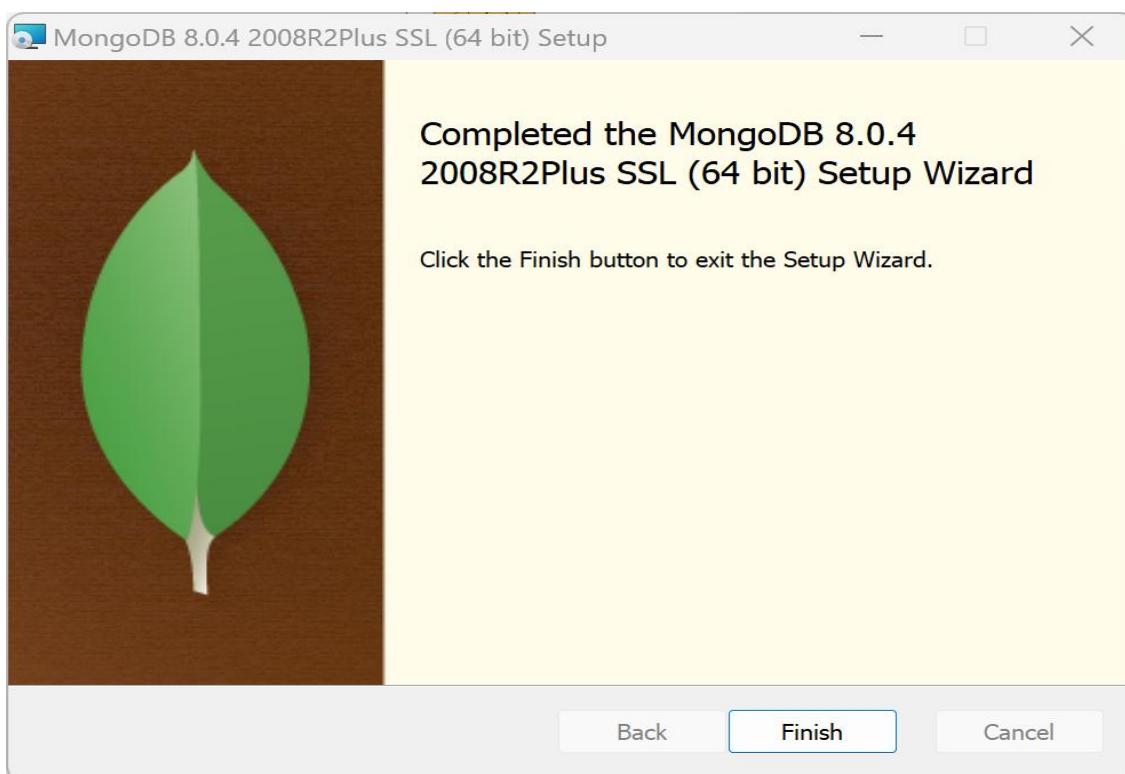
En esta pantalla, se procede a seleccionar la opción "Run service as Network Service user", manteniendo la configuración predeterminada, y pulsamos el botón "Next" para avanzar en el proceso de instalación.



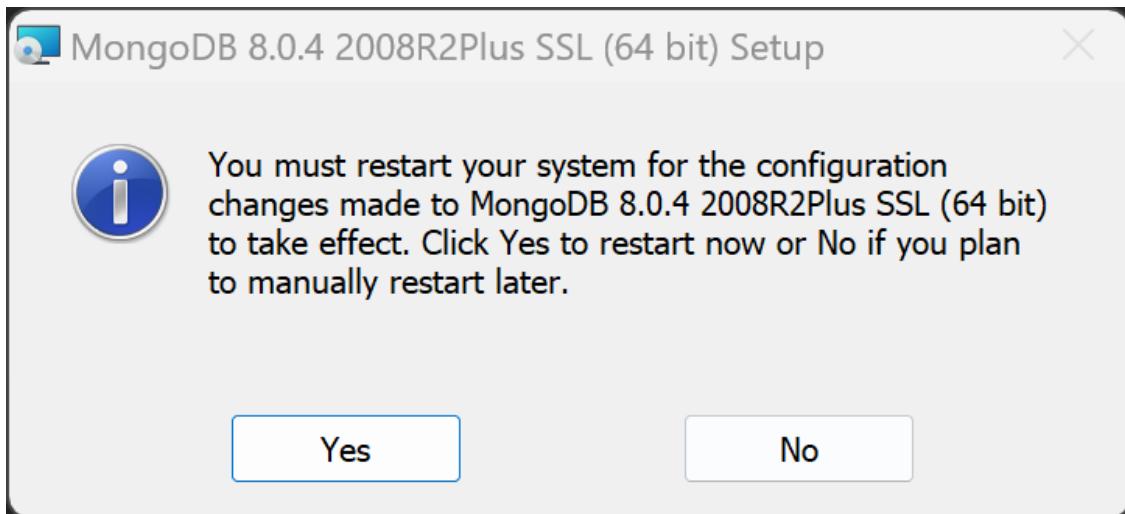
En esta pantalla, se avanza con la configuración predeterminada y se hace clic en el botón "Next" para continuar con el proceso de instalación.



En esta pantalla se procede a ejecutar la instalación pulsando la tecla install para ejecutar el proceso de instalación.



En esta pantalla, se concluye el proceso de instalación al hacer clic en el botón "Finish".



En esta pantalla, se procede a reiniciar el ordenador para que la instalación surta efecto, y se hace clic en el botón "Yes".

### 2.1.2.- Descargar MongoDB Compass (GUI)

Accede a la página oficial de MongoDB:  
<https://www.mongodb.com/try/download/compass>.

The screenshot shows the MongoDB official website at <https://www.mongodb.com/try/download/compass>. The page has a dark header with links for "Archivo", "Editar", "Ver", "Historial", "Marcadores", "Herramientas", and "Ayuda". Below the header, there's a navigation bar with "MongoDB.", "Products", "Resources", "Solutions", "Company", "Pricing", "Q", "Support", "Sign In", and a green "Try Free" button. On the left, a sidebar lists products: "MongoDB Atlas", "MongoDB Enterprise Advanced", "MongoDB Community Edition", "Tools", "MongoDB Shell", "MongoDB Compass (GUI)" (which is highlighted in green), "Atlas CLI", "Atlas Kubernetes Operator", "MongoDB CLI for Cloud Manager and Ops Manager", and "MongoDB Cluster-to-Cluster Sync". The main content area for "MongoDB Compass (GUI)" includes a description: "This version disables all network connections except the connection to the MongoDB instance.", a "Learn more" link, dropdown menus for "Version" (set to "1.45.0 (Stable)"), "Platform" (set to "Windows 64-bit (10+)"), and "Package" (set to "exe"), and a "Download" button. To the right, there's a large "Try Free" button.

En esta pantalla procedemos a descargar MongoDB Compass es una interfaz gráfica de usuario (GUI) para MongoDB que permite explorar y manipular bases de datos de forma intuitiva, ofreciendo

visualización de esquemas, métricas de rendimiento en tiempo real y capacidades de consulta avanzadas, disponible en tres versiones: completa, solo lectura y aislada.



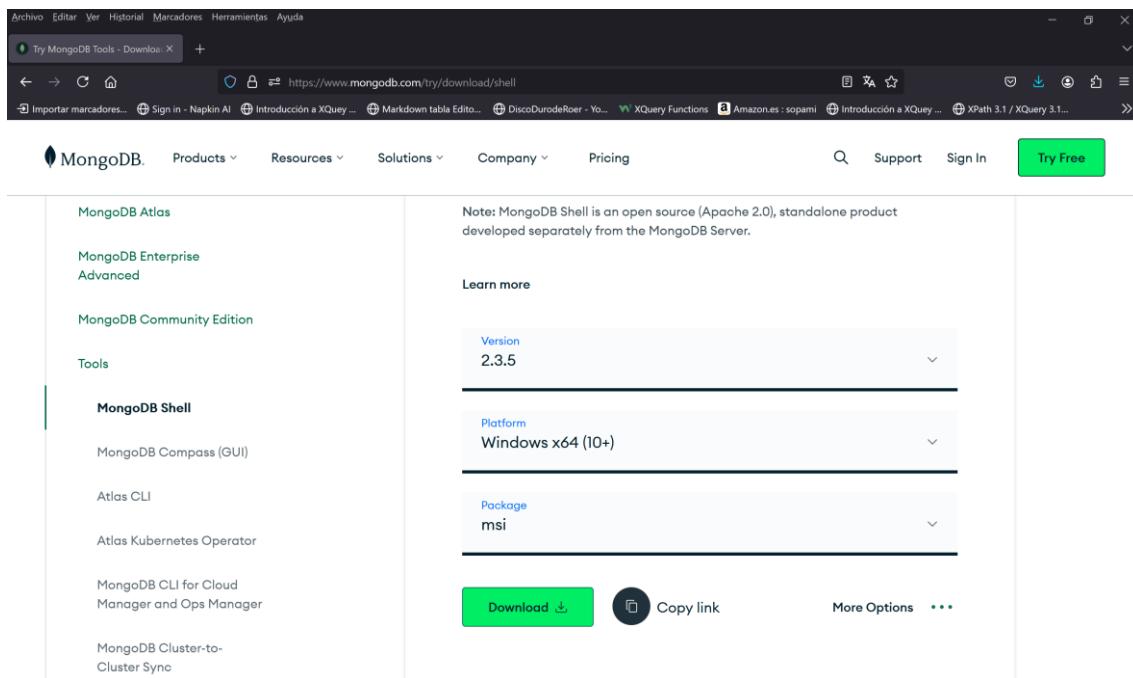
MongoDB Compass is being installed.

It will launch once it is done.

Iniciamos la ejecución del archivo ejecutable `mongodb-compass-1.45.0-win32-x64.exe` para proceder con la instalación de la interfaz gráfica de usuario de MongoDB2. Esta herramienta nos permitirá gestionar y visualizar nuestras bases de datos de manera intuitiva.

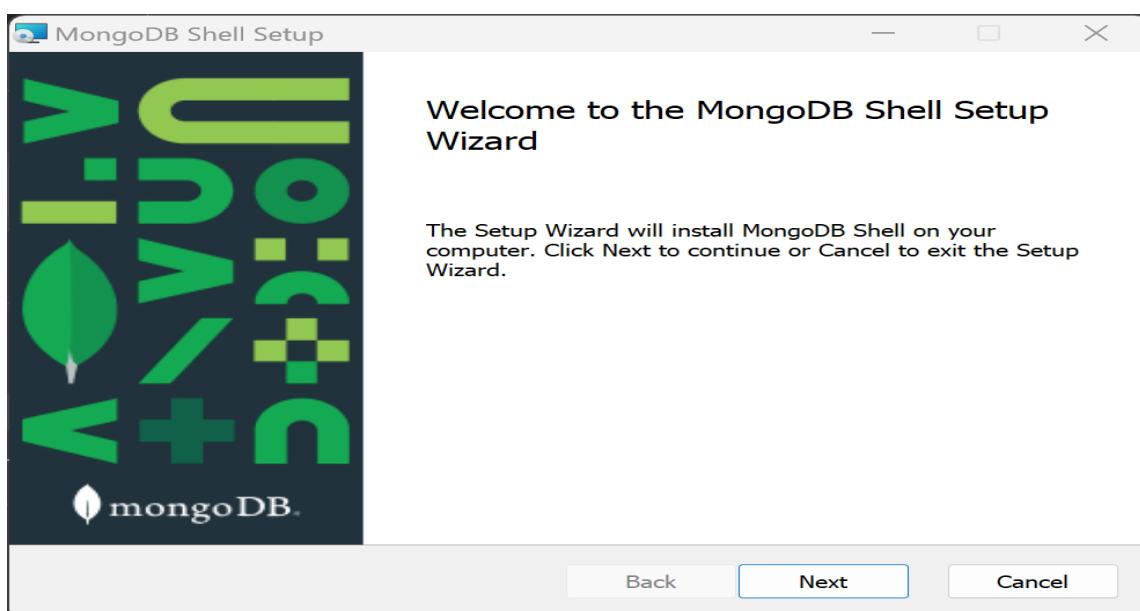
### 2.1.3.- Descargar Mongosh

Accede a la página oficial de MongoDB:  
<https://www.mongodb.com/try/download/shell>.

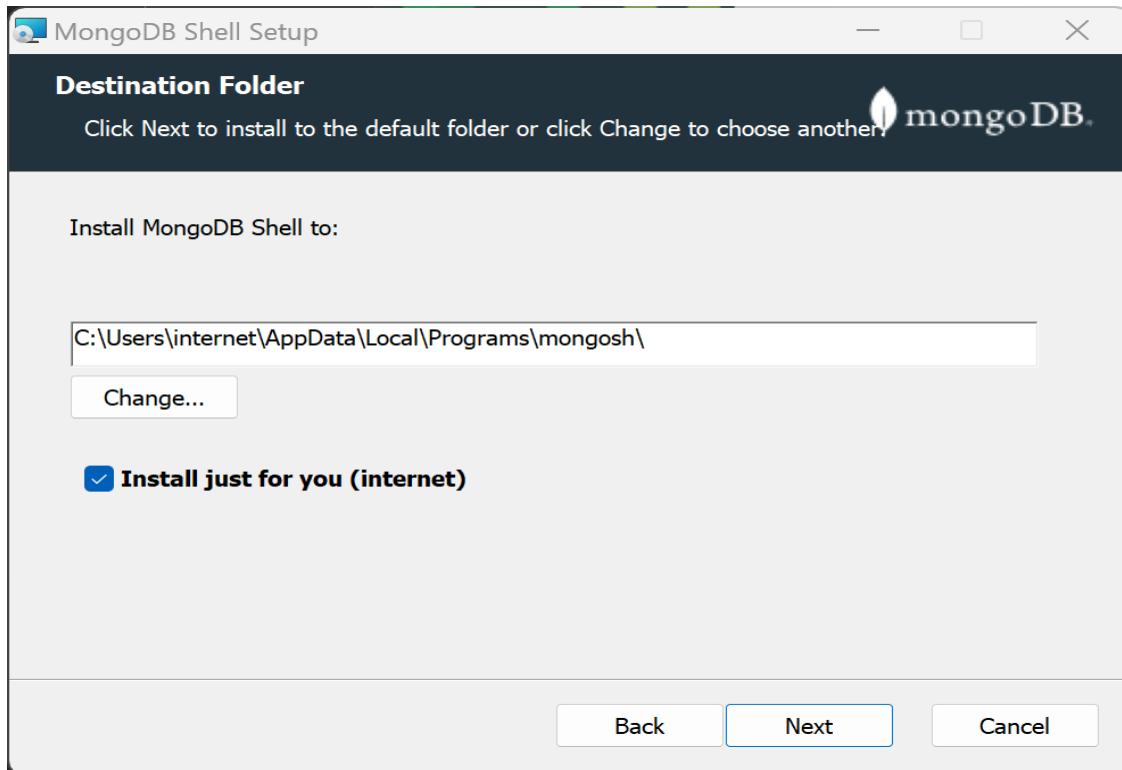


The screenshot shows the MongoDB website's download section for the MongoDB Shell. On the left, there's a sidebar with links to MongoDB Atlas, Enterprise Advanced, Community Edition, and various tools like Compass, CLI, and Kubernetes Operator. The main content area has a note about the MongoDB Shell being an open-source product developed separately from the MongoDB Server. It shows dropdown menus for selecting Version (2.3.5), Platform (Windows x64 (10+)), and Package (msi). At the bottom, there are 'Download' and 'Copy link' buttons, along with 'More Options' and a three-dot menu.

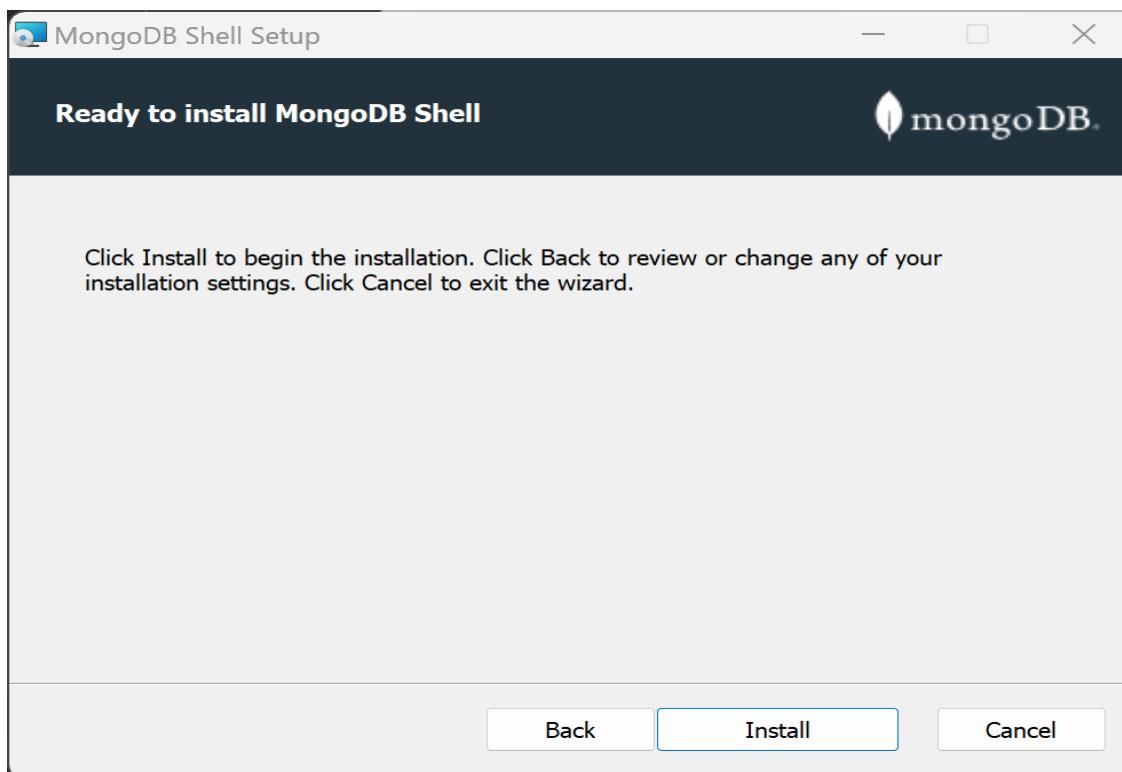
En esta pantalla se procede a descargar MongoDB Shell, es una interfaz de línea de comandos moderna y de código abierto (Apache 2.0) que permite conectarse rápidamente a MongoDB, consultar datos, configurar ajustes y ejecutar acciones con funcionalidades avanzadas como resaltado de sintaxis y autocompletado inteligente.



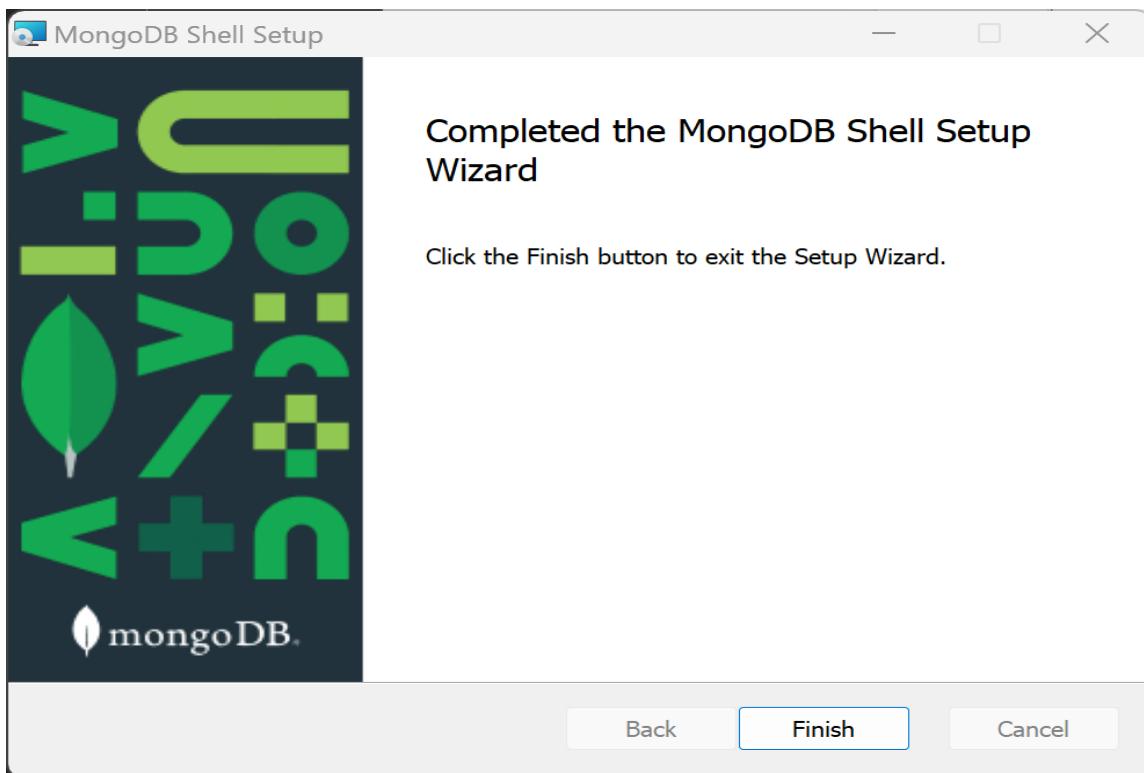
Procedemos a instalar el archivo mongosh-2.3.5-x64.msi para integrar las herramientas de MongoDB, asegurándonos de seguir las indicaciones del asistente de instalación.



En esta pantalla, mantenemos la configuración predeterminada y hacemos clic en el botón "Next".



En esta pantalla, procedemos a la instalación del MongoDB Shell haciendo clic en el botón "Install" para comenzar el proceso de instalación.



En esta pantalla, finalizamos el proceso de instalación haciendo clic en el botón "Finish".

The screenshot shows a terminal window titled "mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000&appName=mongosh+2.3.5". The output of the command is displayed, including connection details and startup warnings. The warnings indicate that access control is not enabled for the database and that read and write access to data and configuration is unrestricted. The prompt "test>" is visible at the bottom of the terminal.

```
C:\Users\internet>mongosh
Current Mongosh Log ID: 675c7cd845811aff2f0deb2f
Connecting to: mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000&appName=mongosh+2.3.5
Using MongoDB: 8.0.4
Using Mongosh: 2.3.5

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

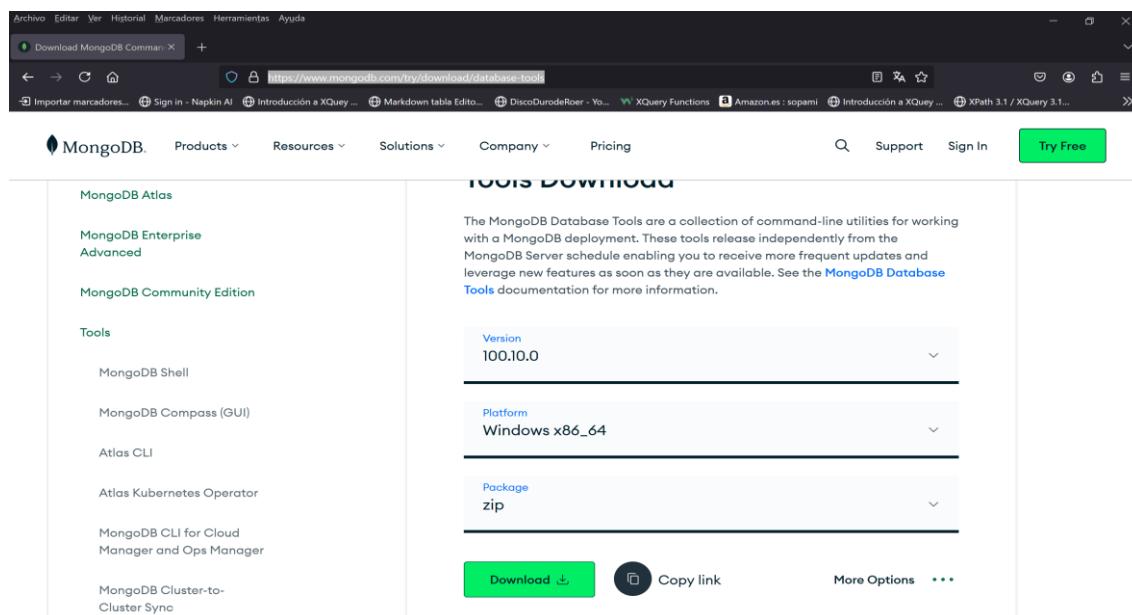
-----
The server generated these startup warnings when booting
2024-12-13T19:07:31.822+01:00: Access control is not enabled for the database
e. Read and write access to data and configuration is unrestricted
-----

test>
```

Para verificar la instalación de MongoDB, abre una terminal y ejecuta `mongosh` para acceder a la shell interactiva de MongoDB, lo que confirma que la instalación se ha realizado correctamente y que MongoDB está listo para su uso.

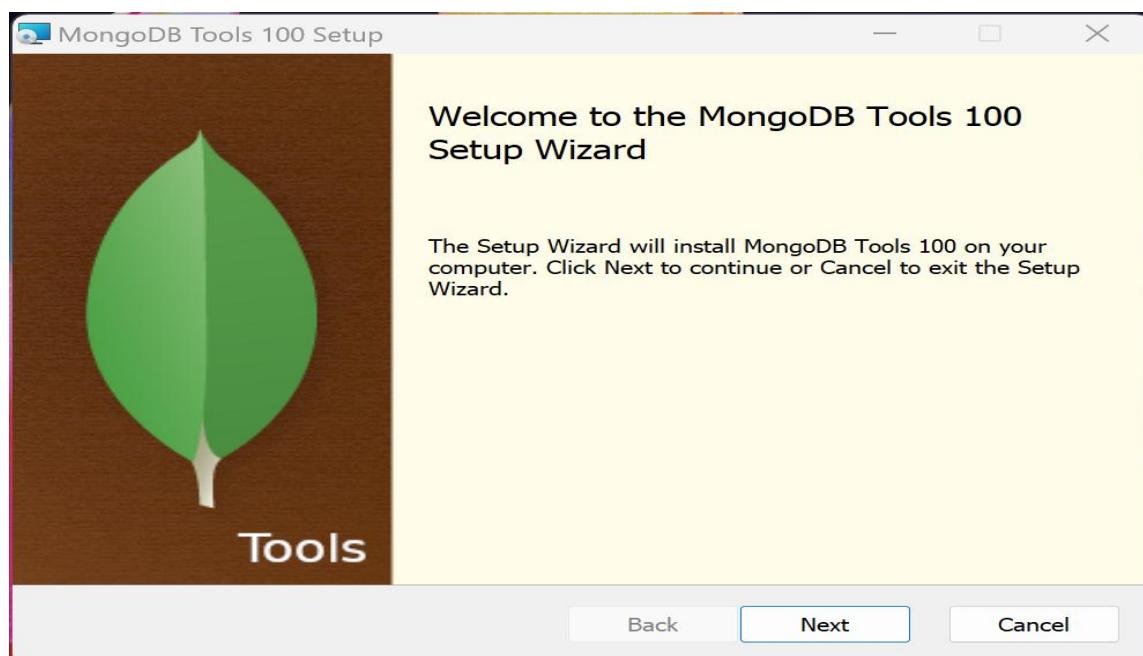
## 2.1.4.- Descargar Tools

Accede a la página oficial de MongoDB:  
<https://www.mongodb.com/try/download/database-tools>.

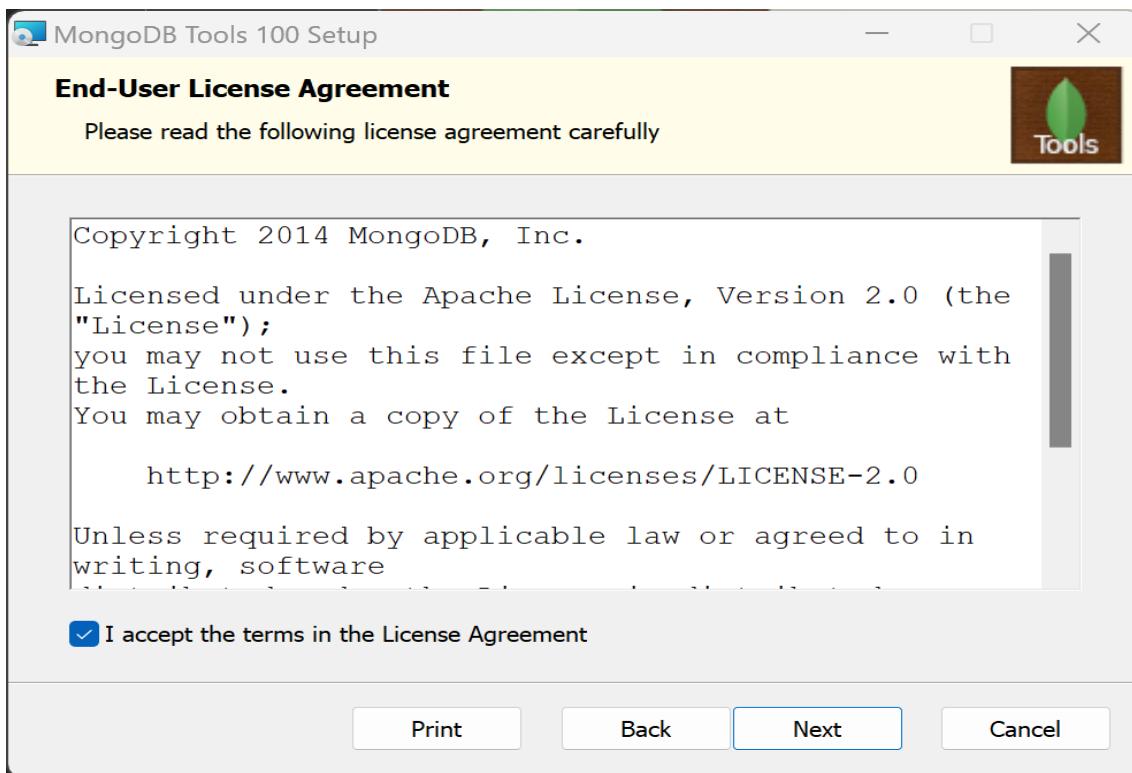


The screenshot shows the MongoDB website's 'Tools Download' page. On the left, there's a sidebar with links to MongoDB Atlas, Enterprise Advanced, Community Edition, and various tools like Shell, Compass, CLI, Kubernetes Operator, Cloud Manager, and Cluster Sync. The main content area is titled 'TOOLS DOWNLOAD' and contains three dropdown menus: 'Version' set to '100.10.0', 'Platform' set to 'Windows x86\_64', and 'Package' set to 'zip'. At the bottom, there are buttons for 'Download' (with a link icon), 'Copy link', 'More Options', and a 'Try Free' button.

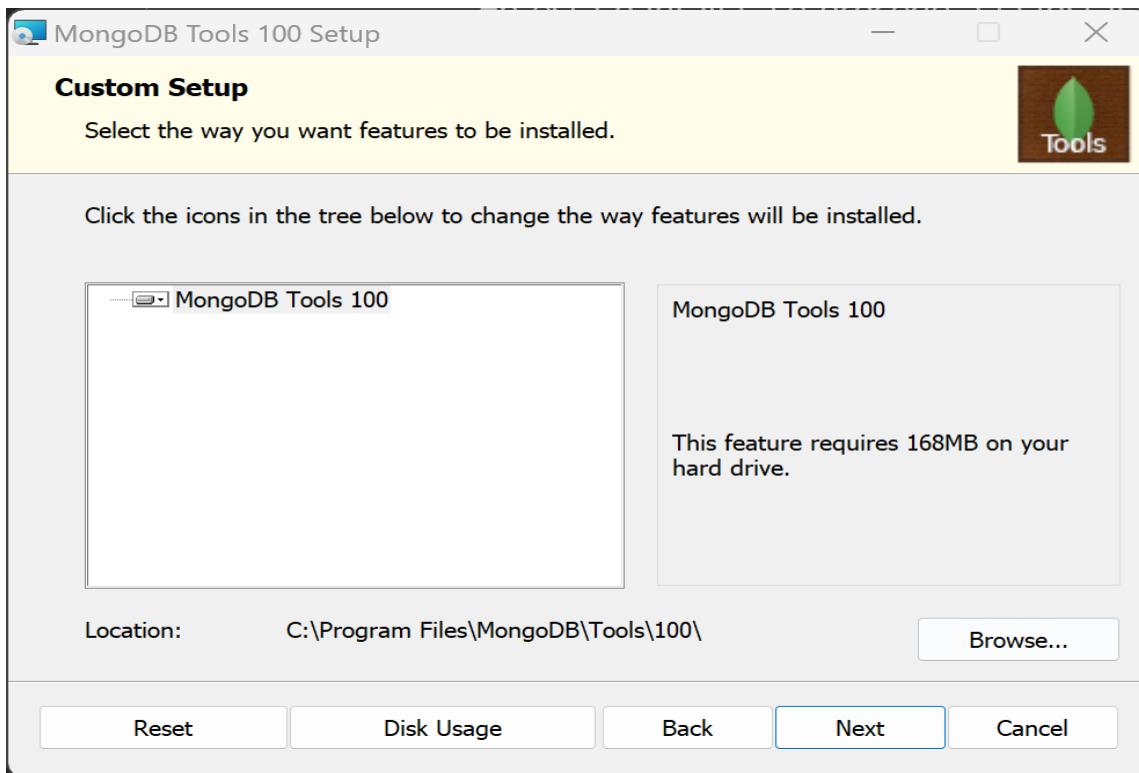
En esta pantalla, se procede a descargar las herramientas de línea de comandos de MongoDB son un conjunto de utilidades independientes que permiten gestionar, importar, exportar y manipular datos de manera eficiente en despliegues de bases de datos MongoDB.



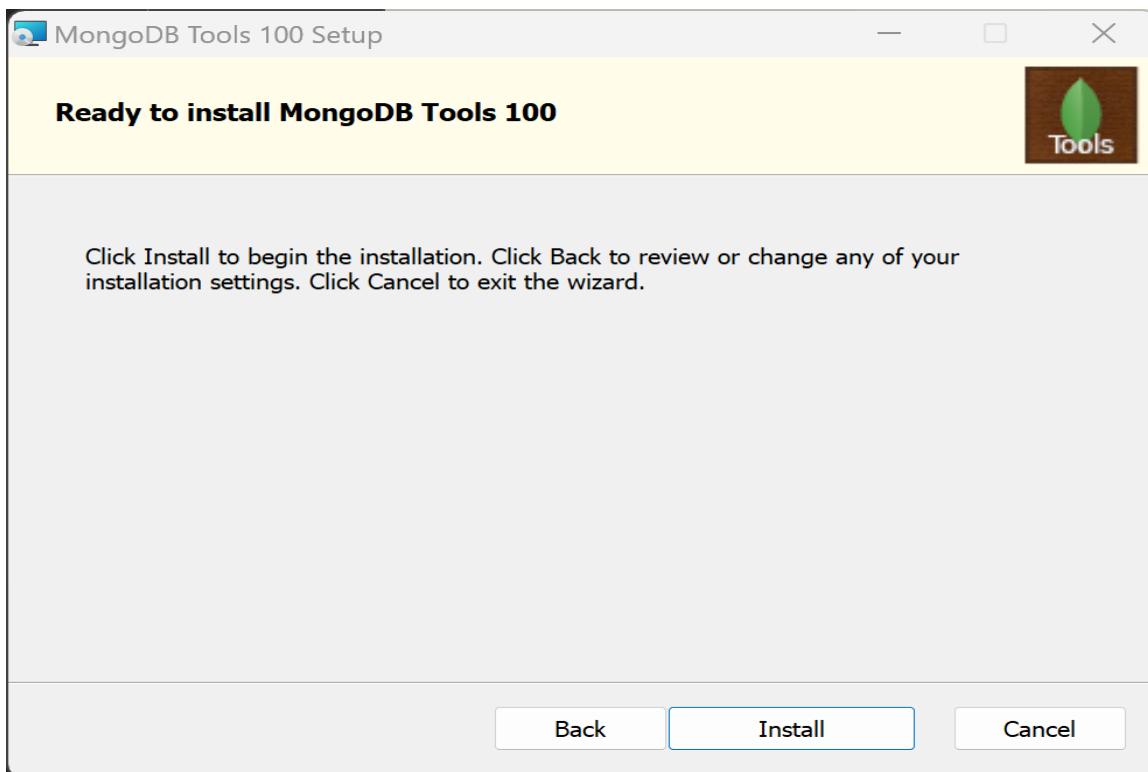
En esta pantalla, se procede a la instalación del archivo mongodb-database-tools-windows-x86\_64-100.10.0.msi y hacemos clic en el botón "Next".



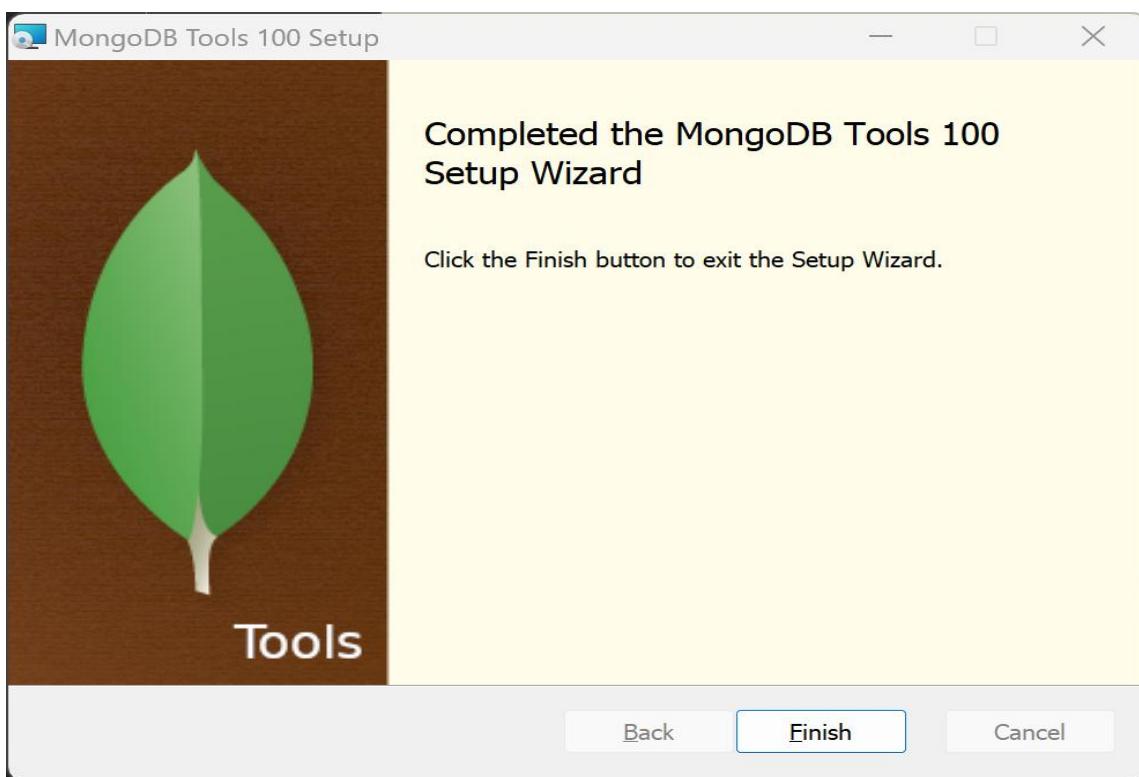
En esta pantalla, aceptamos los términos de la licencia y hacemos clic en el botón "Next" para continuar.



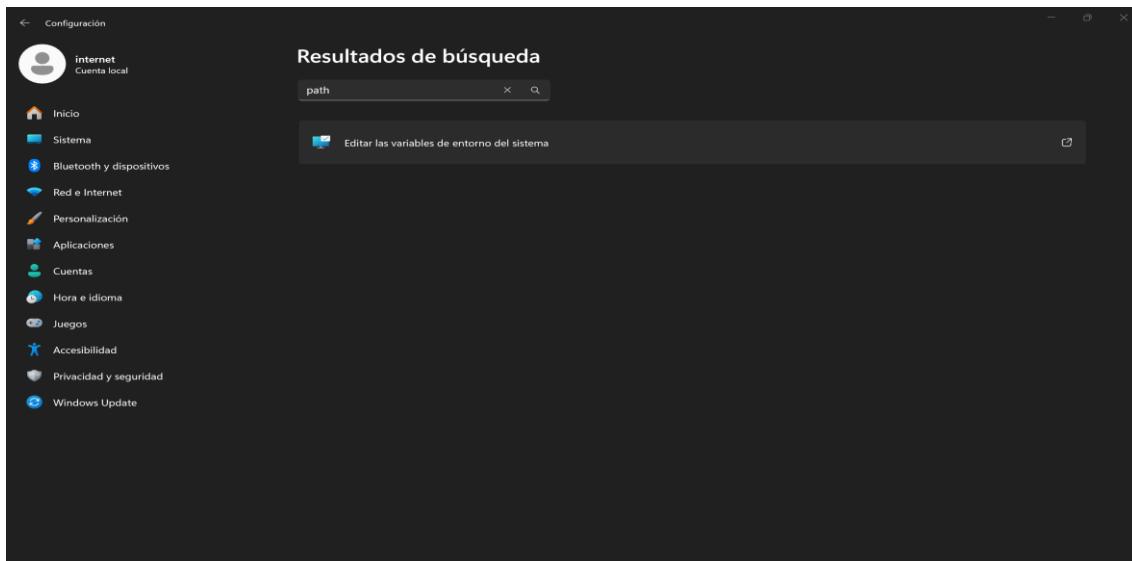
En esta pantalla, mantenemos la configuración predeterminada y hacemos clic en el botón "Next" para continuar.



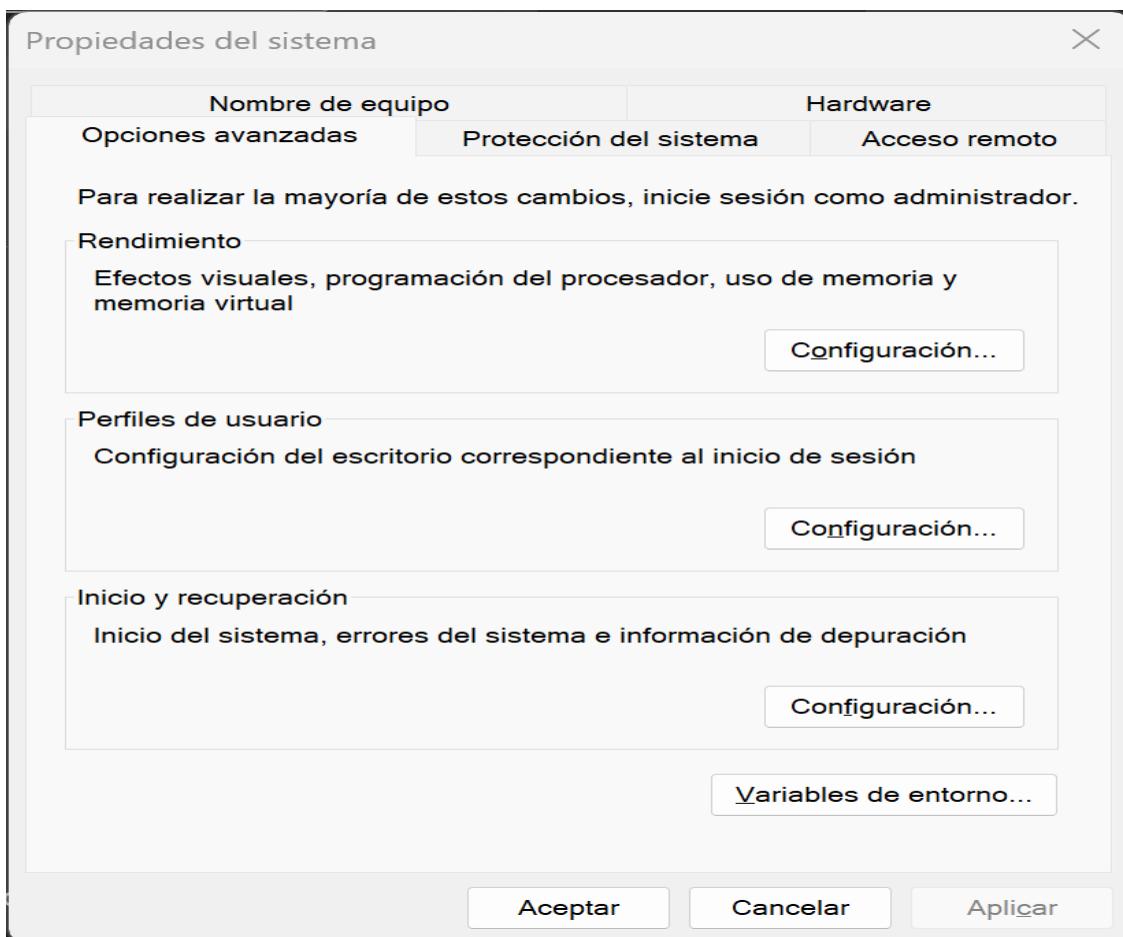
En esta pantalla, procedemos a la instalación de MongoDB Tools versión 100 y hacemos clic en el botón "Install" para continuar con el proceso.



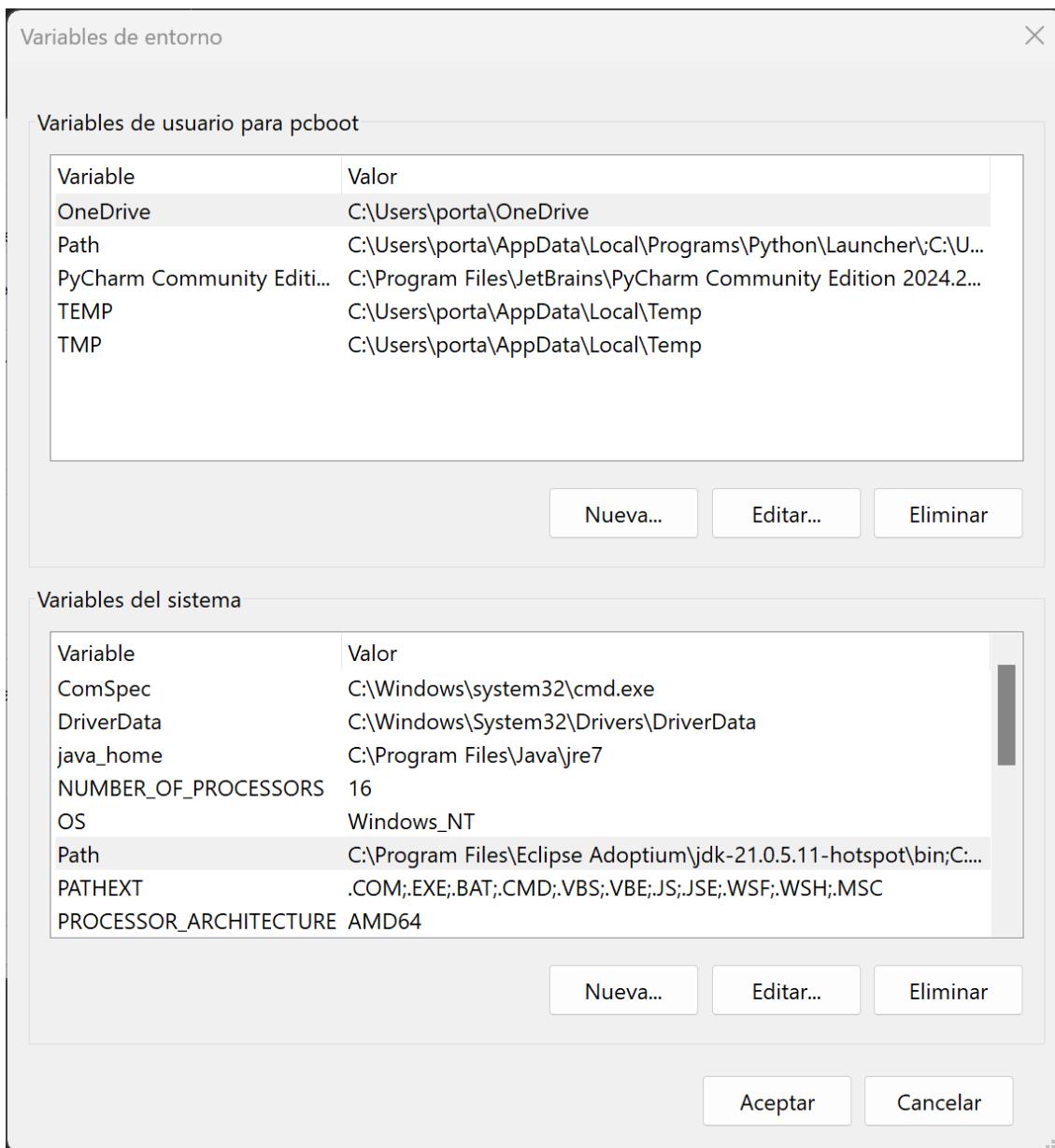
En esta pantalla, finalizamos la instalación haciendo clic en el botón "Finish".



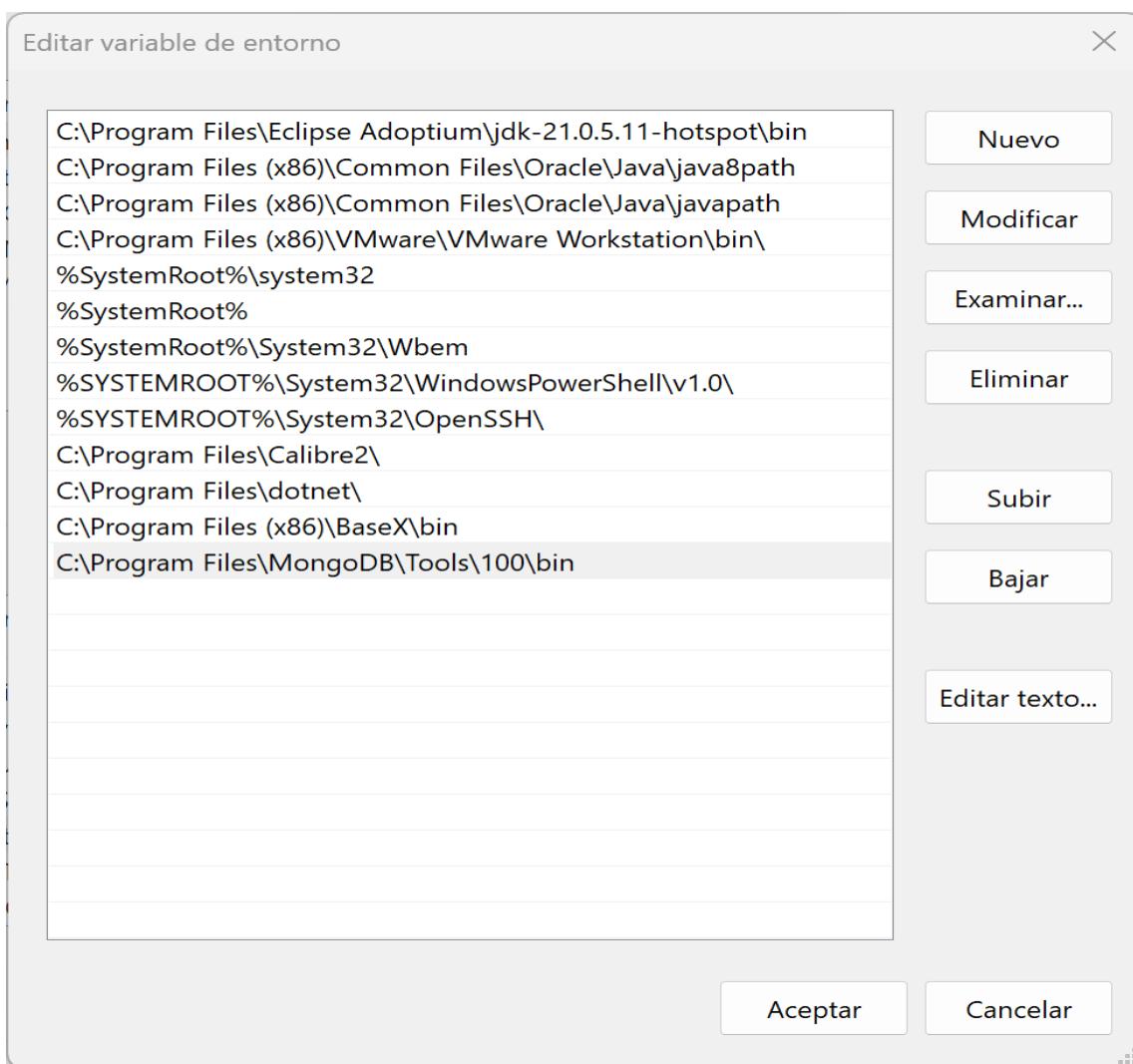
En la configuración avanzada de Windows, procederemos a modificar las variables de entorno del sistema para incorporar la ruta C:\Program Files\MongoDB\Tools\100\bin, donde se encuentran los ejecutables de las herramientas de MongoDB. Esta acción permitirá invocar dichos comandos desde cualquier ubicación en el sistema.



En esta pantalla, procederemos a hacer clic en el botón de "Variables de entorno".



En esta pantalla, se procederá a editar la variable **Path** para añadir la ubicación de las herramientas de **MongoDB Tools**, que es C:\Program Files\MongoDB\Tools\100\bin.



En esta pantalla, se ha procedido a añadir la ruta C:\Program Files\MongoDB\Tools\100\bin al **Path** pulsando el botón "Nuevo" para incluirla.

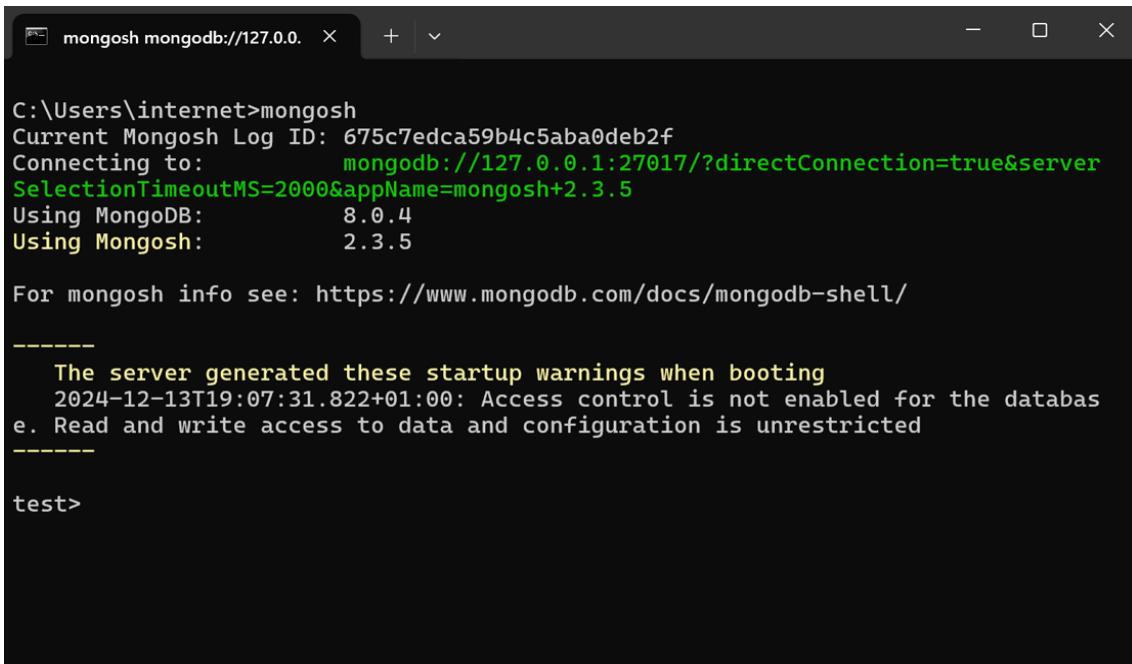
En resumen, con esto concluimos exitosamente la instalación del entorno necesario, lo que nos permitirá utilizar todas las herramientas y funcionalidades de manera óptima y eficiente. Este proceso de configuración es fundamental para garantizar un rendimiento adecuado y facilitar el trabajo con la base de datos, asegurando así que podamos aprovechar al máximo las capacidades que ofrece el sistema. Ahora estamos listos para comenzar a explorar y utilizar las diversas características disponibles en nuestro entorno recién instalado.

### 3.- Ejercicios

Ejercicio 1: Creación de la Base de Datos [citibike](#).

Paso 1: Abrir la shell de MongoDB

Primero, abre la consola o terminal del sistema operativo y escribe el siguiente comando para iniciar la shell de MongoDB:



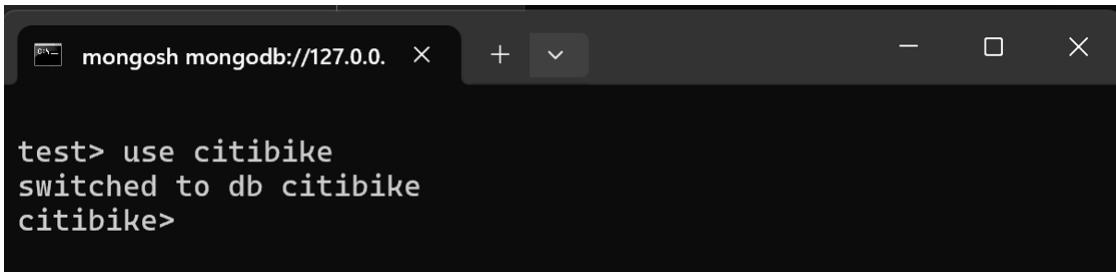
```
mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000&appName=mongosh+2.3.5
Using MongoDB: 8.0.4
Using Mongosh: 2.3.5

For mongosh info see: https://www.mongodb.com/docs/mongodb-shell/
-----
The server generated these startup warnings when booting
2024-12-13T19:07:31.822+01:00: Access control is not enabled for the database.
Read and write access to data and configuration is unrestricted
-----
test>
```

En esta pantalla, procedemos a ejecutar `mongosh` y presionamos la tecla Enter para iniciar la shell interactiva de MongoDB.

El comando `mongosh` inicia la shell interactiva de MongoDB, permitiéndote interactuar directamente con las bases de datos, ejecutar consultas, gestionar colecciones y realizar operaciones administrativas en un entorno de línea de comandos.

Paso 2: Cambiar al contexto de la base de datos



```
test> use citibike
switched to db citibike
citibike>
```

En esta pantalla, se procede a ejecutar el comando `use citibike` para cambiar el contexto de trabajo a la base de datos "citibike", creándola automáticamente si no existe, y permitiéndote realizar operaciones sobre ella.

### Paso 3: Verificar la creación de la base de datos

```
mongosh mongodb://127.0.0.1:27017/test
-----
test> show dbs
admin    40.00 KiB
config   108.00 KiB
local    72.00 KiB
test>
```

Nota: La base de datos `citibike` no aparecerá en la lista hasta que contenga al menos una colección con datos.

### Ejercicio 2: Crear una Colección con las Iniciales del Nombre y Apellidos.

#### Paso 1: Moverse a la base de datos `citibike`

Desde la shell de MongoDB, cambia al contexto de la base de datos `citibike` creada en el **Ejercicio 1** utilizando:

```
mongosh mongodb://127.0.0.1:27017/citibike
-----
citibike> use citibike
already on db citibike
citibike>
```

En esta pantalla, se ha procedido a ejecutar el comando `use citibike` para acceder a la base de datos, lo que permite trabajar con ella y realizar operaciones específicas dentro de su contexto.

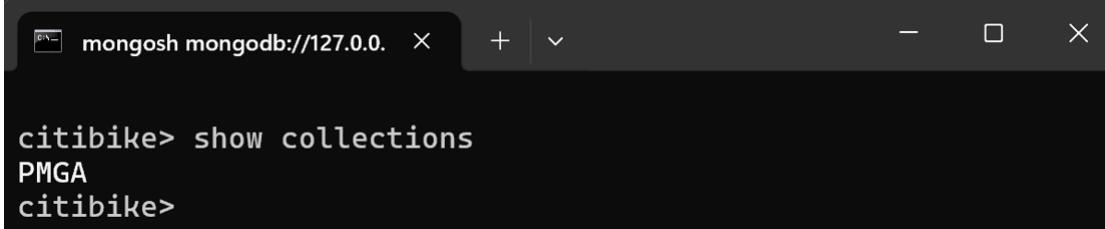
## Paso 2: Crear la Colección



```
mongosh mongodb://127.0.0. X + | v
citibike> db.createCollection("PMGA")
{ ok: 1 }
citibike>
```

En esta pantalla, procedemos a crear una nueva colección utilizando el comando `db.createCollection("PMGA")` en la base de datos `citibike` y presionamos la tecla Enter. Este comando crea una colección vacía llamada "PMGA" en la base de datos actual, donde podremos almacenar documentos posteriormente.

## Paso 3: Verificar la creación de la colección



```
mongosh mongodb://127.0.0. X + | v
citibike> show collections
PMGA
citibike>
```

En esta pantalla, procedemos a verificar si la colección se ha creado correctamente ejecutando el comando `show collections` y presionando la tecla Enter, lo que nos mostrará una lista de las colecciones existentes en la base de datos actual.

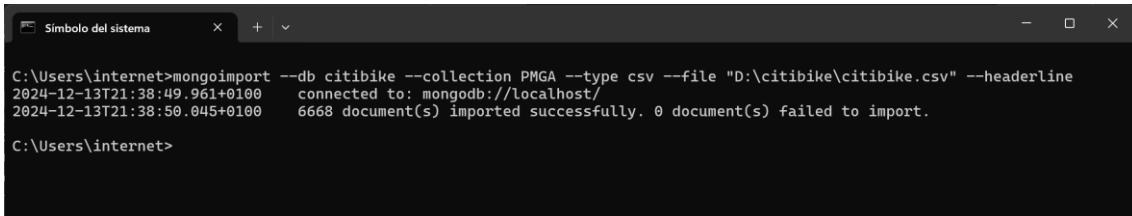
## Ejercicio 3: Importación del Archivo `citibike.csv`.

### Paso 1: Ubicar el archivo CSV

Antes de proceder con el comando, asegúrate de que el archivo `citibike.csv` se encuentra en un directorio accesible desde la terminal. Nota la ruta completa del archivo, ya que la necesitarás para el comando de importación.

## Paso 2: Ejecutar el comando mongoimport

Desde la terminal (no dentro de la shell de MongoDB), ejecuta el siguiente comando:

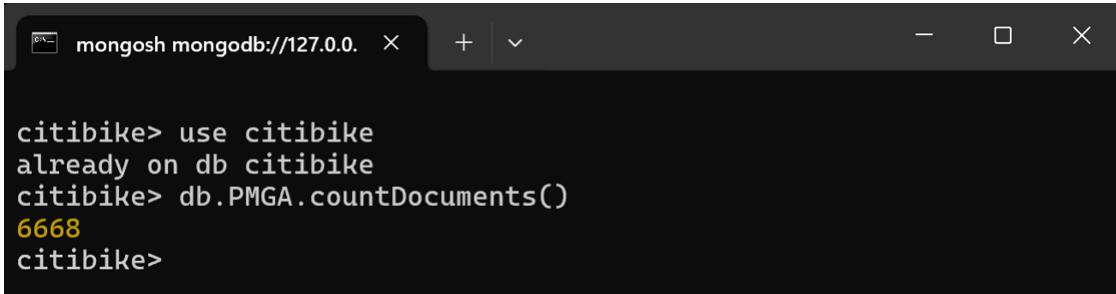


```
C:\Users\internet>mongoimport --db citibike --collection PMGA --type csv --file "D:\citibike\citibike.csv" --headerline
2024-12-13T21:38:49.961+0100      connected to: mongodb://localhost/
2024-12-13T21:38:50.045+0100      6668 document(s) imported successfully. 0 document(s) failed to import.
C:\Users\internet>
```

En esta pantalla, se ejecuta el comando `mongoimport --db citibike --collection PMGA --type csv --file "D:\citibike\citibike.csv" --headerline` y se presiona la tecla "Enter" para importar la base de datos.

## Ejercicio 4: Mostrar el Número de Viajes en la Colección.

Para conocer el número total de documentos (viajes) almacenados en la colección creada en el Ejercicio 2, se utiliza el siguiente comando en la shell de MongoDB con el comando mongosh :



```
mongosh mongodb://127.0.0. × + | v
citibike> use citibike
already on db citibike
citibike> db.PMGA.countDocuments()
6668
citibike>
```

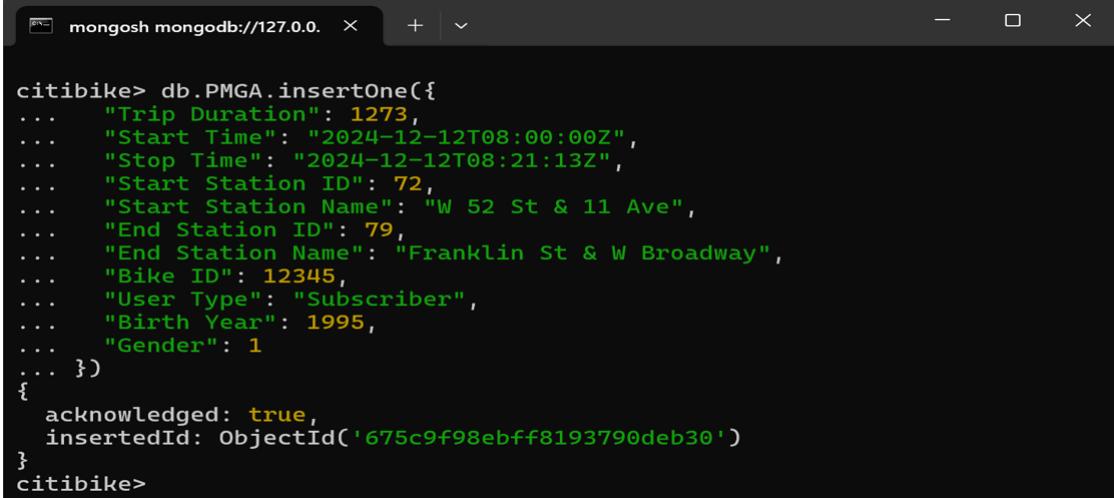
En esta pantalla, se procede a ejecutar el comando `use citibike` para acceder a la base de datos, y posteriormente se emplea la instrucción `db.PMGA.countDocuments()` para obtener el recuento total de documentos en la colección PMGA, que representa el número de viajes registrados.

## Ejercicio 5: Inserción de un Nuevo Viaje.

En este ejercicio, se requiere insertar manualmente un nuevo viaje en la colección creada en el **Ejercicio 2**, utilizando datos inventados. A continuación, se detalla el proceso:

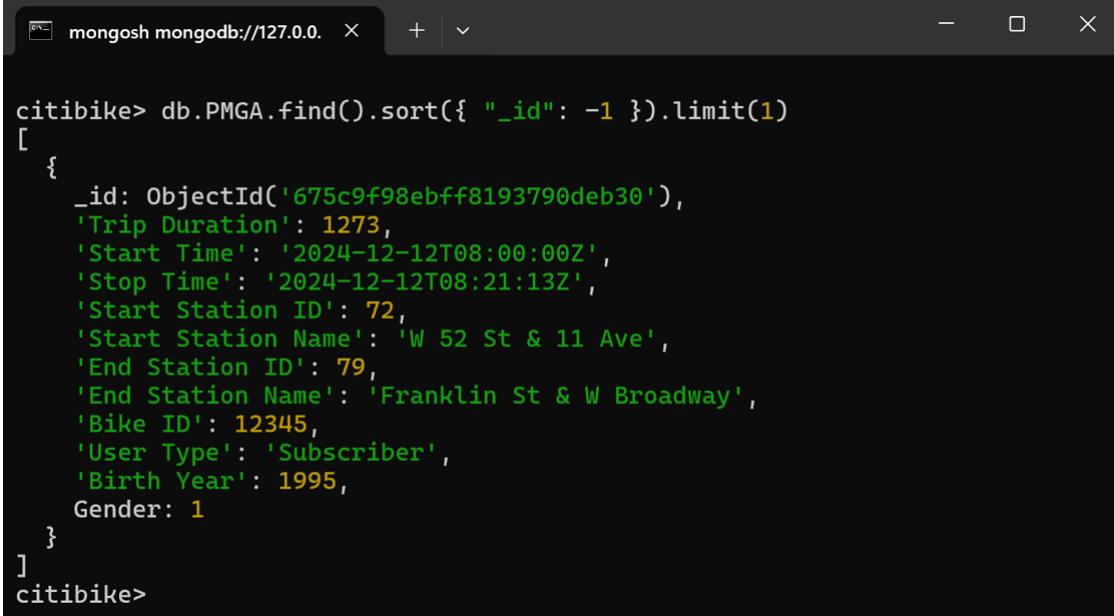
## Comando para Insertar el Viaje

En la shell de MongoDB, utiliza el siguiente comando para insertar un nuevo documento en la colección:



```
mongosh mongodb://127.0.0.1:27017/citibike> db.PMGA.insertOne({
...   "Trip Duration": 1273,
...   "Start Time": "2024-12-12T08:00:00Z",
...   "Stop Time": "2024-12-12T08:21:13Z",
...   "Start Station ID": 72,
...   "Start Station Name": "W 52 St & 11 Ave",
...   "End Station ID": 79,
...   "End Station Name": "Franklin St & W Broadway",
...   "Bike ID": 12345,
...   "User Type": "Subscriber",
...   "Birth Year": 1995,
...   "Gender": 1
... })
{
  acknowledged: true,
  insertedId: ObjectId('675c9f98ebff8193790deb30')
}
citibike>
```

En esta pantalla, se procede a implementar una operación de inserción utilizando el método `insertOne()`. Esta función incorpora un nuevo registro en la colección PMGA, el cual representa un trayecto específico con información detallada que incluye la duración del viaje, los momentos precisos de inicio y finalización, las estaciones de partida y llegada, así como datos apropiados sobre el usuario y la bicicleta empleada.



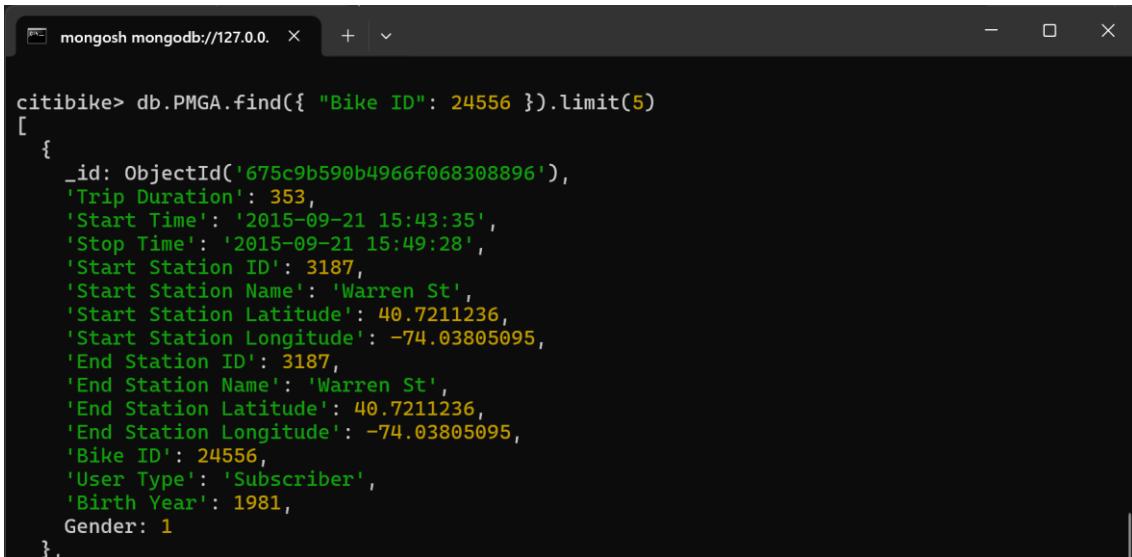
```
mongosh mongodb://127.0.0.1:27017/citibike> db.PMGA.find().sort({ "_id": -1 }).limit(1)
[
  {
    _id: ObjectId('675c9f98ebff8193790deb30'),
    'Trip Duration': 1273,
    'Start Time': '2024-12-12T08:00:00Z',
    'Stop Time': '2024-12-12T08:21:13Z',
    'Start Station ID': 72,
    'Start Station Name': 'W 52 St & 11 Ave',
    'End Station ID': 79,
    'End Station Name': 'Franklin St & W Broadway',
    'Bike ID': 12345,
    'User Type': 'Subscriber',
    'Birth Year': 1995,
    Gender: 1
  }
]
citibike>
```

En esta pantalla, se ejecuta la consulta `db.PMGA.find().sort({ "_id": -1 }).limit(1)` para recuperar y visualizar el documento más

reciente insertado en la colección PMGA, permitiendo así verificar la correcta incorporación del nuevo registro.

### Ejercicio 6: Obtener los 5 Primeros Viajes para la Bicicleta con Identificador 24556.

En este ejercicio, se requiere obtener los primeros 5 viajes registrados para una bicicleta específica con el identificador 24556.



```
mongosh mongodb://127.0.0.1:27017/citibike> db.PMGA.find({ "Bike ID": 24556 }).limit(5)
[{"_id": ObjectId('675c9b590b4966f068308896'), "Trip Duration": 353, "Start Time": "2015-09-21 15:43:35", "Stop Time": "2015-09-21 15:49:28", "Start Station ID": 3187, "Start Station Name": "Warren St", "Start Station Latitude": 40.7211236, "Start Station Longitude": -74.03805095, "End Station ID": 3187, "End Station Name": "Warren St", "End Station Latitude": 40.7211236, "End Station Longitude": -74.03805095, "Bike ID": 24556, "User Type": "Subscriber", "Birth Year": 1981, "Gender": 1}
```

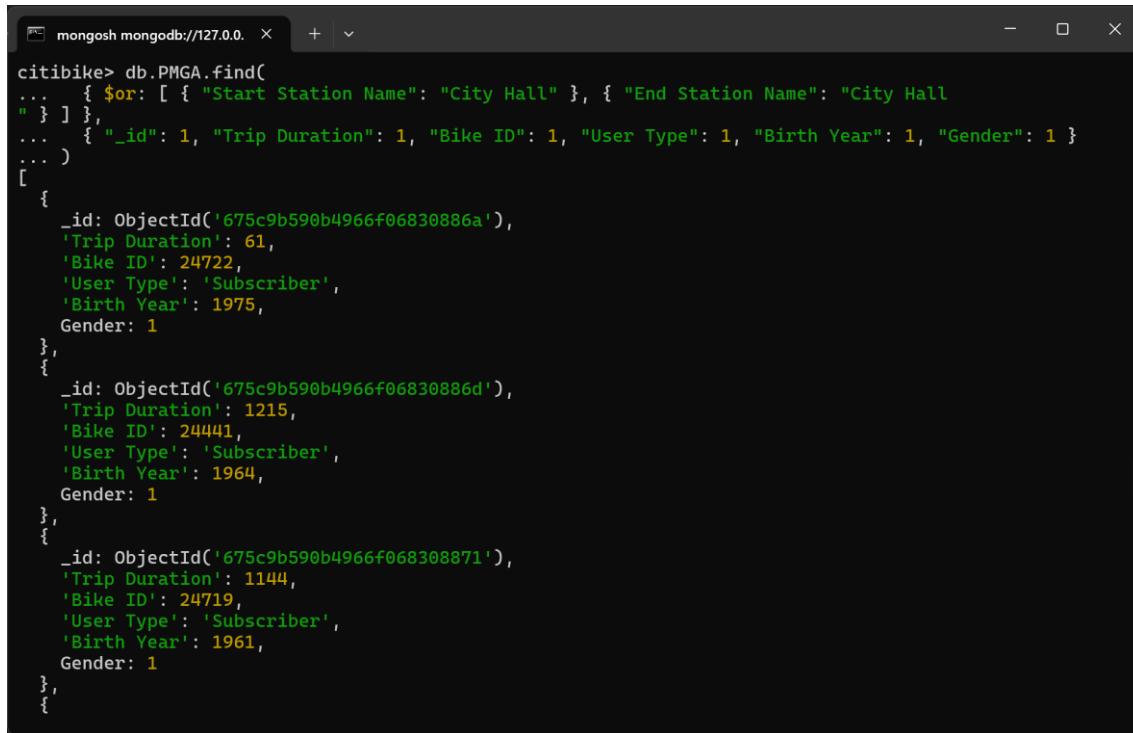
Esta instrucción ejecuta una consulta distinguida en la colección PMGA de la base de datos citibike, filtrando y exhibiendo los cinco primeros registros donde el identificador de bicicleta corresponde al número 24556. La operación revela información detallada sobre estos trayectos específicos. El código en cuestión es: `db.PMGA.find({ "Bike ID": 24556 }).limit(5)`

_id	Trip Duration	Start Time	Stop Time	Start Station Name	End Station Name	Bike ID	User Type	Birth Year	Gender
675c9b590b4966f068308896	353	21/09/2015 15:43:35	21/09/2015 15:49:28	Warren St	Warren St	24556	Subscriber	1981	1
675c9b590b4966f068308cd	550	23/09/2015 12:49	23/09/2015 12:59	Warren St	Newport PATH	24556	Subscriber	1985	2
675c9b590b4966f068308cd	598	23/09/2015 14:32	23/09/2015 14:42	Newport PATH	Silver Lake Park	24556	Subscriber	1977	2
675c9b590b4966f068308d24	3390	23/09/2015 17:18	23/09/2015 18:15	Silver Lake Park	Newport PATH	24556	Customer	(Vacio)	0
675c9b590b4966f068308d6d	260	23/09/2015 18:15	23/09/2015 18:20	Newport PATH	Warren St	24556	Subscriber	1988	2

El resultado muestra una lista de 5 registros de viajes realizados por usuarios, con detalles como la duración del viaje, estaciones de inicio y fin, identificador de la bicicleta, tipo de usuario, año de nacimiento y género, incluyendo usuarios masculinos y femeninos, y un viaje con un género desconocido y año de nacimiento vacío.

## Ejercicio 7: Obtener los Viajes cuya Estación de Salida o Llegada sea City Hall.

En este ejercicio, se requiere obtener los viajes donde la estación de salida o de llegada sea "**City Hall**", y mostrar solo ciertos campos de cada viaje: identificador del viaje, duración, identificador de la bicicleta, tipo de usuario, año de nacimiento y género.



```
mongosh mongodb://127.0.0.1:27017
db.PMGA.find(
  { $or: [ { "Start Station Name": "City Hall" }, { "End Station Name": "City Hall" } ] },
  { "_id": 1, "Trip Duration": 1, "Bike ID": 1, "User Type": 1, "Birth Year": 1, "Gender": 1 }
)
[ {
  _id: ObjectId('675c9b590b4966f06830886a'),
  'Trip Duration': 61,
  'Bike ID': 24722,
  'User Type': 'Subscriber',
  'Birth Year': 1975,
  Gender: 1
},
{
  _id: ObjectId('675c9b590b4966f06830886d'),
  'Trip Duration': 1215,
  'Bike ID': 24441,
  'User Type': 'Subscriber',
  'Birth Year': 1964,
  Gender: 1
},
{
  _id: ObjectId('675c9b590b4966f068308871'),
  'Trip Duration': 1144,
  'Bike ID': 24719,
  'User Type': 'Subscriber',
  'Birth Year': 1961,
  Gender: 1
},
{
  _id: ObjectId('675c9b590b4966f068308872'),
  'Trip Duration': 1967,
  'Bike ID': 24706,
  'User Type': 'Subscriber',
  'Birth Year': 1952,
  Gender: 1
},
{
  _id: ObjectId('675c9b590b4966f06830887c'),
  'Trip Duration': 192,
  'Bike ID': 24442,
  'User Type': 'Subscriber',
  'Birth Year': 1976,
  Gender: 1
},
{
  _id: ObjectId('675c9b590b4966f068308888'),
  'Trip Duration': 216,
  'Bike ID': 24442,
  'User Type': 'Subscriber',
  'Birth Year': 1976,
  Gender: 1
},
{
  _id: ObjectId('675c9b590b4966f06830888d'),
  'Trip Duration': 448,
  'Bike ID': 24523,
  'User Type': 'Subscriber',
  'Birth Year': 1961,
  Gender: 1
},
{
  _id: ObjectId('675c9b590b4966f0683088a6'),
  'Trip Duration': 1076,
  'Bike ID': 24716,
  'User Type': 'Subscriber',
  'Birth Year': 1970,
  Gender: 1
},
{
  _id: ObjectId('675c9b590b4966f0683088ab'),
  'Trip Duration': 546,
  'Bike ID': 24444,
  'User Type': 'Subscriber',
  'Birth Year': 1990,
  Gender: 1
},
{
  _id: ObjectId('675c9b590b4966f0683088ae'),
  'Trip Duration': 310,
  'Bike ID': 24521,
  'User Type': 'Subscriber',
  'Birth Year': 1983,
  Gender: 2
},
{
  _id: ObjectId('675c9b590b4966f0683088b2'),
  'Trip Duration': 330,
  'Bike ID': 24488,
  'User Type': 'Subscriber',
  'Birth Year': 1962,
  Gender: 1
},
{
  _id: ObjectId('675c9b590b4966f0683088b4'),
  'Trip Duration': 307,
  'Bike ID': 24450,
  'User Type': 'Subscriber',
  'Birth Year': 1960,
  Gender: 2
},
{
  _id: ObjectId('675c9b590b4966f0683088c6'),
  'Trip Duration': 437,
  'Bike ID': 24625,
  'User Type': 'Subscriber',
  'Birth Year': 1981,
  Gender: 2
},
{
  _id: ObjectId('675c9b590b4966f0683088c9'),
  'Trip Duration': 757,
  'Bike ID': 24427,
  'User Type': 'Subscriber',
  'Birth Year': 1962,
  Gender: 1
},
{
  _id: ObjectId('675c9b590b4966f0683088d6'),
  'Trip Duration': 587,
  'Bike ID': 24422,
  'User Type': 'Subscriber',
  'Birth Year': 1960,
  Gender: 2
},
{
  _id: ObjectId('675c9b590b4966f0683088dc'),
  'Trip Duration': 1644,
  'Bike ID': 24688,
  'User Type': 'Subscriber',
  'Birth Year': 1988,
  Gender: 1
},
{
  _id: ObjectId('675c9b590b4966f0683088dd'),
  'Trip Duration': 129,
  'Bike ID': 24617,
  'User Type': 'Subscriber',
  'Birth Year': 1970,
  Gender: 1
},
{
  _id: ObjectId('675c9b590b4966f0683088ee'),
  'Trip Duration': 727,
  'Bike ID': 24623,
  'User Type': 'Subscriber',
  'Birth Year': 1974,
  Gender: 1
},
{
  _id: ObjectId('675c9b590b4966f0683088ef'),
  'Trip Duration': 965,
  'Bike ID': 24448,
  'User Type': 'Subscriber',
  'Birth Year': 1985,
  Gender: 1
},
{
  _id: ObjectId('675c9b590b4966f0683088f9'),
  'Trip Duration': 303,
  'Bike ID': 24617,
  'User Type': 'Subscriber',
  'Birth Year': 1989,
  Gender: 1
} ]
```

Esta instrucción ejecuta una búsqueda avanzada en la colección PMGA para identificar trayectos que tengan como punto de partida o destino la estación "City Hall". La consulta extrae campos específicos, incluyendo la duración del viaje, el identificador único de la bicicleta y datos demográficos del usuario. El código correspondiente es:

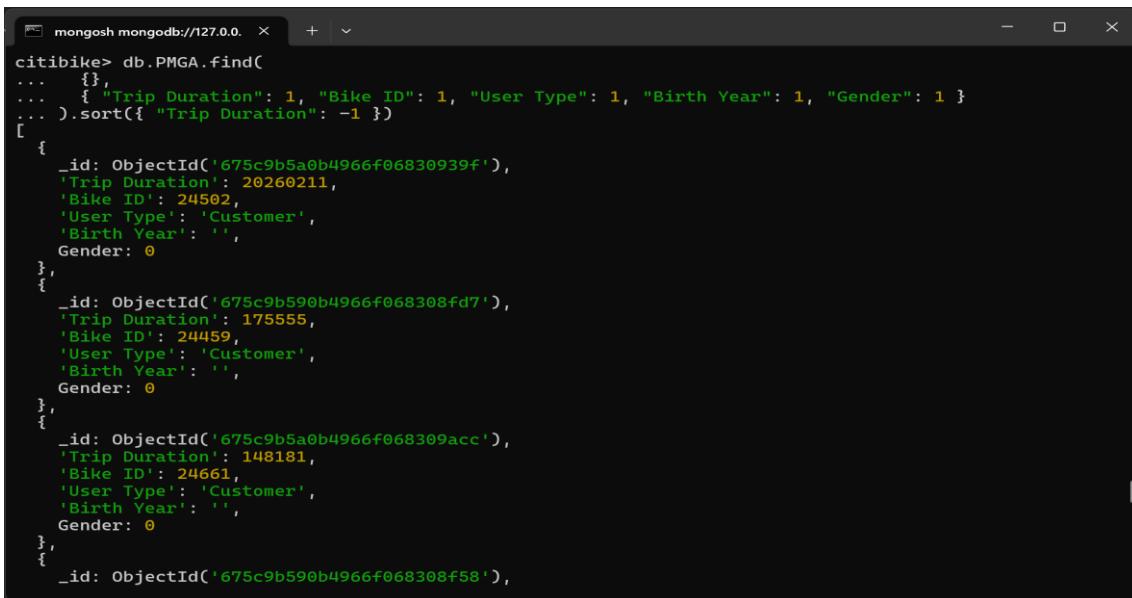
```
db.PMGA.find( { $or: [ { "Start Station Name": "City Hall" }, { "End Station Name": "City Hall" } ] }, { "_id": 1, "Trip Duration": 1, "Bike ID": 1, "User Type": 1, "Birth Year": 1, "Gender": 1 } ).
```

_id	Trip Duration	Bike ID	User Type	Birth Year	Gender
675c9b590b4966f06830886a	61	24722	Subscriber	1975	1
675c9b590b4966f06830886d	1215	24441	Subscriber	1964	1
675c9b590b4966f068308871	1144	24719	Subscriber	1961	1
675c9b590b4966f068308872	1967	24706	Subscriber	1952	1
675c9b590b4966f06830887c	192	24442	Subscriber	1976	1
675c9b590b4966f068308888	216	24442	Subscriber	1976	1
675c9b590b4966f06830888d	448	24523	Subscriber	1961	1
675c9b590b4966f0683088a6	1076	24716	Subscriber	1970	1
675c9b590b4966f0683088ab	546	24444	Subscriber	1990	1
675c9b590b4966f0683088ae	310	24521	Subscriber	1983	2
675c9b590b4966f0683088b2	330	24488	Subscriber	1962	1
675c9b590b4966f0683088b4	307	24450	Subscriber	1960	2
675c9b590b4966f0683088c6	437	24625	Subscriber	1981	2
675c9b590b4966f0683088c9	757	24427	Subscriber	1962	1
675c9b590b4966f0683088d6	587	24422	Subscriber	1960	2
675c9b590b4966f0683088dc	1644	24688	Subscriber	1988	1
675c9b590b4966f0683088dd	129	24617	Subscriber	1970	1
675c9b590b4966f0683088ee	727	24623	Subscriber	1974	1
675c9b590b4966f0683088ef	965	24448	Subscriber	1985	1
675c9b590b4966f0683088f9	303	24617	Subscriber	1989	1

La tabla muestra datos de viajes en bicicleta, incluyendo duración, identificadores únicos de viaje y bicicleta, tipo de usuario, año de nacimiento y género, para analizar patrones de uso y demografía.

### Ejercicio 8: Obtener los Viajes Ordenados por Duración.

En este ejercicio, se requiere obtener los viajes ordenados por el tiempo de duración en orden descendente, es decir, los viajes más largos deben aparecer primero. Además, se deben mostrar solo los siguientes campos: duración del viaje, identificador de la bicicleta, tipo de usuario, año de nacimiento y género.



```
mongosh mongodb://127.0.0.1:27017/citibike> db.PMGA.find({},{ "Trip Duration": 1, "Bike ID": 1, "User Type": 1, "Birth Year": 1, "Gender": 1 }).sort({ "Trip Duration": -1 })
[ { _id: ObjectId('675c9b5a0b4966f06830939f'), 'Trip Duration': 20260211, 'Bike ID': 24502, 'User Type': 'Customer', 'Birth Year': '', Gender: 0 }, { _id: ObjectId('675c9b590b4966f068308fd7'), 'Trip Duration': 175555, 'Bike ID': 24459, 'User Type': 'Customer', 'Birth Year': '', Gender: 0 }, { _id: ObjectId('675c9b5a0b4966f068309acc'), 'Trip Duration': 148181, 'Bike ID': 24661, 'User Type': 'Customer', 'Birth Year': '', Gender: 0 }, { _id: ObjectId('675c9b590b4966f068308f58')},
```

Este código ejecuta una consulta sofisticada en la colección PMGA, extrayendo campos selectos de todos los registros y organizándolos en orden decreciente según la duración del trayecto. La consulta exhibe los veinte viajes de mayor extensión temporal, proporcionando información detallada que incluye el identificador único de la bicicleta utilizada y la categoría del usuario que realizó el viaje:  
`db.PMGA.find( {}, { "Trip Duration": 1, "Bike ID": 1, "User Type": 1, "Birth Year": 1, "Gender": 1 } ).sort({ "Trip Duration": -1 })`.

_id	Trip Duration	Bike ID	User Type	Birth Year	Gender
675c9b5a0b4966f06830939f	20260211	24502	Customer	(Vacío)	0
675c9b590b4966f068308fd7	175555	24459	Customer	(Vacío)	0
675c9b5a0b4966f068309acc	148181	24661	Customer	(Vacío)	0
675c9b590b4966f068308f58	109651	24663	Customer	(Vacío)	0
675c9b5a0b4966f06830984c	100524	24488	Subscriber	1985	2
675c9b590b4966f068308c81	98790	24637	Subscriber	1996	2
675c9b5a0b4966f0683099b3	88131	24697	Subscriber	1975	2
675c9b590b4966f0683088b1	87733	24432	Customer	(Vacío)	0
675c9b5a0b4966f0683093b3	86730	24633	Customer	(Vacío)	0
675c9b5a0b4966f068309e84	85649	24641	Customer	(Vacío)	0
675c9b590b4966f06830892c	85113	24535	Customer	(Vacío)	0
675c9b5a0b4966f068309397	83521	24575	Customer	(Vacío)	0
675c9b5a0b4966f0683093df	83349	24586	Customer	(Vacío)	0
675c9b5a0b4966f068309469	82716	24660	Customer	(Vacío)	0
675c9b5a0b4966f06830a0ee	81246	24595	Customer	(Vacío)	0
675c9b590b4966f068308a8e	75700	24566	Customer	(Vacío)	0
675c9b5a0b4966f068309688	65256	24645	Customer	(Vacío)	0
675c9b590b4966f068308d92	65242	24584	Customer	(Vacío)	0
675c9b590b4966f068308b2e	56671	24535	Subscriber	1985	1
675c9b590b4966f068308fb4	52577	24686	Customer	(Vacío)	0

El resultado muestra una lista de viajes realizados por usuarios (de tipo "Customer" y "Subscriber"), con detalles sobre la duración del viaje, el identificador de la bicicleta, el tipo de usuario, el año de nacimiento (vacío en algunos casos) y el género (siendo la mayoría con género desconocido, indicado por 0).

### Ejercicio 9: Actualización de la Denominación de la Estación con Identificador 3203.

En este ejercicio, se requiere actualizar el nombre de la estación con el identificador 3203, cambiando su nombre de "**Hamilton Park**" a "**Silver Lake Park**". La actualización debe aplicarse tanto a los viajes con origen como a los viajes con destino en esta estación.

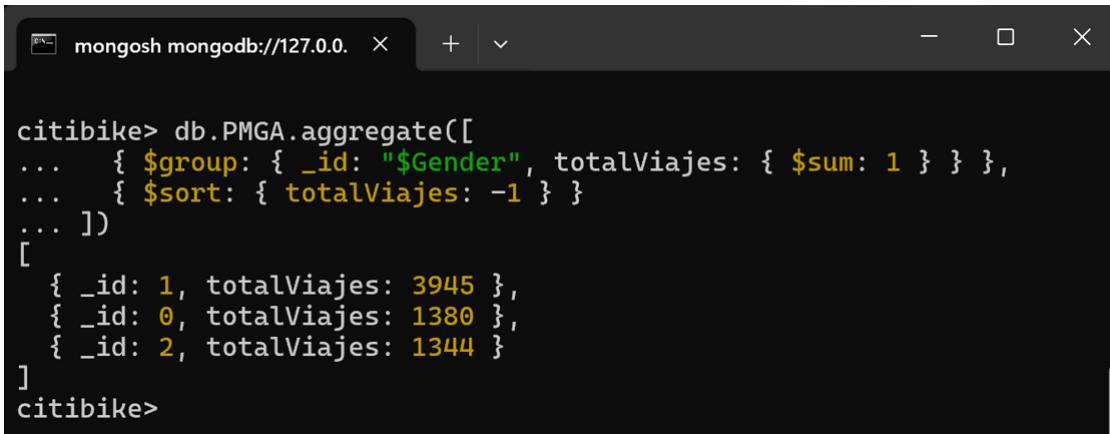
```
mongosh mongodb://127.0.0.1:27017
citibike> db.PMGA.updateMany(
...   { "Start Station ID": 3203 },
...   { $set: { "Start Station Name": "Silver Lake Park" } }
... )
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 530,
  modifiedCount: 530,
  upsertedCount: 0
}
citibike> db.PMGA.updateMany(
...   { "End Station ID": 3203 },
...   { $set: { "End Station Name": "Silver Lake Park" } }
... )
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 580,
  modifiedCount: 580,
  upsertedCount: 0
}
citibike>
```

Este código ejecuta dos operaciones de actualización masiva en la colección PMGA, modificando el nombre de la estación con ID 3203 a "Silver Lake Park" tanto para los viajes que inician como para los que terminan en dicha estación, utilizando el método `updateMany()` de MongoDB.

```
db.PMGA.updateMany({ "Start Station ID": 3203 }, { $set: { "Start Station Name": "Silver Lake Park" } })
db.PMGA.updateMany({ "End Station ID": 3203 }, { $set: { "End Station Name": "Silver Lake Park" } })
```

### Ejercicio 10: Determinar quién ha hecho más viajes: Usuarios Masculinos o Femeninos.

En este ejercicio, se utiliza el comando de agregación para agrupar los viajes según el género del usuario y contar cuántos viajes corresponden a cada género.



```
mongosh mongodb://127.0.0.1:27017
citibike> db.PMGA.aggregate([
...   { $group: { _id: "$Gender", totalViajes: { $sum: 1 } } },
...   { $sort: { totalViajes: -1 } }
... ])
[{"_id": 1, "totalViajes": 3945}, {"_id": 0, "totalViajes": 1380}, {"_id": 2, "totalViajes": 1344}
]
citibike>
```

El código realiza un análisis en la colección PMGA, agrupando los documentos por el campo `Gender` y calculando el total de viajes realizados por cada género (`totalViajes`). Luego, ordena los resultados en orden descendente según el número total de viajes.

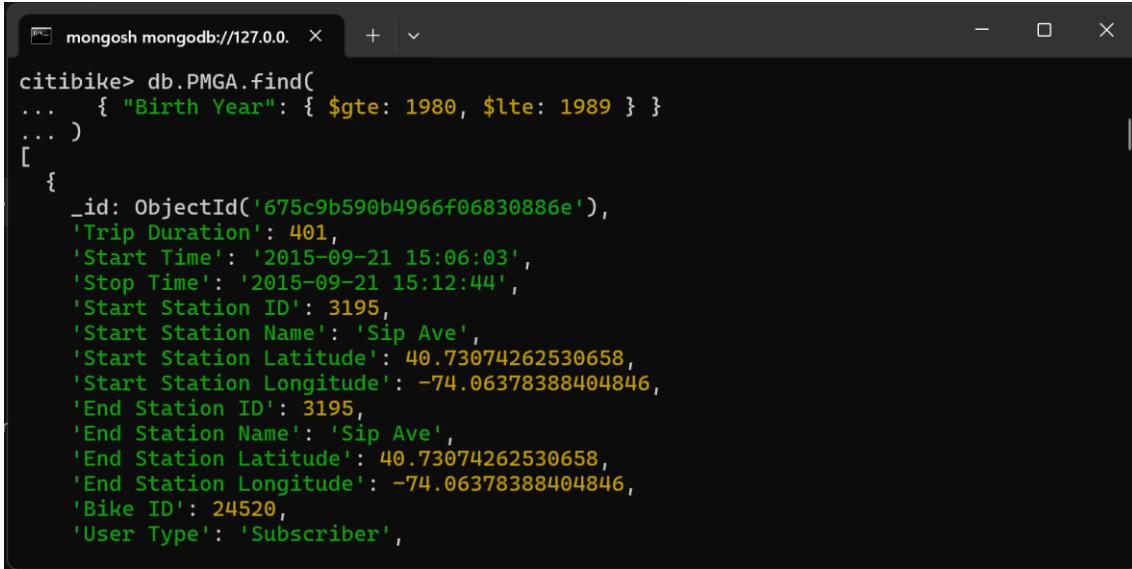
Interpretación del resultado:

- **\_id: 1 (3945 viajes):** Los usuarios masculinos han realizado la mayoría de los viajes.
- **\_id: 0 (1380 viajes):** Los usuarios con género desconocido están en segundo lugar.
- **\_id: 2 (1344 viajes):** Los usuarios femeninos han realizado la menor cantidad de viajes.

En conclusión, los **usuarios masculinos** han realizado más viajes que los usuarios femeninos o los de género desconocido.

## Ejercicio 11: Mostrar los Viajes de los Usuarios Nacidos en la Década de los 80.

En este ejercicio, se requiere mostrar los viajes realizados por usuarios nacidos entre **1980** y **1989**, ambos años incluidos.



```
mongosh mongodb://127.0.0.1:27017/citibike> db.PMGA.find({ "Birth Year": { $gte: 1980, $lte: 1989 } })
```

The screenshot shows a terminal window with the title 'mongosh mongodb://127.0.0.1:27017/citibike'. It displays a MongoDB query to find documents where the 'Birth Year' field is between 1980 and 1989, inclusive. The output shows one document with the following details:

```
_id: ObjectId('675c9b590b4966f06830886e'),  
'Trip Duration': 401,  
'Start Time': '2015-09-21 15:06:03',  
'Stop Time': '2015-09-21 15:12:44',  
'Start Station ID': 3195,  
'Start Station Name': 'Sip Ave',  
'Start Station Latitude': 40.73074262530658,  
'Start Station Longitude': -74.06378388404846,  
'End Station ID': 3195,  
'End Station Name': 'Sip Ave',  
'End Station Latitude': 40.73074262530658,  
'End Station Longitude': -74.06378388404846,  
'Bike ID': 24520,  
'User Type': 'Subscriber',
```

**El comando** busca `db.PMGA.find({ "Birth Year": { $gte: 1980, $lte: 1989 } })` y muestra todos los viajes realizados por usuarios nacidos entre 1980 y 1989, filtrando por el campo **Birth Year**.

<code>_id</code>	<code>Trip Duration</code>	<code>Start Station Name</code>	<code>End Station Name</code>	<code>Bike ID</code>	<code>User Type</code>	<code>Birth Year</code>	<code>Gender</code>
675c9b590b4966f06830886e	401	Sip Ave	Sip Ave	24520	Subscriber	1987	2
675c9b590b4966f068308870	205	Sip Ave	Sip Ave	24555	Subscriber	1987	2
675c9b590b4966f068308878	236	Newport PATH	Silver Lake Park	24400	Subscriber	1984	2
675c9b590b4966f068308879	961	Newark Ave	Grove St PATH	24617	Subscriber	1983	1
675c9b590b4966f06830887b	355	Newark Ave	Silver Lake Park	24443	Subscriber	1982	1
675c9b590b4966f06830887f	349	Silver Lake Park	Newark Ave	24443	Subscriber	1982	1
675c9b590b4966f068308884	375	Paulus Hook	Warren St	24481	Subscriber	1982	1
675c9b590b4966f06830888a	195	Heights Elevator	Riverview Park	24471	Subscriber	1989	2
675c9b590b4966f068308890	1045	Warren St	Oakland Ave	24481	Subscriber	1989	1
675c9b590b4966f068308896	353	Warren St	Warren St	24556	Subscriber	1981	1

El resultado muestra viajes realizados por usuarios nacidos entre 1980 y 1989, incluyendo detalles como la duración del viaje, estaciones de inicio y fin, identificador de la bicicleta, tipo de usuario, año de nacimiento y género. Por ejemplo, un usuario nacido en 1987 realizó un viaje de 401 segundos desde "Sip Ave" hasta "Sip Ave" utilizando la bicicleta 24520.

## Ejercicio 12: Crear una Nueva Colección "genderless" con Viajes sin Género Conocido.

En este ejercicio, vamos a crear una nueva colección llamada **genderless** que contendrá todos los viajes donde el género del usuario

es desconocido (es decir, donde el campo `Gender` tenga el valor `0`). Despu s, mostraremos el n mero de viajes en esta nueva colecci n.

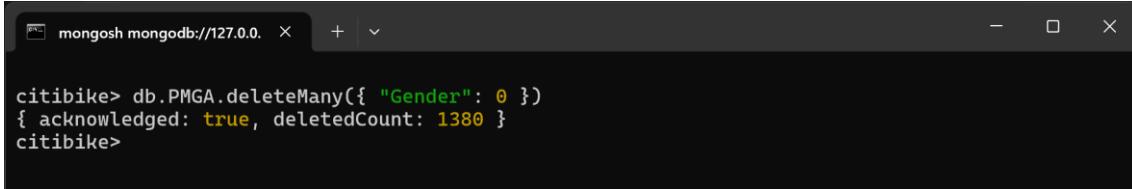


```
mongosh mongodb://127.0.0.1:27017/citibike> db.PMGA.aggregate([ { $match: { "Gender": 0 } }, { $out: "genderless" } ] )  
citibike> db.genderless.countDocuments()  
1380  
citibike> show collections  
genderless  
PMGA  
citibike>
```

El c digo filtra los viajes de la colecci n `PMGA` donde el g nero del usuario es desconocido (`Gender: 0`), los copia a una nueva colecci n llamada `genderless`, y luego muestra el n mero de documentos en la colecci n `genderless` (1380 viajes), confirmando que la colecci n fue creada correctamente.

### Ejercicio 13: Eliminar Viajes con G nero Desconocido y Mostrar el N mero de Viajes Restantes.

En este ejercicio, se requiere eliminar todos los viajes de la colecci n `PMGA` donde el g nero del usuario es desconocido (es decir, donde el campo `Gender` tiene el valor `0`), y luego mostrar el n mero de viajes que no han sido eliminados.



```
mongosh mongodb://127.0.0.1:27017/citibike> db.PMGA.deleteMany({ "Gender": 0 })  
{ acknowledged: true, deletedCount: 1380 }  
citibike>
```

El c digo elimina todos los viajes de la colecci n `PMGA` donde el g nero del usuario es desconocido (`Gender: 0`), y el resultado muestra que se han eliminado 1380 documentos correctamente.

### Ejercicio 14: Calcular la Duraci n Promedio de los Viajes.

En este ejercicio, vamos a calcular la duraci n promedio de los viajes en la colecci n `PMGA`. Utilizaremos el comando de agregaci n en MongoDB para realizar este c lculo.



```
mongosh mongodb://127.0.0.1:27017/citibike> db.PMGA.aggregate([
...   { $group: { _id: null, averageDuration: { $avg: "$Trip Duration" } } }
... ])
[ { _id: null, averageDuration: 825.0538854225751 } ]
```

El código calcula la duración promedio de los viajes en la colección `PMGA`, y el resultado muestra que la duración promedio es de 825.05 segundos.

### Ejercicio 15: Contar el Número de Viajes por Tipo de Usuario.

En este ejercicio, vamos a contar el número de viajes por tipo de usuario en la colección `PMGA`. Utilizaremos un comando de agregación en MongoDB para agrupar los viajes por el campo `user Type` y contar cuántos viajes tiene cada tipo de usuario.

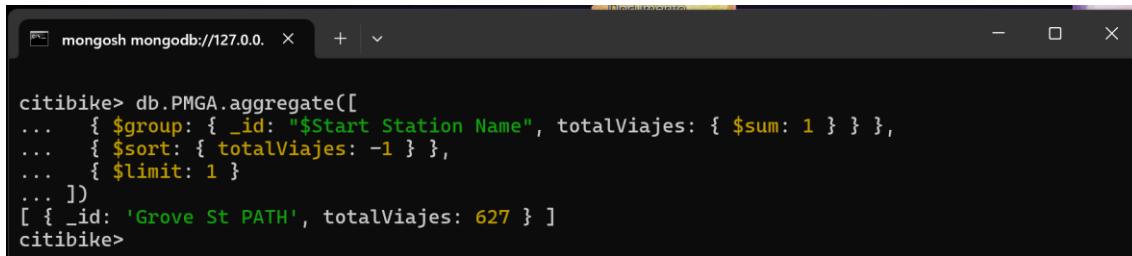


```
mongosh mongodb://127.0.0.1:27017/citibike> db.PMGA.aggregate([
...   { $group: { _id: "$User Type", totalViajes: { $sum: 1 } } }
... ])
[ { _id: 'Subscriber', totalViajes: 5289 } ]
```

El código agrupa los viajes por tipo de usuario (`user Type`) y cuenta cuántos viajes hay de cada tipo, mostrando que hay **5289 viajes** realizados por usuarios de tipo "**Subscriber**".

### Ejercicio 16: Mostrar el Nombre de la Estación más Popular como Estación de Salida.

En este ejercicio, vamos a encontrar la estación de salida que ha tenido el mayor número de viajes. Para ello, utilizaremos el comando de agregación de MongoDB para contar cuántos viajes han comenzado en cada estación de salida y luego ordenar los resultados para mostrar la estación más popular.

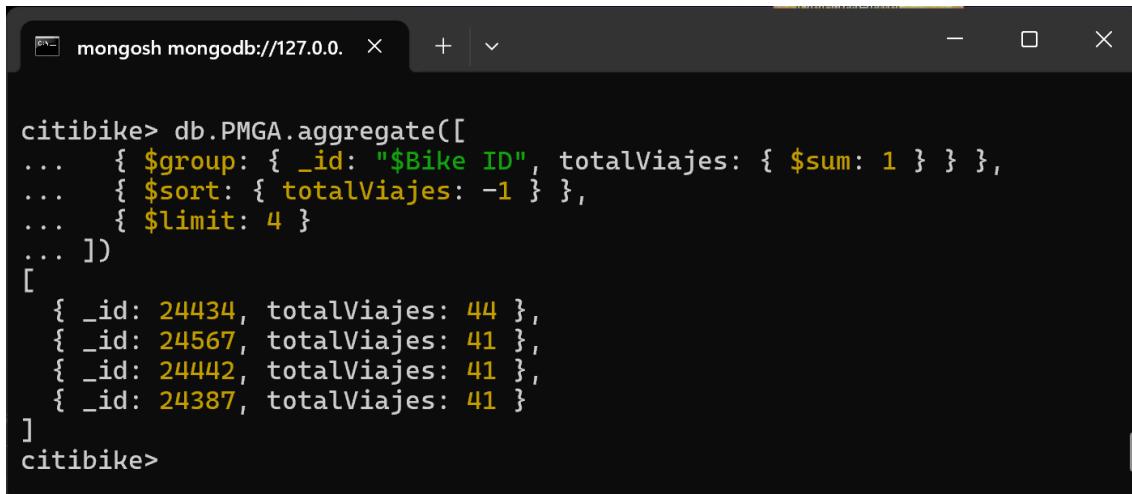


```
mongosh mongodb://127.0.0.1:27017/citibike> db.PMGA.aggregate([
...   { $group: { _id: "$Start Station Name", totalViajes: { $sum: 1 } } },
...   { $sort: { totalViajes: -1 } },
...   { $limit: 1 }
... ])
[ { _id: 'Grove St PATH', totalViajes: 627 } ]
citibike>
```

El código agrupa los viajes por el nombre de la estación de salida, cuenta el número de viajes por estación, ordena las estaciones por el número de viajes de mayor a menor, y muestra la estación "**Grove St PATH**" como la más popular, con **627 viajes** iniciados allí.

### Ejercicio 17: Obtener los Identificadores de las 4 Bicicletas Más Usadas.

En este ejercicio, vamos a obtener los identificadores de las 4 bicicletas que han sido utilizadas en la mayor cantidad de viajes. Utilizaremos el comando de agregación para contar cuántos viajes se han realizado con cada bicicleta y luego ordenaremos los resultados para mostrar las 4 bicicletas más usadas.

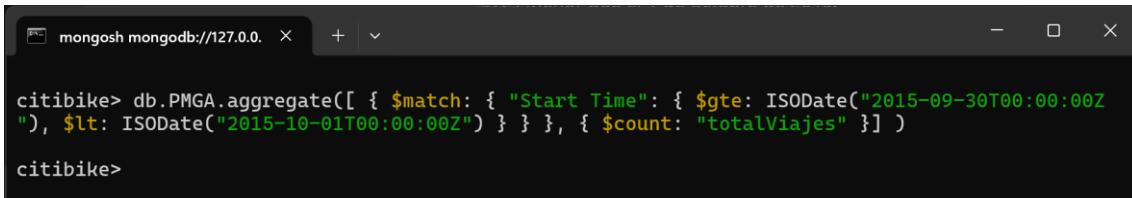


```
mongosh mongodb://127.0.0.1:27017/citibike> db.PMGA.aggregate([
...   { $group: { _id: "$Bike ID", totalViajes: { $sum: 1 } } },
...   { $sort: { totalViajes: -1 } },
...   { $limit: 4 }
... ])
[ { _id: 24434, totalViajes: 44 },
  { _id: 24567, totalViajes: 41 },
  { _id: 24442, totalViajes: 41 },
  { _id: 24387, totalViajes: 41 } ]
citibike>
```

El código agrupa los viajes por el identificador de la bicicleta (`Bike ID`), cuenta cuántos viajes se han realizado con cada bicicleta, ordena las bicicletas de mayor a menor según el número de viajes, y muestra las 4 bicicletas más usadas, con los identificadores **24434**, **24567**, **24442** y **24387**, que han sido usadas 44 y 41 veces respectivamente.

## Ejercicio 18: Obtener el Número de Viajes Realizados el 30 de septiembre de 2015.

En este ejercicio, vamos a contar el número de viajes realizados el 30 de septiembre de 2015. Para hacerlo, utilizaremos el comando de agregación de MongoDB para filtrar los documentos que correspondan a esa fecha y luego contar cuántos viajes se realizaron en ese día.

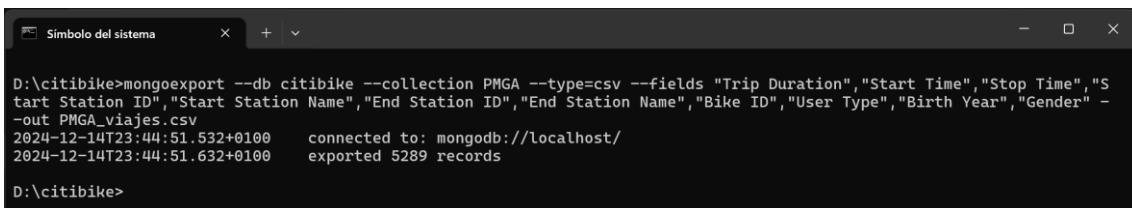


```
mongosh mongodb://127.0.0.1:27017/citibike> db.PMGA.aggregate([ { $match: { "Start Time": { $gte: ISODate("2015-09-30T00:00:00Z"), $lt: ISODate("2015-10-01T00:00:00Z") } } }, { $count: "totalViajes" } ])
```

El código filtra los viajes realizados el 30 de septiembre de 2015 (desde las 00:00:00 hasta las 23:59:59) y luego calcula la cantidad de viajes realizados en ese día, devolviendo el número total de viajes en el campo **totalViajes**. No se ha obtenido ningún resultado porque no existen registros que coincidan con esa condición.

## 4.- Exportación de la base de datos a formato csv.

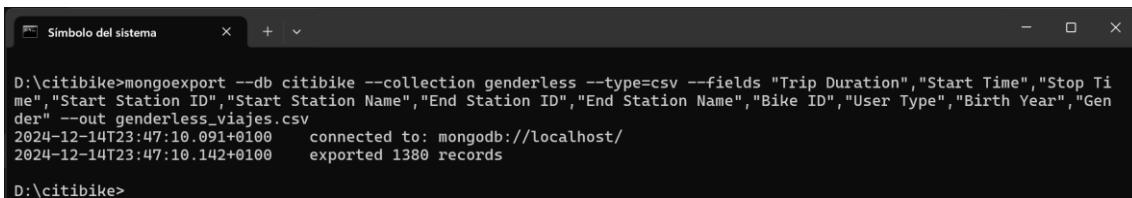
### 4.1.- Explotación de la base de datos en un archivo PMGA\_viajes.csv.



```
D:\citibike>mongoexport --db citibike --collection PMGA --type=csv --fields "Trip Duration","Start Time","Stop Time","Start Station ID","Start Station Name","End Station ID","End Station Name","Bike ID","User Type","Birth Year","Gender" -out PMGA_viajes.csv
2024-12-14T23:44:51.532+0100      connected to: mongodb://localhost/
2024-12-14T23:44:51.632+0100      exported 5289 records
D:\citibike>
```

Este código exporta los datos de la colección **PMGA** de la base de datos **citibike** a un archivo **CSV** llamado **PMGA\_viajes.csv**, incluyendo únicamente los campos especificados: **"Trip Duration"**, **"Start Time"**, **"Stop Time"**, **"Start Station ID"**, **"Start Station Name"**, **"End Station ID"**, **"End Station Name"**, **"Bike ID"**, **"User Type"**, **"Birth Year"**, y **"Gender"**. El comando muestra que se han exportado **5289 registros**.

#### 4.2.- Exportación de la base de datos en un archivo genderless\_viajes.csv.



```
D:\citibike>mongoexport --db citibike --collection genderless --type=csv --fields "Trip Duration","Start Time","Stop Ti me","Start Station ID","Start Station Name","End Station ID","End Station Name","Bike ID","User Type","Birth Year","Gen der" --out genderless_viajes.csv
2024-12-14T23:47:10.091+0100    connected to: mongodb://localhost/
2024-12-14T23:47:10.142+0100    exported 1380 records
D:\citibike>
```

Este código exporta los datos de la colección `genderless` de la base de datos `citibike` a un archivo **CSV** llamado `genderless_viajes.csv`, incluyendo los campos especificados: "**Trip Duration**", "**Start Time**", "**Stop Time**", "**Start Station ID**", "**Start Station Name**", "**End Station ID**", "**End Station Name**", "**Bike ID**", "**User Type**", "**Birth Year**", y "**Gender**". El comando muestra que se han exportado **1380 registros**.

### 5.- Conclusiones

En este apartado se abordan los resultados obtenidos durante el análisis y la gestión de los datos, así como las limitaciones que se encontraron a lo largo del proceso y las recomendaciones para mejorar en futuros análisis.

#### 5.1.- Análisis del Proceso y Resultados

A lo largo del proceso, hemos realizado varias tareas clave para gestionar y analizar los datos relacionados con los viajes de bicicletas en la ciudad. En primer lugar, filtramos los viajes donde el género de los usuarios era desconocido, creando una nueva colección `genderless` que permitió analizar los viajes sin información de género. Este proceso nos ayudó a entender la cantidad de datos con información incompleta y a hacer una limpieza más eficiente.

- **Creación de la colección "genderless":** Se pudo identificar cuántos viajes no tenían información sobre el género del usuario, lo que resultó en una cantidad significativa de registros, lo cual es valioso para conocer la extensión de los datos incompletos en la base de datos.
- **Eliminación de los datos con género desconocido:** A través de la eliminación de estos registros en la colección `PMGA`, logramos limpiar la base de datos de datos incompletos, lo que facilita el

análisis de los viajes de usuarios con información de género precisa. Al contar los documentos restantes en la colección después de la eliminación, confirmamos que la operación fue exitosa.

- **Resultados del análisis:** Los análisis realizados sobre las colecciones (por ejemplo, contar los viajes por tipo de usuario, calcular la duración promedio de los viajes, o encontrar la estación de salida más popular) han proporcionado información clave sobre el comportamiento de los usuarios de bicicletas. Los resultados reflejan patrones importantes, como el tipo de usuario más frecuente y las estaciones más utilizadas.

El proceso de filtrado y análisis fue efectivo para generar insights sobre los datos de los viajes, pero también dejó en evidencia áreas de mejora, especialmente en la calidad de los datos relacionados con el género de los usuarios.

## 5.2.- Limitaciones y Recomendaciones.

A pesar de los buenos resultados obtenidos, el proceso de gestión y análisis de datos presentó algunas limitaciones que deben ser consideradas para futuros análisis.

Limitaciones:

1. **Datos incompletos:** Aunque se eliminaron los registros con género desconocido, la falta de datos completos en algunas partes de la base de datos puede influir en la calidad de los análisis. Es importante tener en cuenta que los registros incompletos pueden afectar los resultados, especialmente cuando se hace un análisis segmentado por características como el género o la edad de los usuarios.
2. **Dependencia de la exactitud de las fechas:** En algunos casos, la precisión de las fechas y horas de los viajes podría no ser completamente fiable debido a posibles errores de registro en el sistema de captura de datos. Este tipo de inconsistencia podría afectar el análisis de viajes realizados en días específicos o durante períodos de tiempo muy cortos.
3. **Limitación en el análisis de género:** Aunque la eliminación de los registros con género desconocido mejoró la limpieza de los datos, el análisis del género está limitado a los registros donde esta información está presente. Esto implica que cualquier

análisis realizado sobre el género de los usuarios podría no ser completamente representativo si los datos faltantes son sustanciales.

#### 5.2.1.- Recomendaciones

1. **Mejorar la recopilación de datos:** Se recomienda implementar mecanismos para garantizar que los datos de género y otras variables clave sean capturados de manera más consistente en el futuro. Esto incluiría mejorar los formularios de entrada de datos o utilizar procesos automáticos para verificar la integridad de los registros.
2. **Ampliar el análisis de otros factores:** Sería útil expandir los análisis para incluir otros factores relevantes, como la duración de los viajes por tipo de usuario, el análisis de tendencias estacionales o la evaluación de la disponibilidad y el uso de las bicicletas en diferentes horarios del día.
3. **Automatización del proceso de limpieza de datos:** Se recomienda crear procesos automatizados para detectar y eliminar o corregir registros incompletos o inconsistentes. Esto podría incluir la implementación de validaciones en tiempo real para asegurar que los datos ingresados sean completos y correctos.
4. **Análisis en tiempo real:** Se sugiere considerar la implementación de herramientas de análisis en tiempo real que permitan la monitorización continua de los datos y la detección de patrones a medida que se vayan recopilando nuevos registros.

#### 5.2.2.- Resumen Final

El proceso de gestión de datos ha sido exitoso en términos de filtrado, limpieza y análisis, aunque las limitaciones relacionadas con la calidad de los datos y la precisión de las fechas deben ser consideradas en el futuro. Las recomendaciones para mejorar la calidad de los datos y ampliar los análisis permitirán realizar estudios más completos y precisos en futuras investigaciones sobre el uso de bicicletas.

## 6.- Bibliografía

La bibliografía es una parte fundamental para cualquier trabajo de investigación, ya que proporciona las fuentes y referencias que sustentan el proceso y los resultados obtenidos. A continuación, se incluyen las principales fuentes consultadas y utilizadas durante este trabajo:

### 6.1.- Libros y Textos Académicos

1. **Cormen, T., Leiserson, C., Rivest, R., & Stein, C.** (2009). *Introduction to Algorithms* (3rd ed.). MIT Press. Este libro es una referencia fundamental sobre algoritmos y estructuras de datos, proporcionando una base sólida sobre cómo se gestionan y procesan los datos, lo cual es aplicable en el contexto del análisis de grandes volúmenes de información como el realizado en este trabajo.
2. **Russell, S., & Norvig, P.** (2020). *Artificial Intelligence: A Modern Approach* (4th ed.). Pearson Education. Aporta conceptos de inteligencia artificial, aprendizaje automático y procesamiento de datos, que son útiles para el análisis de patrones en datos como los de este estudio sobre el uso de bicicletas.

### 6.2.- Documentación Técnica

3. **MongoDB Documentation** (n.d.). *MongoDB Manual*. MongoDB, Inc.  
Disponible en: <https://docs.mongodb.com>

La documentación oficial de MongoDB es una fuente esencial para aprender sobre el uso de comandos de agregación, la creación de colecciones y la gestión de datos en bases de datos NoSQL, temas centrales en este trabajo.

### 6.3.- Artículos y Recursos en Línea

4. **MongoDB University** (n.d.). *MongoDB for Developers*. MongoDB, Inc.  
Disponible en: <https://university.mongodb.com>

Este recurso en línea proporciona formación sobre MongoDB, cubriendo desde los conceptos básicos hasta técnicas avanzadas, que son cruciales para comprender cómo trabajar

con bases de datos NoSQL y ejecutar consultas y operaciones en MongoDB.

5. **Wikipedia contributors**. (2024). *MongoDB*. Wikipedia. Disponible en: <https://es.wikipedia.org/wiki/MongoDB>

Un artículo introductorio sobre MongoDB, explicando sus conceptos fundamentales y su uso generalizado en sistemas de bases de datos.

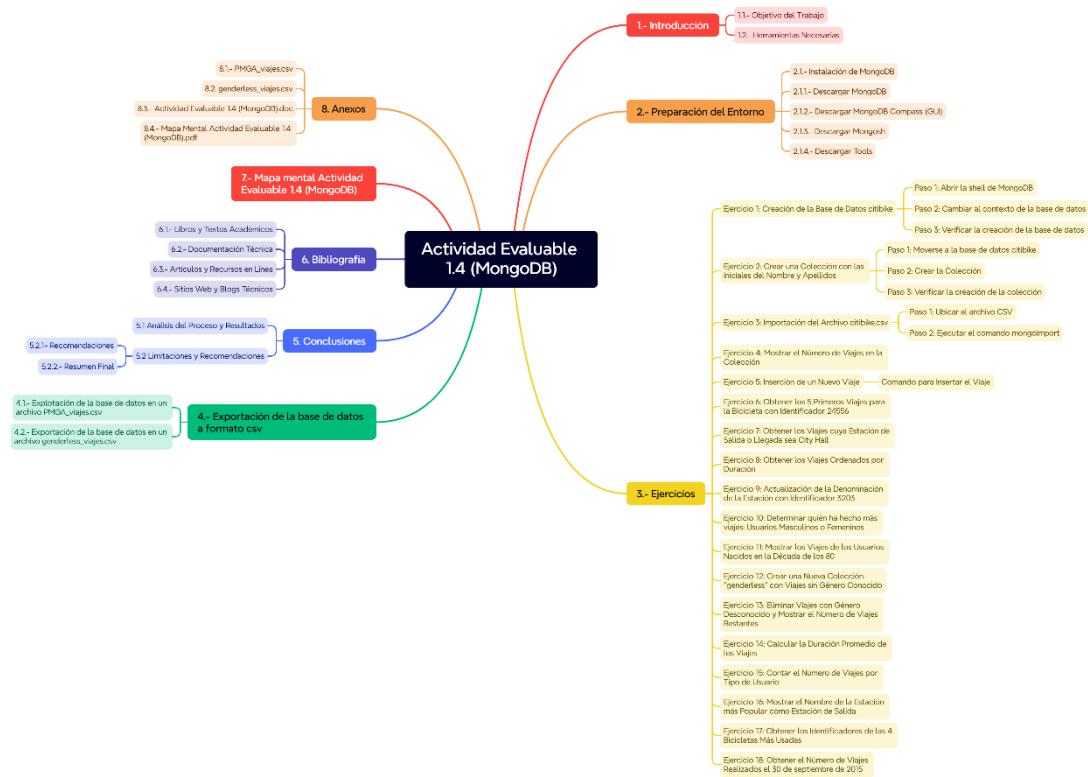
#### 6.4.- Sitios Web y Blogs Técnicos

6. **TutorialsPoint** (n.d.). *MongoDB Tutorial*. TutorialsPoint. Disponible en: <https://www.tutorialspoint.com/mongodb> Este sitio ofrece tutoriales y guías sobre MongoDB, que ayudan a comprender cómo realizar operaciones como el filtrado, agrupamiento y análisis de datos en colecciones de MongoDB.

7. **Stack Overflow** (n.d.). *MongoDB Queries*. Stack Overflow. Disponible en: <https://stackoverflow.com>

Una comunidad en línea donde se resuelven dudas sobre consultas en MongoDB, proporcionando soluciones prácticas a problemas comunes durante el análisis y manipulación de datos en bases de datos NoSQL.

## 7.- Mapa mental Actividad Evaluable 1.4 (MongoDB)



Resume el proceso de trabajo con MongoDB, abarcando desde la instalación y configuración del entorno hasta la ejecución de ejercicios prácticos relacionados con la creación, manipulación y exportación de bases de datos de viajes de bicicletas.

## 8.- Anexos

### 8.1.- PMGA\_viajes.csv

Este archivo contiene los datos exportados de la colección `PMGA` de la base de datos `citibike`. Incluye los siguientes campos: **Trip Duration, Start Time, Stop Time, Start Station ID, Start Station Name, End Station ID, End Station Name, Bike ID, User Type, Birth Year** y **Gender**. El archivo fue generado mediante el comando `mongoexport` para analizar los viajes realizados, filtrados por los campos especificados. Este archivo sirve como un respaldo de la colección `PMGA` en formato CSV, que puede ser utilizado para análisis posteriores o para la verificación de los resultados obtenidos en el trabajo.

## 8.2.- **genderless\_viajes.csv**

Este archivo contiene los registros de la colección `genderless`, que incluye los viajes realizados por usuarios con género desconocido. Los datos fueron filtrados de la colección original `PMGA` y exportados a este archivo CSV para realizar un análisis de los viajes sin datos de género. Este archivo es crucial para realizar el análisis sobre los viajes con género desconocido, los cuales fueron eliminados posteriormente de la colección `PMGA`.

## 8.3.- **Actividad Evaluable 1.4 (MongoDB).pdf**

Este archivo en formato `.doc` contiene la documentación completa de la **Actividad Evaluable 1.4** sobre **MongoDB**. Incluye detalles sobre el proceso de trabajo realizado, los resultados obtenidos, los comandos ejecutados y las conclusiones finales. Este documento sirve como un reporte formal del trabajo realizado durante la actividad evaluable. Este archivo proporciona una descripción detallada del análisis de los datos en MongoDB y se entrega como parte del trabajo final de la actividad evaluable.

## 8.4.- **Mapa Mental Actividad Evaluable 1.4 (MongoDB).pdf**

Este archivo **PDF** contiene un mapa mental que resume de manera gráfica los pasos, conceptos y herramientas utilizados en la **Actividad Evaluable 1.4** de **MongoDB**. El mapa mental ofrece una visión clara y concisa del proceso de trabajo, ayudando a visualizar la conexión entre los distintos pasos del análisis y los comandos utilizados. Este archivo sirve como una representación visual del flujo de trabajo realizado en la actividad, permitiendo una comprensión rápida de los procesos y las relaciones entre los conceptos clave.

## Índice Alfabético

---

### A

abre ..... 14, 21  
acceder ..... 14, 22, 24  
actividad ..... 42  
actividades ..... 4, 5  
actual ..... 23  
actualización ..... 4, 29, 30  
actualizar ..... 29  
afectar ..... 37  
agregación ..... 30, 32, 33, 34, 35, 39  
agrupa ..... 33, 34  
agrupar ..... 30, 33  
almacenar ..... 4, 23  
ampliar ..... 38  
análisis ..... 4, 30, 36, 37, 38, 39, 40, 41, 42  
analizar ..... 4, 36, 41  
añadir ..... 19, 20  
año ..... 5, 26, 27, 28, 29, 31  
aparecerá ..... 7, 22  
archivo ..... 2, 3, 4, 11, 13, 16, 23, 35, 36, 41, 42  
artificial ..... 39  
así ..... 20, 25, 26, 36  
aunque ..... 38  
avanzada ..... 18, 27  
avanzadas ..... 11, 12, 39

---

### B

base ..... 2, 3, 4, 20, 21, 22, 23, 24, 26, 35, 36, 37, 39, 41  
bases ..... 4, 5, 6, 10, 11, 15, 21, 39, 40, 41  
bibliografía ..... 39  
bicicleta ..... 25, 26, 27, 28, 29, 31, 34  
bicicletas ..... 4, 5, 34, 36, 37, 38, 39, 41  
bin ..... 18, 19, 20  
botón ..... 6, 7, 8, 9, 10, 13, 14, 16, 17, 18, 19, 20  
busca ..... 4, 31

---

### C

cada ..... 27, 30, 33, 34  
calcula ..... 33, 35  
calcular ..... 4, 32, 37  
calidad ..... 37, 38  
cambia ..... 22  
campo ..... 30, 31, 32, 33, 35  
campos ..... 27, 28, 35, 36, 41  
cantidad ..... 30, 34, 35, 36  
capacidades ..... 11, 20  
características ..... 20, 37  
casos ..... 29, 37  
citibike ..... 2, 21, 22, 23, 24, 26, 35, 36, 41  
ciudad ..... 4, 36  
clave ..... 36, 37, 38, 42  
clic ..... 7, 9, 10, 13, 14, 16, 17, 18, 19  
código ..... 12, 26, 27, 28, 30, 32, 33, 34, 35, 36  
colección ..... 2, 4, 22, 23, 24, 25, 26, 27, 28, 30, 31, 32, 33, 35, 36, 41, 42  
colecciones ..... 21, 23, 37, 39, 40  
com ..... 5, 10, 15, 39, 40  
comando2, 4, 21, 22, 23, 24, 25, 30, 31, 32, 33, 34, 35, 36, 41  
comandos ..... 4, 5, 6, 12, 15, 18, 21, 39, 42  
comenzar ..... 14, 20  
community ..... 5  
cómo ..... 39, 40  
compass ..... 10, 11, 12  
completa ..... 11, 23, 42  
completamente ..... 5, 37, 38  
completos ..... 37, 38  
comprender ..... 39, 40  
comunitaria ..... 4, 5  
conceptos ..... 39, 40, 42  
conclusiones ..... 42  
configuración ..... 8, 9, 13, 17, 18, 20, 41  
conjunto ..... 4, 15  
conocer ..... 24, 36  
consideradas ..... 37, 38  
consola ..... 5, 21

---

consulta ..... 4, 11, 25, 26, 27, 28  
consultas ..... 4, 21, 40  
contar ..... 30, 33, 34, 35, 37  
contexto ..... 2, 21, 22, 39  
contiene ..... 41, 42  
continuación ..... 5, 24, 39  
continuar ..... 7, 9, 16, 17  
correctamente ..... 5, 14, 23, 32  
creación ..... 2, 4, 22, 23, 39, 41  
creada ..... 22, 24, 32  
crear ..... 23, 31, 38  
csv ..... 2, 3, 23, 24, 35, 36, 41, 42  
cuálquier ..... 18, 37, 39  
cuántos ..... 30, 33, 34, 35, 36  
cuenta ..... 33, 34, 37

---

### D

database ..... 15, 16  
dataset ..... 4  
datos ..... 2, 3, 4, 5, 6, 10, 11, 12, 15, 20, 21, 22, 23, 24, 25, 26, 27, 35, 36, 37, 38, 39, 40, 41, 42  
debe ..... 7, 29  
deben ..... 28, 37, 38  
decir ..... 28, 32  
dentro ..... 22, 24  
desarrollar ..... 4, 5  
descargar ..... 10, 12, 15  
descendente ..... 28, 30  
desconocido ..... 26, 29, 30, 32, 36, 37, 42  
después ..... 37  
destino ..... 5, 27, 29  
detallada ..... 25, 26, 28, 42  
detalles ..... 5, 26, 29, 31, 42  
día ..... 35, 38  
diferentes ..... 4, 38  
disponible ..... 11  
documentación ..... 39, 42  
documento ..... 25, 42  
documentos ..... 23, 24, 30, 32, 35, 37  
download ..... 5, 10, 15  
duración ..... 5, 25, 26, 27, 28, 29, 31, 32, 33, 37, 38

---

### E

eficiente ..... 5, 15, 20, 36  
ejecución ..... 6, 11, 41  
ejecuta ..... 14, 24, 25, 26, 27, 28, 30  
ejecutar ..... 4, 5, 9, 12, 21, 22, 24, 40  
ejemplo ..... 31, 37  
ejercicio ..... 24, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35  
ejercicios ..... 4, 41  
eliminación ..... 36, 37  
eliminados ..... 32, 42  
eliminar ..... 32, 38  
encontrar ..... 33, 37  
entorno ..... 5, 18, 19, 20, 21, 41  
especialmente ..... 37  
especificados ..... 35, 36, 41  
específicos ..... 4, 26, 27, 37  
estación ..... 27, 29, 30, 33, 34, 37  
estaciones ..... 5, 25, 26, 31, 34, 37  
evaluable ..... 42  
exe ..... 5, 11  
explorar ..... 10, 20  
exporta ..... 35, 36  
exportación ..... 41  
exportado ..... 35, 36  
exportados ..... 41, 42  
exportar ..... 4, 15  
extensión ..... 28, 36

---

### F

factores ..... 38  
fechas ..... 37, 38  
femeninos ..... 26, 30  
filtra ..... 32, 35

---

---

filtrado .....	37, 38, 40
filtrados .....	41, 42
filtrando.....	26, 31
fin.....	26, 31
final .....	42
finalizamos .....	14, 18
find.....	25, 26, 27, 28, 31
forma.....	5, 10
formato .....	3, 4, 6, 35, 41, 42
fuentes.....	39
funcionalidades.....	12, 20
fundamental.....	5, 6, 20, 39
futuro.....	38
futuros .....	36, 37

---

## G

garantizar .....	20, 38
genderless .....	2, 3, 31, 32, 36, 42
general.....	5
género .....	5, 26, 27, 28, 29, 30, 31, 32, 36, 37, 38, 42
gestión .....	36, 37, 38, 39
gestionar .....	4, 11, 15, 21, 36
gráfica.....	10, 11, 42
grandes .....	4, 5, 39

## H

habilidades .....	4, 5
hace.....	9, 10, 37
hacemos.....	13, 16, 17
hacer .....	7, 10, 19, 36
haciendo.....	14, 18
herramienta .....	11
herramientas.....	4, 5, 13, 15, 18, 19, 20, 38, 42
https.....	5, 10, 15, 39, 40

## I

identificador .....	26, 27, 28, 29, 31, 34
identificadores.....	5, 34
identificar .....	4, 27, 36
implementación .....	38
implementar .....	25, 38
importación.....	23
importar .....	4, 15, 24
incluir.....	6, 38
incluye .....	25, 28, 42
incluyendo .....	4, 5, 26, 27, 31, 35, 36
incompletos .....	36, 37, 38
información.....	4, 5, 25, 26, 28, 36, 37, 39
iniciales .....	5
iniciamos .....	6
iniciar.....	21
inicio .....	25, 26, 31
inscripción.....	25
insertar .....	24, 25
instalación .....	5, 6, 8, 9, 10, 11, 13, 14, 16, 17, 18, 20, 41
instalar.....	13
install .....	9
instrucción .....	24, 26, 27
interactiva.....	14, 21
interactuar.....	4, 21
interfaz.....	6, 10, 11, 12
intuitiva.....	10, 11

## L

largo .....	36
licencia.....	7, 16
limit .....	25, 26
limitaciones.....	36, 37, 38
limpieza.....	36, 37, 38
línea.....	4, 12, 15, 21, 39, 40
lista .....	22, 23, 26, 29

## LI

llamada .....	23, 31, 32
---------------	------------

---

llamado .....	35, 36
llegada .....	25, 27

---

## L

luego.....	32, 33, 34, 35
lugar .....	30, 36

## M

manera.....	11, 15, 20, 38, 42
manipulación .....	4, 40, 41
manipular .....	10, 15
mantenemos .....	13, 17
mapa.....	42
masculinos .....	26, 30
mayor.....	28, 33, 34
mayoría .....	29, 30
mediante .....	6, 41
mejorar .....	36, 38
menor .....	30, 34
mental.....	3, 41, 42
método .....	25, 30
mongodb .....	5, 10, 11, 15, 16, 39, 40
mongoimport .....	2, 4, 24
mongosh.....	6, 13, 14, 21, 24
mostrar .....	5, 27, 28, 31, 32, 33, 34
msi.....	13, 16
muestra .....	26, 29, 31, 32, 33, 34, 35, 36

## N

nacidos .....	31
nacimiento .....	5, 26, 27, 28, 29, 31
necesarias.....	5
necesario .....	4, 20
next .....	7
nombre .....	29, 30, 34
nueva .....	23, 31, 32, 36
nuevo .....	24, 25, 26
número.....	24, 26, 30, 32, 33, 34, 35

## O

objetivo .....	4
obtener .....	24, 26, 27, 28, 34
obtenidos .....	4, 36, 37, 39, 41, 42
oficial .....	5, 10, 12, 15, 39
ofrece .....	20, 40, 42
opción .....	7, 8
operación .....	25, 26, 37
operaciones .....	21, 22, 30, 40
operativo .....	5, 21
orden .....	28, 30
ordena .....	30, 34
ordenados .....	28
origen .....	5, 29

## P

página .....	5, 10, 12, 15
pantalla .....	7, 8, 9, 10, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25
parte .....	39, 42
partida .....	25, 27
pasos .....	5, 42
patrones .....	4, 37, 38, 39
pdf .....	3, 42
permite .....	10, 12, 22
permitiendo .....	26, 42
permítiéndote .....	21, 22
permitiría .....	4, 11, 18, 20
podría .....	37, 38
popular .....	33, 34, 37
posteriormente .....	23, 24, 42
precisión .....	37, 38
precisos .....	25, 38
predeterminada .....	8, 9, 13, 17
presionamos .....	21, 23

primero ..... 28  
primeros ..... 26  
procede ..... 8, 9, 10, 12, 15, 16, 22, 24, 25  
procedemos ..... 7, 10, 14, 17, 21, 23  
proceder ..... 11, 23  
procederemos ..... 18, 19  
procedido ..... 20, 22  
proceso ..... 5, 6, 8, 9, 10, 14, 17, 20, 24, 36, 37, 38, 39, 41, 42  
procesos ..... 38, 42  
promedio ..... 32, 33, 37  
proporciona ..... 5, 39, 42  
proporcionando ..... 28, 39, 40  
puede ..... 37, 41  
pulsamos ..... 7  
pulsando ..... 9, 20  
punto ..... 27

---

**R**

real 11, 38  
realiza ..... 5, 30  
realizado ..... 14, 30, 34, 36, 38, 39, 42  
realizados ..... 5, 26, 29, 30, 31, 33, 35, 37, 41, 42  
realizar ..... 4, 5, 21, 22, 32, 38, 40, 42  
realizó ..... 28, 31  
reciente ..... 5, 26  
recomendaciones ..... 36, 38  
recomienda ..... 38  
recursos ..... 4  
registrados ..... 24, 26  
registro ..... 25, 26, 37  
registros ..... 26, 28, 35, 36, 37, 38, 42  
relacionados ..... 36, 37, 41  
rendimiento ..... 11, 20  
representa ..... 24, 25  
requiere ..... 4, 24, 26, 27, 28, 29, 31, 32  
restantes ..... 37  
resultado ..... 26, 29, 30, 31, 32, 33, 35  
resultados ..... 4, 5, 30, 33, 34, 36, 37, 39, 41, 42  
resume ..... 42  
resumen ..... 20  
ruta ..... 18, 20, 23

---

**S**

salida ..... 27, 33, 34, 37  
según ..... 28, 30, 34  
segundos ..... 31, 33  
septiembre ..... 2, 35  
ser ..... 4, 37, 38, 41  
service ..... 8  
servidor ..... 4, 5  
set ..... 30  
shell ..... 2, 14, 21, 22, 24, 25  
sido ..... 32, 34, 38  
siguiente ..... 21, 24, 25  
siguientes ..... 4, 28, 41  
sirve ..... 41, 42  
sistema ..... 5, 6, 18, 20, 21, 37  
sistemas ..... 5, 40  
solo ..... 11, 27, 28  
sort 25, 28

---

**T**

tareas ..... 4, 36  
tecla ..... 9, 21, 23, 24  
terminal ..... 5, 14, 21, 23, 24  
términos ..... 16, 38  
tiempo ..... 11, 28, 37, 38  
tipo ..... 26, 27, 28, 29, 31, 33, 37, 38  
tools ..... 15, 16  
total ..... 24, 30, 35  
trabajar ..... 4, 5, 22, 39  
trabajo ..... 4, 5, 20, 22, 39, 41, 42  
través ..... 4, 36  
trayecto ..... 25, 28  
trayectos ..... 26, 27  
try ..... 5, 10, 15  
tutorialspoint ..... 40  
type ..... 24

---

**U**

ubicación ..... 18, 19  
único ..... 27, 28  
university ..... 39  
usadas ..... 34  
use ..... 22, 24  
user ..... 8  
uso ..... 4, 5, 14, 38, 39, 40  
usuario ..... 10, 11, 25, 26, 27, 28, 29, 30, 31, 32, 33, 36, 37, 38  
usuarios ..... 5, 26, 29, 30, 31, 33, 36, 37, 38, 42  
utilidades ..... 4, 15  
utiliza ..... 24, 25, 30  
utilizadas ..... 34, 37, 39  
utilizados ..... 42  
utilizando ..... 4, 22, 23, 24, 25, 30, 31  
utilizar ..... 20, 38  
utilizaremos ..... 33, 35

---

**V**

vacio ..... 26, 29  
valor ..... 32  
vamos ..... 31, 32, 33, 34, 35  
variables ..... 18, 38  
verificar ..... 14, 23, 26, 38  
versión ..... 4, 5, 6, 17  
viaje ..... 5, 24, 25, 26, 27, 28, 29, 31  
viajes 2, 3, 4, 5, 24, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 41, 42  
visualizar ..... 11, 25, 42  
volúmenes ..... 4, 5, 39

---

**W**

wikipedia ..... 40  
windows ..... 16  
www ..... 5, 10, 15, 40