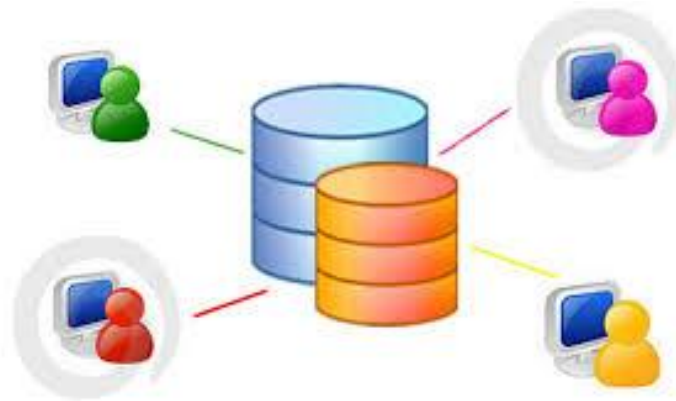


3.1 BASES DE DATOS

Colección de datos, donde los datos están lógicamente relacionados entre sí, tienen una definición y descripción comunes y están estructurados de una forma particular. Una base de datos es también un modelo del mundo real y, como tal, debe poder servir para toda una gama de usos y aplicaciones

3.1.1 Sistema gestor de bases de datos (SGBD)

- El sistema de gestión de la base de datos es una aplicación que permite a los usuarios definir, crear y mantener la base de datos, además de proporcionar un acceso controlado a la misma. Se denomina sistema de bases de datos al conjunto formado por la base de datos, el SGBD y los programas de aplicación que dan servicio a la empresa u organización.



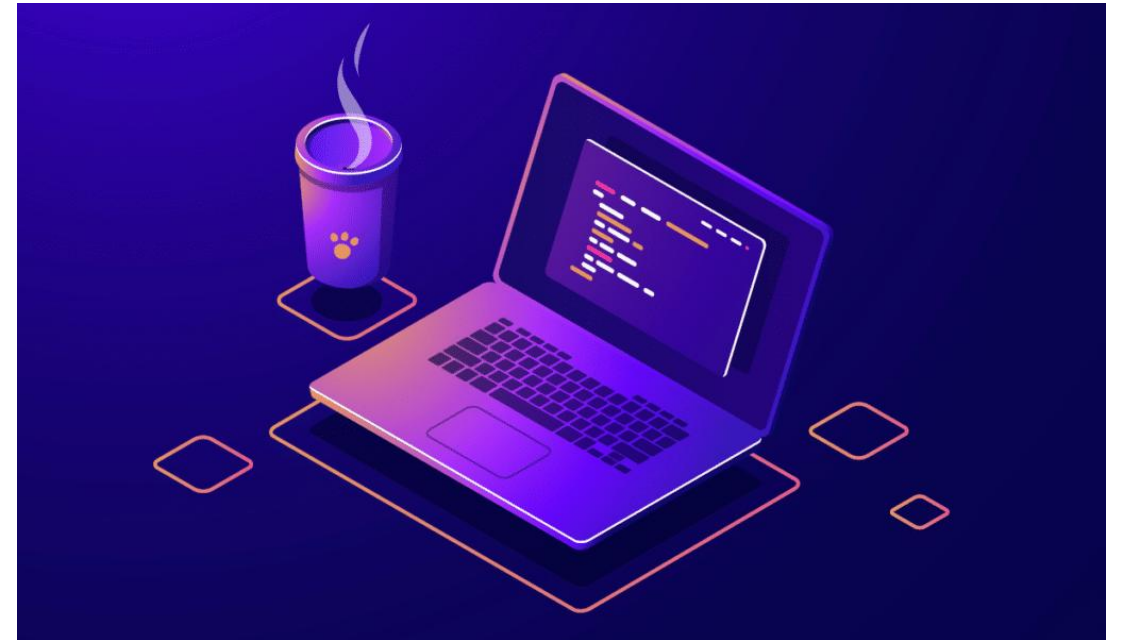
Generalmente, un SGBD proporciona los servicios que se citan a continuación:

- El SGBD permite la definición de la base de datos mediante un lenguaje de definición de datos. Este lenguaje permite especificar la estructura y el tipo de los datos, así como las restricciones sobre los datos.
- Permite la inserción, actualización, eliminación y consulta de datos mediante un lenguaje de manejo de datos.

- El hecho de disponer de un lenguaje para realizar consultas reduce el problema de los sistemas de ficheros, en los que el usuario tiene que trabajar con un conjunto fijo de consultas, o bien, dispone de un gran número de programas de aplicación costosos de gestionar.
- Hay dos tipos de lenguajes de manejo de datos: los procedurales y los no procedurales. Estos dos tipos se distinguen por el modo en que acceden a los datos.



- Los lenguajes procedurales manipulan la base de datos registro a registro.
- En los lenguajes procedurales se especifica qué operaciones se debe realizar para obtener los datos.





- Los lenguajes no procedurales operan sobre conjuntos de registros.
- En los lenguajes no procedurales se especifica qué datos deben obtenerse sin decir cómo hacerlo.
- El lenguaje no procedural más utilizado es el SQL (Structured Query Language) que, de hecho, es un estándar y es el lenguaje de los SGBD relacionales.

El SGBD proporciona un acceso controlado a la base de datos mediante:

- Un sistema de seguridad, de modo que los usuarios no autorizados no puedan acceder a la base de datos.
- Un sistema de integridad que mantiene la integridad y la consistencia de los datos.
- Un sistema de control de concurrencia que permite el acceso compartido a la base de datos.
- Un sistema de control de recuperación que restablece la base de datos después de que se produzca un fallo del hardware o del software.
- Un diccionario de datos o catálogo, accesible por el usuario, que contiene la descripción de los datos de la base de datos.

- A diferencia de los sistemas de ficheros, en los que los programas de aplicación trabajan directamente sobre los ficheros de datos, el SGBD se ocupa de la estructura física de los datos y de su almacenamiento.
- Con esta funcionalidad, el SGBD se convierte en una herramienta de gran utilidad.
- Sin embargo, desde el punto de vista del usuario, se podría discutir que los SGBD han hecho las cosas más complicadas, ya que ahora los usuarios ven más datos de los que realmente quieren o necesitan, puesto que ven la base de datos completa.





- Los SGBD proporcionan un mecanismo de vistas que permite que cada usuario tenga su propia vista o visión de la base de datos.
- El lenguaje de definición de datos permite definir vistas como subconjuntos de la base de datos.
- Todos los SGBD no presentan la misma funcionalidad, depende de cada producto.
- En general, los grandes SGBD multiusuario ofrecen todas las funciones que se acaban de citar e incluso más.

- Los sistemas modernos son conjuntos de programas extremadamente complejos y sofisticados, con millones de líneas de código y con una documentación consistente en varios volúmenes.
- Lo que se pretende es proporcionar un sistema que permita gestionar cualquier tipo de requisitos y que tenga un 100 % de fiabilidad ante cualquier tipo de fallo.
- Los SGBD están en continua evolución, tratando de satisfacer los requisitos de todo tipo de usuarios. Por ejemplo, muchas aplicaciones de hoy en día necesitan almacenar imágenes, vídeo, sonido, etc. Para satisfacer a este mercado, los SGBD deben evolucionar.
- Conforme vaya pasando el tiempo, irán surgiendo nuevos requisitos, por lo que los SGBD nunca permanecerán estáticos.

3.1.2 Tipos de bases de datos

- Los predecesores de los sistemas gestores de bases de datos fueron los sistemas gestores de ficheros o sistemas de archivos tradicionales.



- **1. Archivos tradicionales.** Consiste en almacenar los datos en archivos individuales, exclusivos para cada aplicación particular. En este sistema los datos pueden ser redundantes (repetidos innecesariamente) y la actualización de los archivos es más lenta que en una base de datos.
- **2. Base de datos.** Es un almacenamiento de datos formalmente definido, controlado centralmente para intentar servir a múltiples y diferentes aplicaciones. La base de datos es una fuente de datos que son compartidos por numerosos usuarios para diversas aplicaciones.





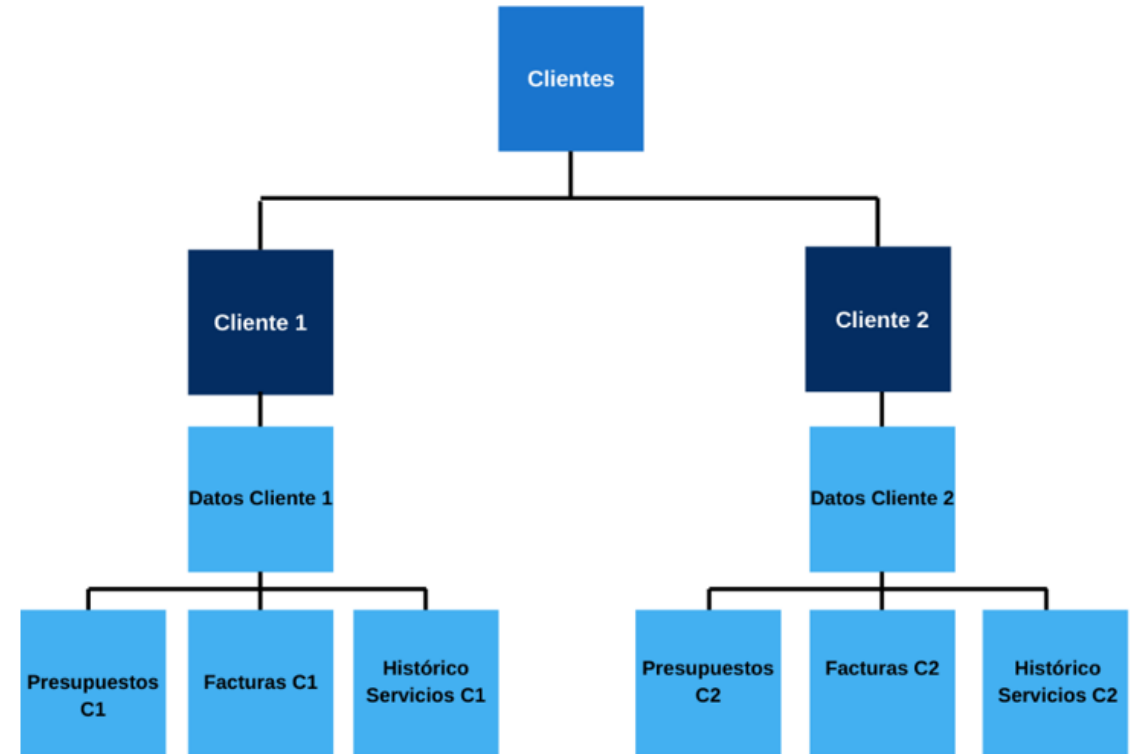
- Así, en un Sistema de archivos tradicional la información está dispersa en varios ficheros de datos y existe un cierto número de programas que los recuperan y agrupan.
- Aunque los sistemas de ficheros o archivos supusieron un gran avance sobre los sistemas manuales, tienen inconvenientes bastante importantes que se solventaron, en gran medida, con la aparición de los sistemas de bases de datos.

Coincidiendo con la evolución histórica de las bases de datos éstas han utilizado distintos modelos:

- Jerárquicos
- En red.
- Relacionales.
- Multidimensionales.

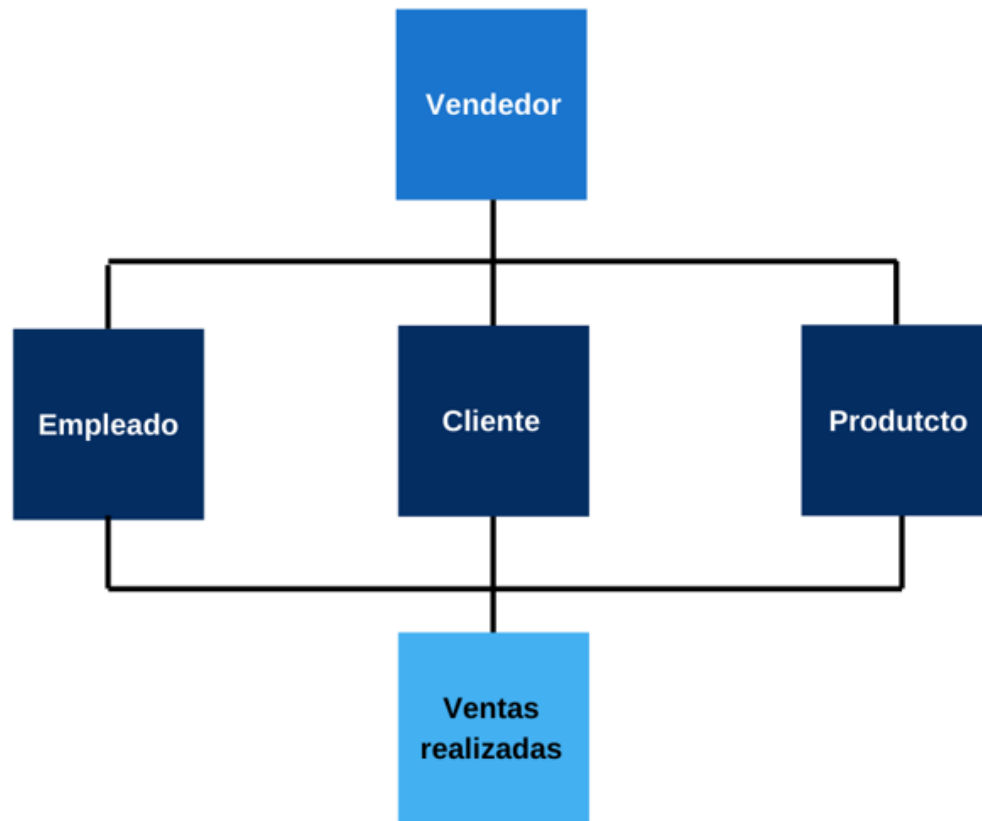
Bases de Datos con estructura jerárquica

- La estructura jerárquica fue usada en las primeras bases de datos.
- Las relaciones entre registros forman una estructura en árbol.
- Actualmente las bases de datos jerárquicas más utilizadas son IMS de IBM y el Registro de Windows de Microsoft.



Base de datos de clientes jerárquica

Bases de Datos con estructura en red

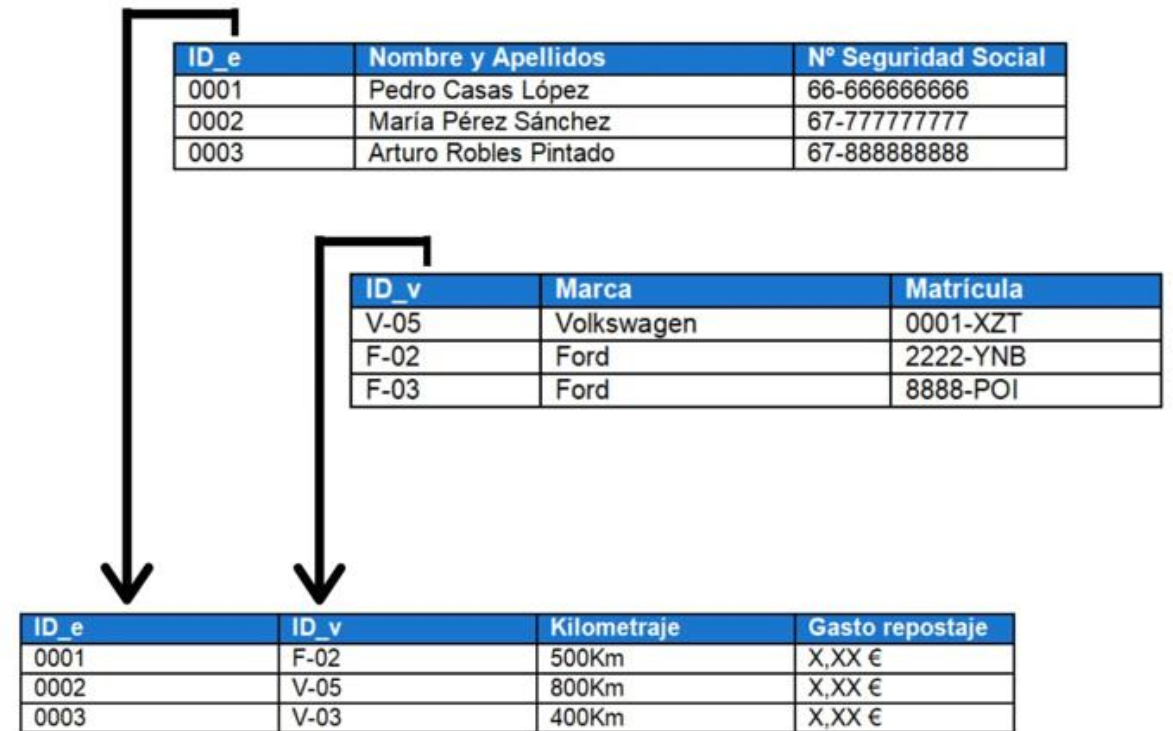


Modelo de base de datos en red

- Esta estructura contiene relaciones más complejas que las jerárquicas.
- Admite relaciones de cada registro con varios que se pueden seguir por distintos caminos.
- El inventor de este modelo fue Charles Bachman, y el estándar fue publicado en 1969 por CODASYL.

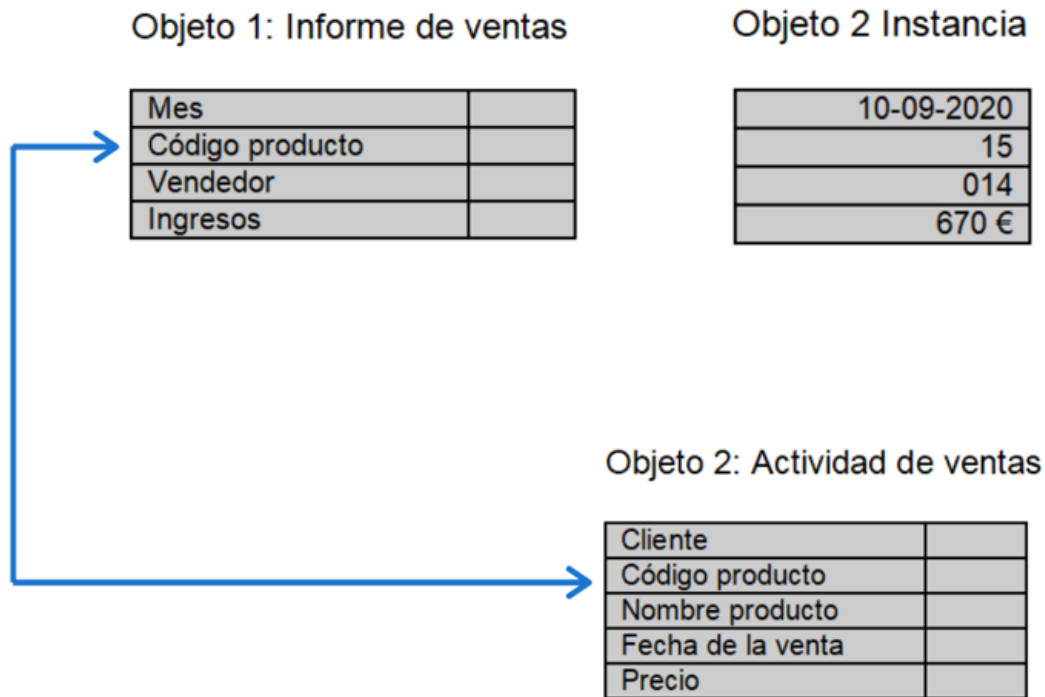
Bases de Datos con estructura relacional

- La estructura relacional es la más extendida hoy en día.
- Almacena los datos en filas o registros (tuplas) y columnas o campos (atributos).
- Estas tablas pueden estar conectadas entre sí por claves comunes.



Modelo de base de datos relacional

Bases de Datos con estructura orientada a objetos



- La estructura orientada a objetos está diseñada siguiendo el paradigma de los lenguajes orientados a objetos.
- De este modo soporta los tipos de datos gráficos, imágenes, voz y texto de manera natural.
- Esta estructura tiene gran difusión en aplicaciones web para aplicaciones multimedia.

3.1.3 Manejadores de bases de datos relacionales (RDBMS)

- Un sistema de gestión de bases de datos relacionales (RDBMS) proporciona una base para numerosas aplicaciones y servicios en una empresa.
- Desde el punto de vista tecnológico, se considera que los RDBMS son antiguos. Sin embargo, la sólida base teórica de la tecnología de bases de datos relacionales ha mantenido su relevancia en un sector volátil.

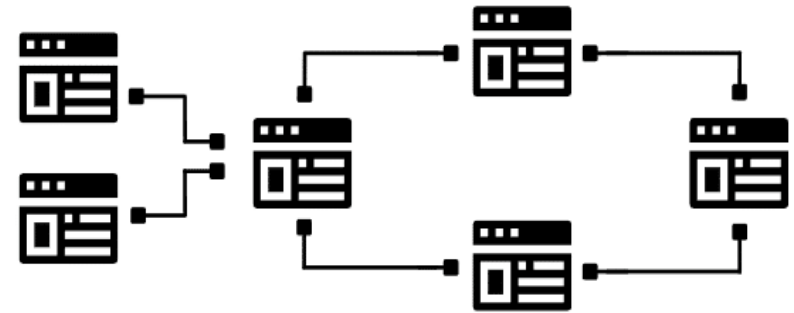




- Esta es la razón por la que incluso las bases de datos no relacionales para almacenar información incluyen funciones similares a RDBMS, como la interfaz SQL en Hadoop.
- Algunos ejemplos populares de sistemas de administración de bases de datos relacionales incluyen **Microsoft SQL Server**, **MySQL** y **mariadb**, que pueden almacenar registros de empleados, registros de clientes y otra información comercial.

- Según el modelo relacional de EF Codd, un RDBMS permite a los usuarios construir, actualizar, administrar e interactuar con una base de datos relacional, lo que permite almacenar datos en forma tabular.
- Por lo tanto, considere RDBMS como un sistema de gestión de datos avanzado que hace que obtener información de los datos sea mucho más fácil. Pero ¿por qué necesitamos una base de datos relacional?

</> BASE DE DATOS RELACIONAL



- Hoy en día, varias empresas utilizan la arquitectura de bases de datos relacionales en lugar de archivos planos o bases de datos jerárquicas para el sistema de gestión de bases de datos (DBMS) de su empresa. Entonces, ¿cuál es la razón para crear una base de datos relacional?
- Una base de datos relacional está diseñada específicamente para manejar de manera eficiente una amplia gama de formatos de datos y consultas de procesos. ¿Y cómo se organizan los datos en un sistema de gestión de bases de datos relacionales?

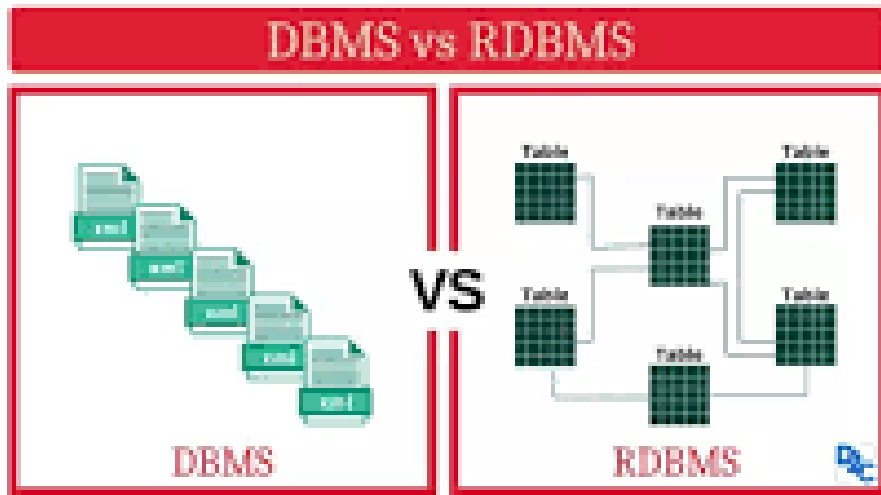




- La respuesta a esto es simple: un sistema de administración de bases de datos relacionales organiza los datos en tablas que se pueden vincular internamente según los datos compartidos.
- Esto permite que un usuario recupere fácilmente una o más tablas con una sola consulta.
- Por otro lado, el archivo plano almacena datos en una estructura de tabla única, que es menos eficiente y consume más espacio y memoria.

- Por lo tanto, necesitamos una base de datos relacional. Un ejemplo de un sistema de gestión de base de datos relacional podría ser un departamento de producción en una organización que aprovecha este modelo para procesar compras y realizar un seguimiento del inventario.
- El sistema de gestión de bases de datos o el sistema de gestión de bases de datos relacionales más disponible comercialmente y para toda la empresa que se utiliza en la actualidad es el lenguaje de consulta estructurado (base de datos SQL) para acceder a la base de datos.
- Otros ejemplos de sistemas de gestión de bases de datos relacionales ampliamente utilizados incluyen **Oracle Database**, **MySQL**, **PostgreSQL** (una base de datos relacional de código abierto) y Microsoft SQL Server. Las estructuras RDBMS se utilizan comúnmente para realizar cuatro operaciones básicas: **CRUD** (crear, leer, actualizar y eliminar), que son fundamentales para respaldar la gestión de datos consistente.

Diferencias entre RDBMS y DBMS



- Existen algunas diferencias contrastantes entre RDBMS y DBMS.
- Un RDBMS es una versión avanzada de un DBMS.
- A diferencia de un DBMS que administra bases de datos en una red informática y discos duros, una base de datos RDBMS ayuda a mantener las relaciones entre sus tablas.

•

Estas son algunas de las principales diferencias entre un RDBMS y un DBMS:

- **Número de operadores:** Un DBMS permite solo un solo operador simultáneamente, mientras que múltiples usuarios pueden operar un RDBMS simultáneamente. Un RDBMS utiliza algoritmos intrincados que permiten que varios usuarios accedan a la base de datos mientras preservan la integridad de los datos simultáneamente, lo que reduce significativamente el tiempo de respuesta.
- **Necesidad de hardware y software:** Un DBMS utiliza menos recursos de recuperación y almacenamiento de datos que un RDBMS. Este último es más complejo debido a su estructura de múltiples tablas y capacidad de referencia cruzada, lo que lo hace más costoso que un DBMS. Los RDBMS también se usan generalmente para aplicaciones de clase empresarial, mientras que los DBMS se usan más comúnmente para aplicaciones más pequeñas y específicas.

- **Modificación de datos:** La alteración de datos en un DBMS es bastante difícil, mientras que usted puede modificar fácilmente los datos en un RDBMS usando una consulta SQL. Por lo tanto, los programadores pueden cambiar / acceder a múltiples elementos de datos simultáneamente. Esta es una de las razones por las que un RDBMS es más eficiente que un DBMS.
- **Volumen de datos:** Un DBMS es más adecuado para manejar un volumen de datos bajo, mientras que un RDBMS puede manejar incluso grandes volúmenes de datos.
- **Claves e índices:** Un DBMS no involucra claves e índices, mientras que un RDBMS especifica una relación entre elementos de datos a través de claves e índices.

- **Consistencia de los datos:** Como un DBMS no sigue el modelo ACID (Atomicidad, Consistencia, Aislamiento y Durabilidad), los datos almacenados pueden tener inconsistencias. En contraste, un RDBMS sigue el modelo ACID, haciéndolo estructurado y consistente.
- **Estructura de la base de datos:** Un DBMS funciona almacenando datos en una estructura jerárquica, mientras que un RDBMS almacena datos en tablas.
- **Velocidad de obtención de datos:** En un DBMS, el proceso es relativamente lento, especialmente cuando los datos son complejos y extensos. Esto se debe a que cada uno de los elementos de datos debe obtenerse individualmente. En un RDBMS, los datos se obtienen más rápido debido al enfoque relacional. Además, SQL facilita una recuperación de datos más rápida en un RDBMS.

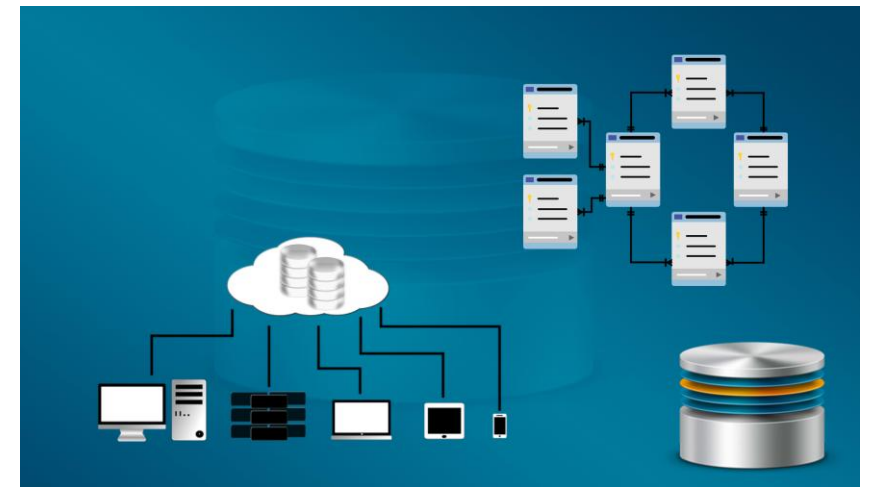
- **Bases de datos distribuidas:** Un DBMS no admite bases de datos distribuidas, mientras que un RDBMS ofrece soporte completo para bases de datos distribuidas.
- **Arquitectura cliente-servidor:** A diferencia de un DBMS, un RDBMS admite arquitectura Cliente-Servidor.



¿Cómo funciona un sistema de gestión de bases de datos relacionales?

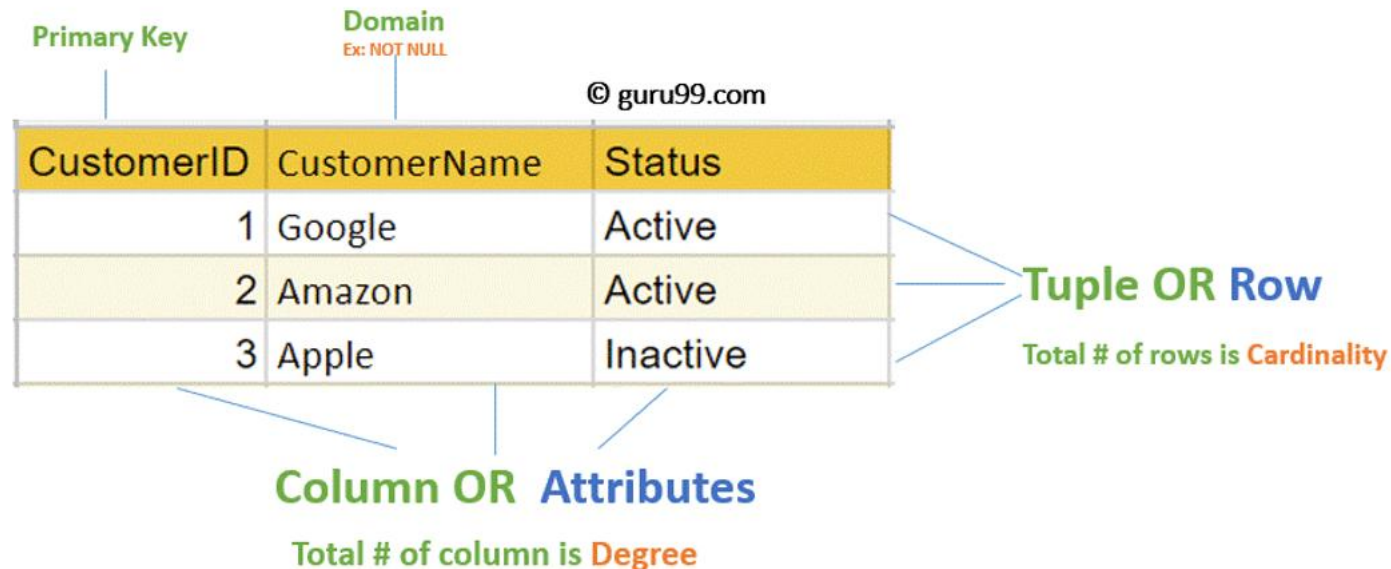
Los datos se almacenan en una base de datos relacional en forma de tablas múltiples. Aquí surge una pregunta clave, ¿cómo funciona una estructura de base de datos y cómo se implementa? Entendamos esto en detalle:

- Una estructura de base de datos funciona organizando cada tabla en **filas** (conocidas como registros/tuplas) y **columnas** (conocidas como campos/atributos).
- Las tablas, las columnas y las filas son los tres componentes principales de una base de datos relacional.



- Aquí hay un ejemplo de un tipo de base de datos comercial para procesar los pedidos de los clientes. En este tipo de base de datos, la primera columna contiene el ID del cliente, que es la clave principal. La clave principal se utiliza para identificar cada registro único en una tabla. Cada columna (o atributo) almacena fragmentos de información y la base de datos asigna una identificación de cliente única a cada fila.

Table also called Relation



Ejemplo de tabla de base de datos relacional.

La estructura de datos utilizada para implementar una base de datos relacional es la siguiente:

- Como se ve en el ejemplo anterior, la estructura de datos de una sola tabla. Aunque lo más común es que los datos sean más complejos y se almacenen en varias tablas. Luego, las tablas se conectan en función de las relaciones preexistentes.

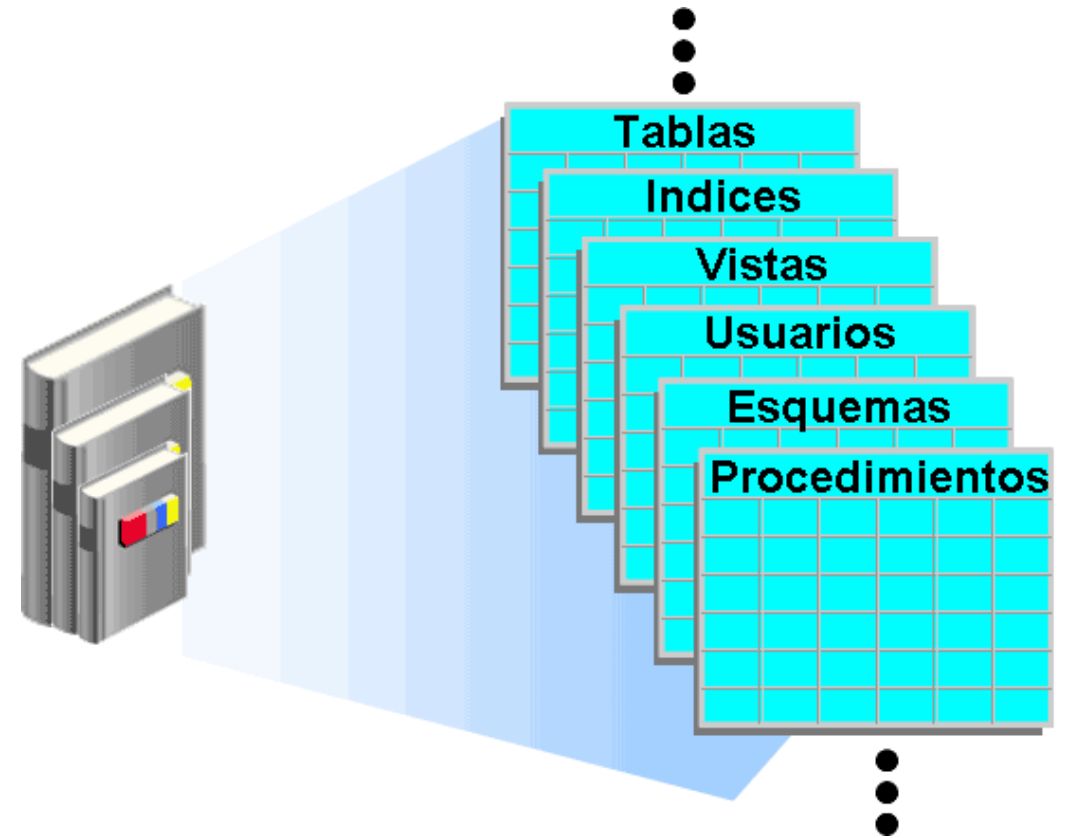
Las tablas se pueden conectar de diferentes formas, como por ejemplo:

- Un registro en una tabla podría estar relacionado con otro registro en una tabla diferente (relación 1: 1)
- Un registro en una tabla puede estar relacionado con varios registros en otras tablas (relación 1: M)
- Varios registros de tabla se pueden vincular a varios registros en una tabla diferente (relación M: N)

Sin embargo, antes de crear las tablas, un sistema de administración de bases de datos relacionales debe garantizar que:

- Cada tabla tiene una clave primaria única, que no tiene valores nulos.
- La clave externa, que se utiliza para relacionarse con 2 tablas, se conserva en una tabla y se refiere a la clave principal de otra tabla.
- Ninguna columna tiene un valor nulo (celda vacía).

- Un RDBMS generalmente viene con diccionarios de datos y colecciones de metadatos, que son beneficiosos para la gestión de datos.
- Un diccionario de datos define los objetos de datos de cada usuario dentro de la base de datos. Como resultado, ayuda a los usuarios a identificar todos los objetos que existen en la base de datos y quién puede acceder a ellos.





- Por ejemplo, es humanamente imposible recordar todas las tablas almacenadas en una enorme base de datos.
- Si desea buscar una tabla específica, como Productos, puede usar un diccionario de datos para buscar todas las tablas denominadas Productos. Luego, puede identificar el requerido de la lista de tablas.

Ventajas del sistema de gestión de bases de datos relacionales

Las ventajas de un sistema de gestión de bases de datos relacionales ofrecen una vista sistemática de los datos, lo que ayuda a las empresas a mejorar sus procesos de toma de decisiones al mejorar diferentes áreas.

Otras ventajas de un modelo de base de datos relacional:

- **Seguridad de datos mejorada:** Las funciones de autorización y control de acceso del software de bases de datos relacionales admiten el cifrado y descifrado avanzados, lo que permite a los administradores de la base de datos gestionar el acceso a los datos almacenados. Esto ofrece importantes beneficios en términos de seguridad. Además, los operadores pueden modificar el acceso a las tablas de la base de datos e incluso limitar los datos disponibles a otros. Esto convierte a los RDBMS en una solución de almacenamiento de datos ideal para empresas en las que la alta dirección necesita controlar el acceso a los datos para los trabajadores y los clientes.

- **Retener la consistencia de datos:** Es más fácil agregar nuevos datos o modificar tablas existentes en un RDBMS mientras se mantiene la coherencia de los datos con el formato existente. Esto se debe principalmente a que un RDBMS es compatible con ACID.
- **Mejor flexibilidad y escalabilidad:** Un RDBMS ofrece más flexibilidad a la hora de actualizar los datos, ya que las modificaciones solo deben realizarse una vez. Por ejemplo, actualizar los detalles en la tabla principal actualizará automáticamente los archivos relevantes y le evitará la molestia de cambiar varios archivos uno por uno. Además, cada mesa se puede modificar de forma independiente sin molestar a las demás. Esto hace que las bases de datos relacionales sean escalables para volúmenes de datos crecientes.

- **Fácil de mantener:** Las bases de datos relacionales se consideran de bajo mantenimiento porque los usuarios pueden probar, regular, corregir y hacer copias de seguridad de los datos rápidamente, ya que la herramienta de automatización en RDBMS ayuda a sistematizar estas tareas.
- **Riesgo reducido de errores:** En el software de bases de datos relacionales, puede verificar fácilmente si hay errores en los datos de diferentes registros. Además, como cada elemento de datos se almacena en una sola ubicación, no hay posibilidad de que versiones anteriores difuminen la imagen.

3.1.4 Instalación y configuración de un SMBD

Instalación de mariadb en Raspberry Pi 5 Server

- `sudo apt install mariadb-server mariadb-client -y`

Asegurar MariaDB

- `sudo mysql_secure_installation`

Recomendaciones:

- Cambiar contraseña root
- Eliminar usuarios anónimos
- Eliminar base de datos de prueba
- Deshabilitar acceso remoto root

Configurar base de datos y usuario

- `sudo mysql -u root -p`

Dentro de MariaDB:

- `CREATE DATABASE testdb;`
- `CREATE USER 'nodeuser'@'localhost' IDENTIFIED BY 'password_seguro';`
- `GRANT ALL PRIVILEGES ON testdb.* TO 'nodeuser'@'localhost';`
- `FLUSH PRIVILEGES;`
- `EXIT;`

Prueba conexión:

- `mysql -u nodeuser -p testdb`

Instalación de mariadb en Windows

- Entrar a: <https://mariadb.org/download>
- Seleccionar sistema operativo Windows , la arquitectura de la maquina donde se instalara y msi package -> Clic en Downndload



The screenshot shows the MariaDB download page with the following configuration:

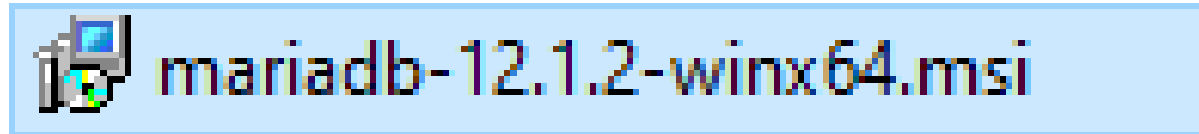
- MariaDB Server Version:** MariaDB Server 12.1.2 Rolling
- Display older releases:**
- Operating System:** Windows
- Architecture:** x86_64
- Package Type:** MSI Package
- Mirror:** Accretive Networks

Additional information at the bottom of the form:

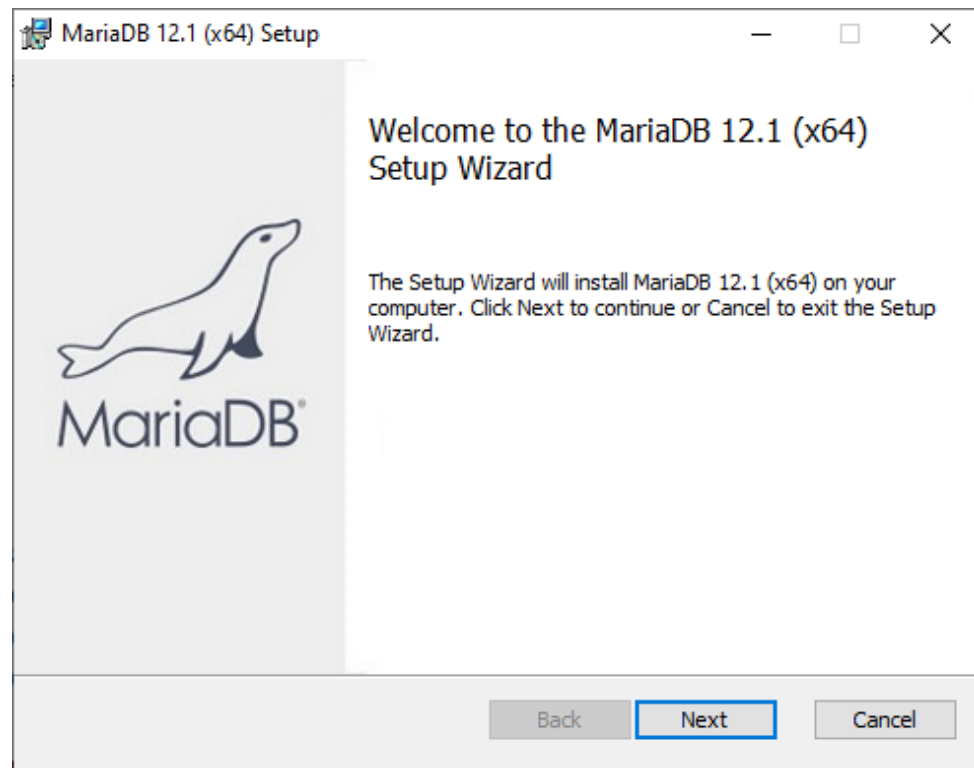
- Release date: 2025-11-19
- File name: mariadb-12.1.2-winx64.msi
- File size: 83.0 MB

The left sidebar contains navigation links: Download MariaDB Server, REST API, Reporting Bugs, MariaDB Server Statistics, Download MariaDB Server Documentation, View all releases for: MariaDB Server, Connector/C, Connector/J, Connector/ODBC, Connector/Python, Connector/Node.js, and Community Server Support.

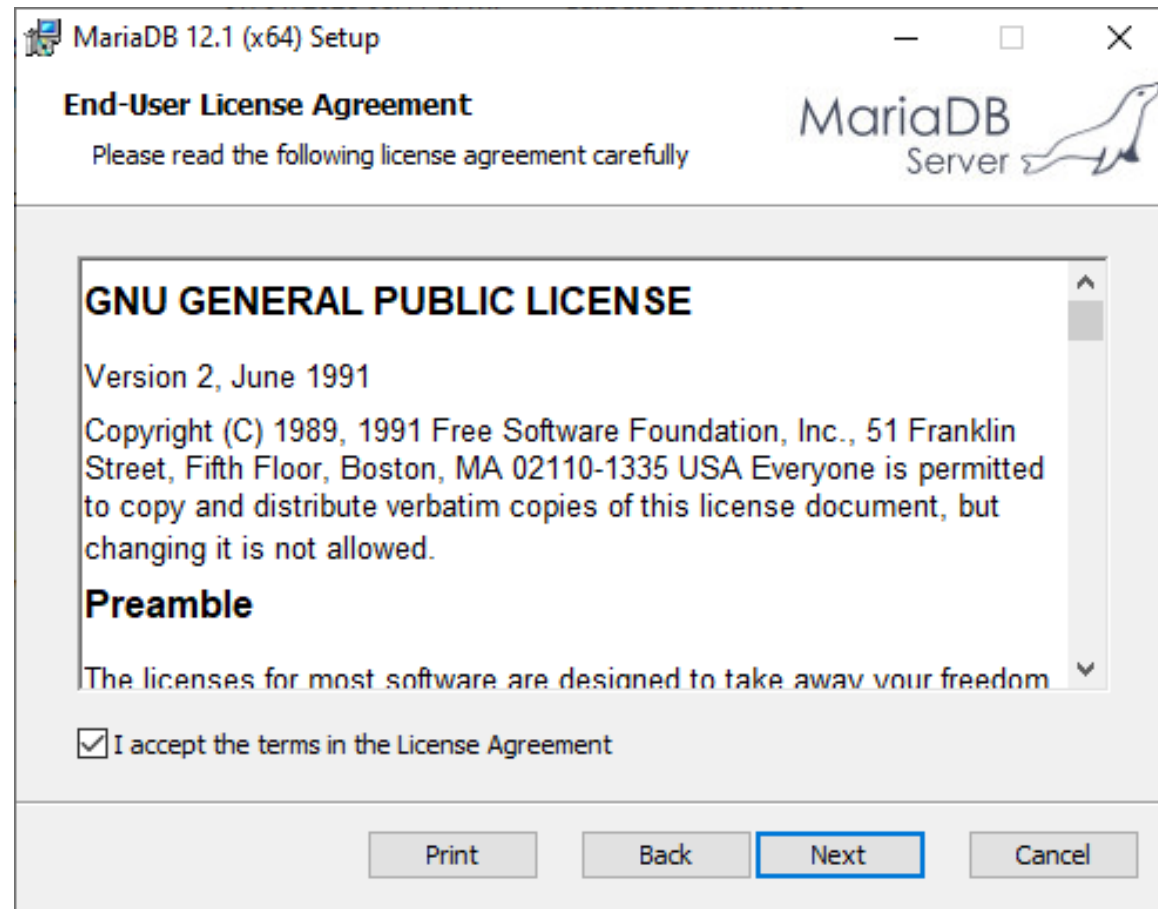
- Dar clic en el archivo .msi para la instalación del SMBD



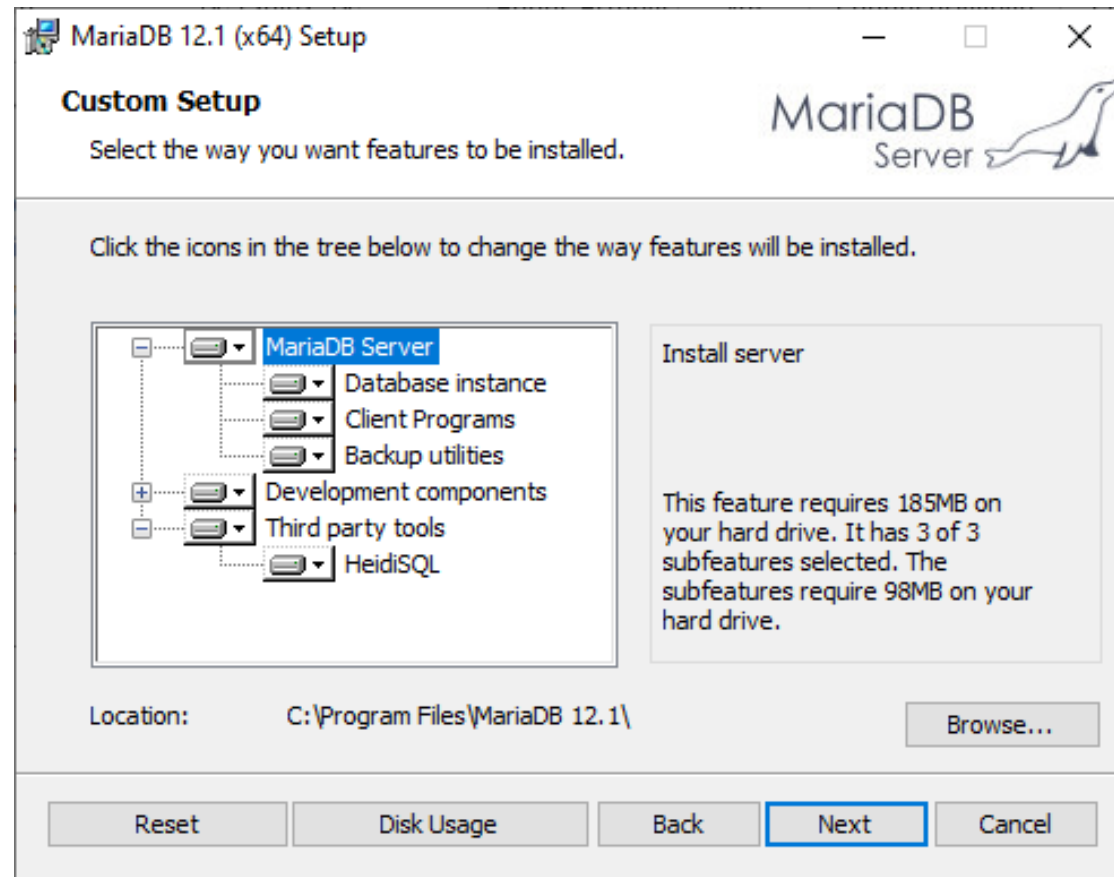
- Abrirá la siguiente ventana, dar clic en Next



- Aceptar términos de la licencia



- En la siguiente ventana dar clic en Next, esta ventana corresponde a las características que se van a instalar



- En la siguiente ventana ingresar un password

User settings

Default instance properties
MariaDB 12.1 (x64) database configuration

MariaDB Server

Modify password for database user 'root'

New root password: Enter new root password

Confirm: Retype the password

Enable access from remote machines for 'root' user


Enable the Feedback plugin and submit anonymous usage information
Collected statistics can be viewed at <https://mariadb.org/feedback-plugin>

Data directory

- En la siguiente imagen muestra el nombre del servicio y el puerto que utilizara , sin modificar nada dar clic en Next

Database settings

Default instance properties
MariaDB 12.1 (x64) database configuration

MariaDB Server 

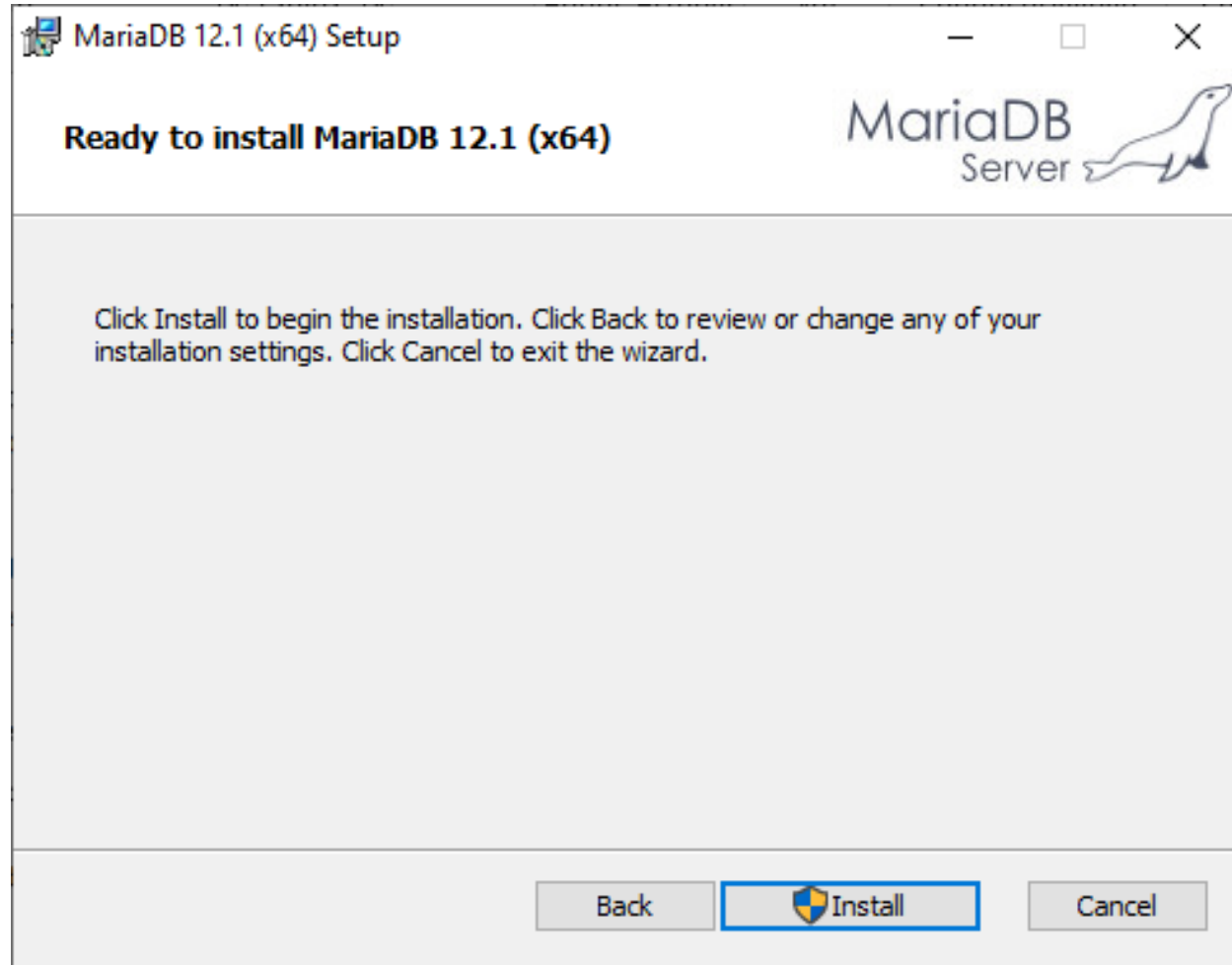
Install as service
Service Name:

Enable networking
TCP port:

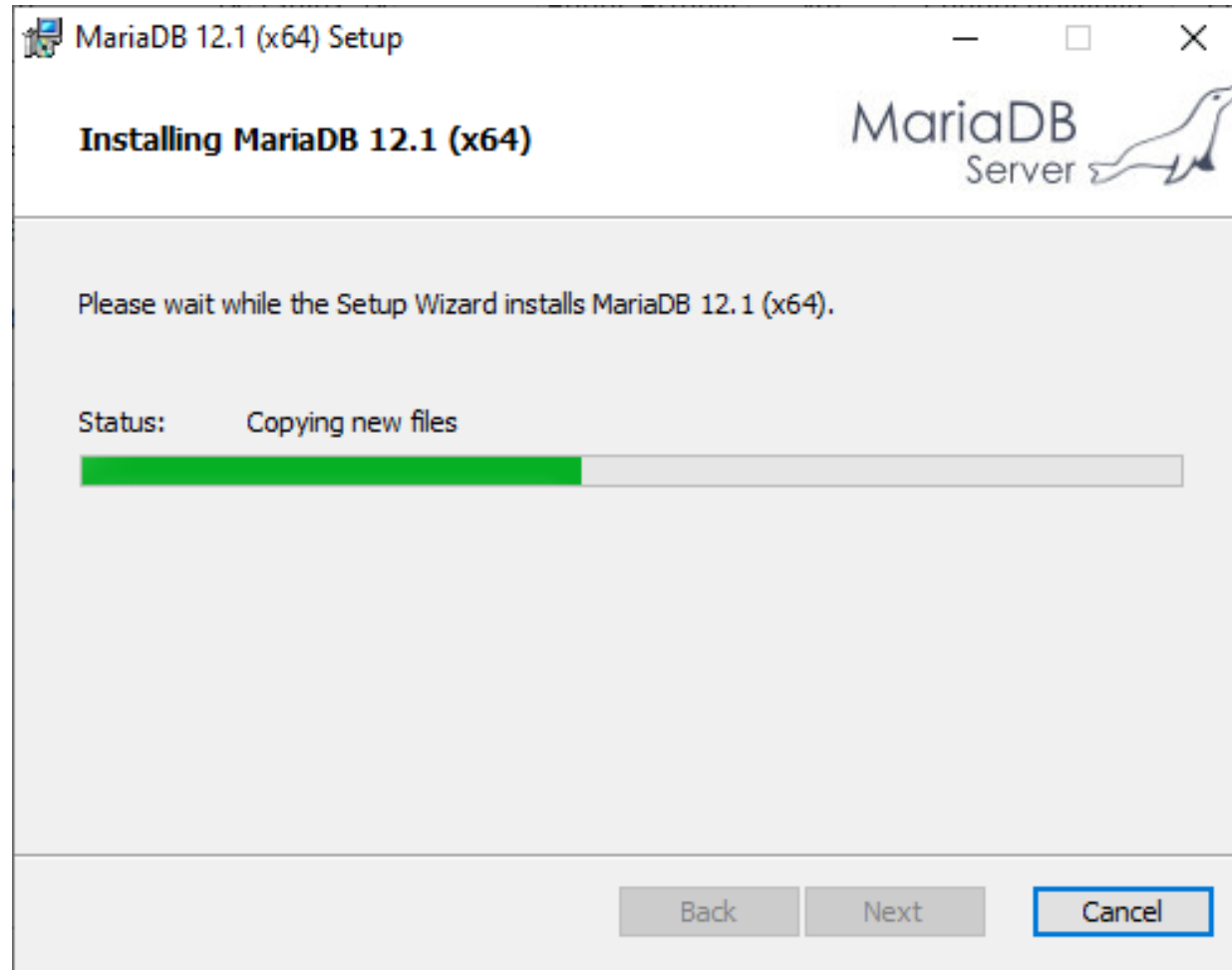
InnoDB engine settings
Buffer pool size: MB
Page size: KB

Back Next Cancel

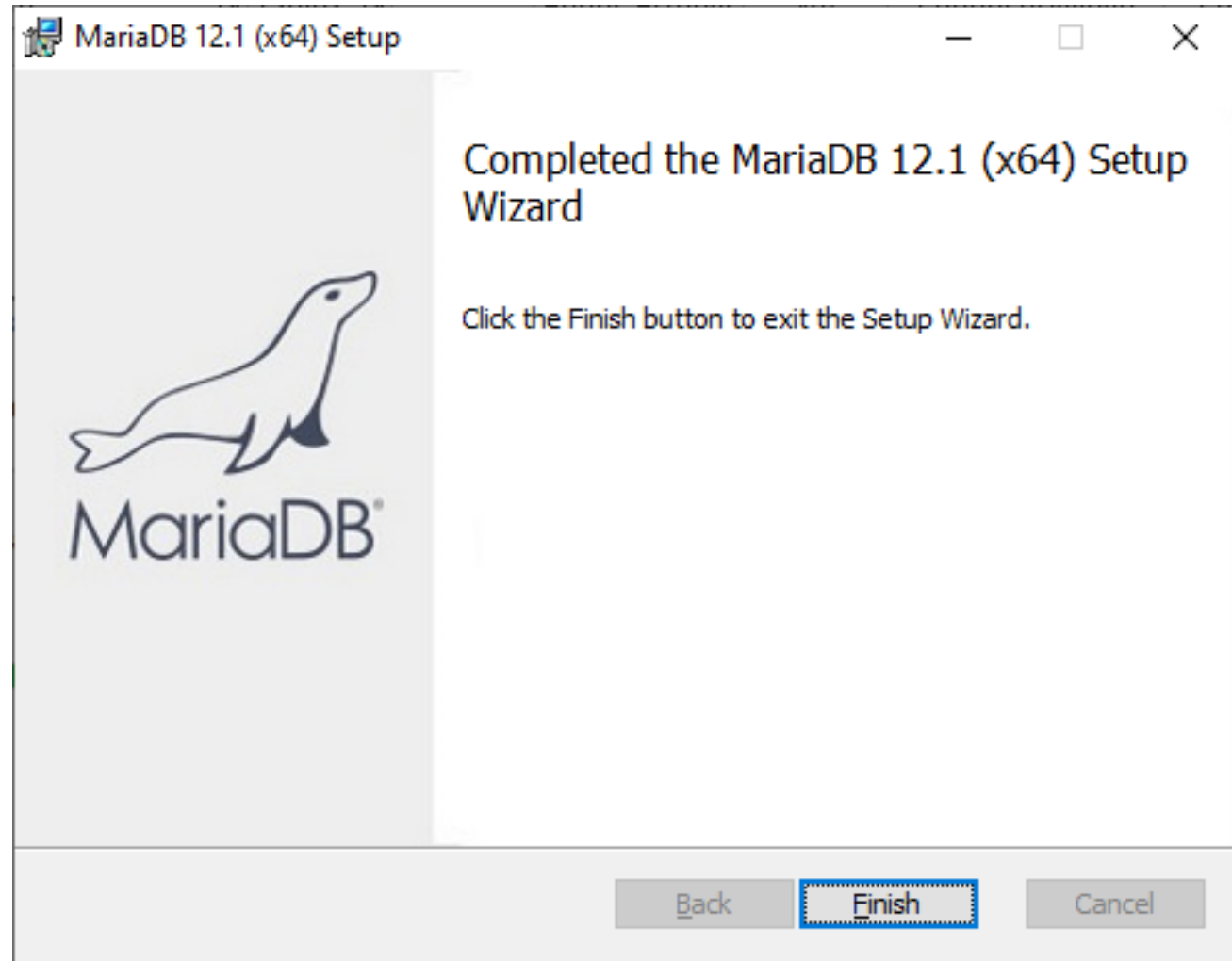
- En la siguiente ventana dar clic en Install



- A continuación la ventana muestra el proceso de instalación

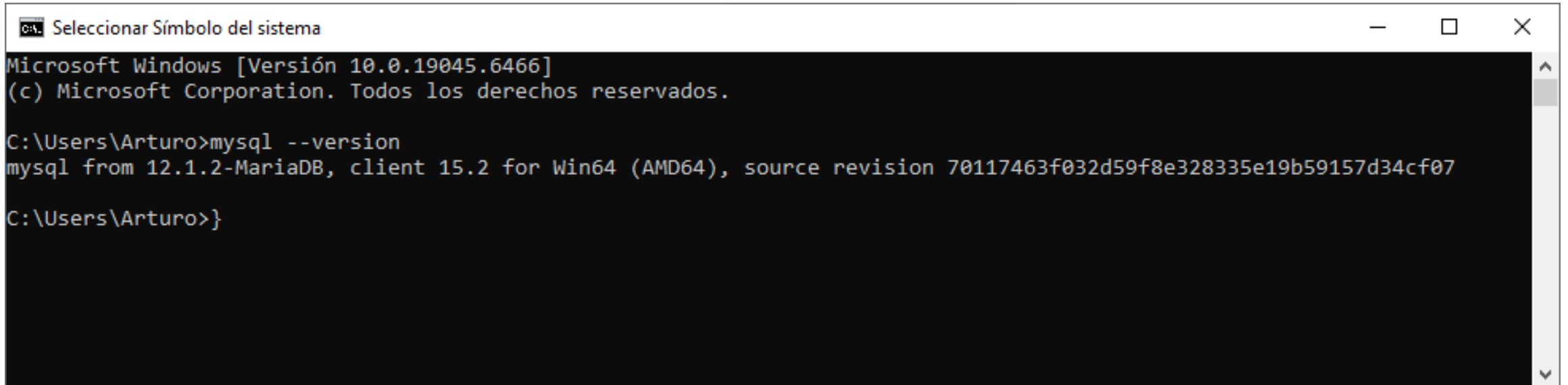


- Una vez Finalizado dar clic en Finish



Verificar que mariadb funcione:

- Desde CMD o Powershell
- `mysql --version`

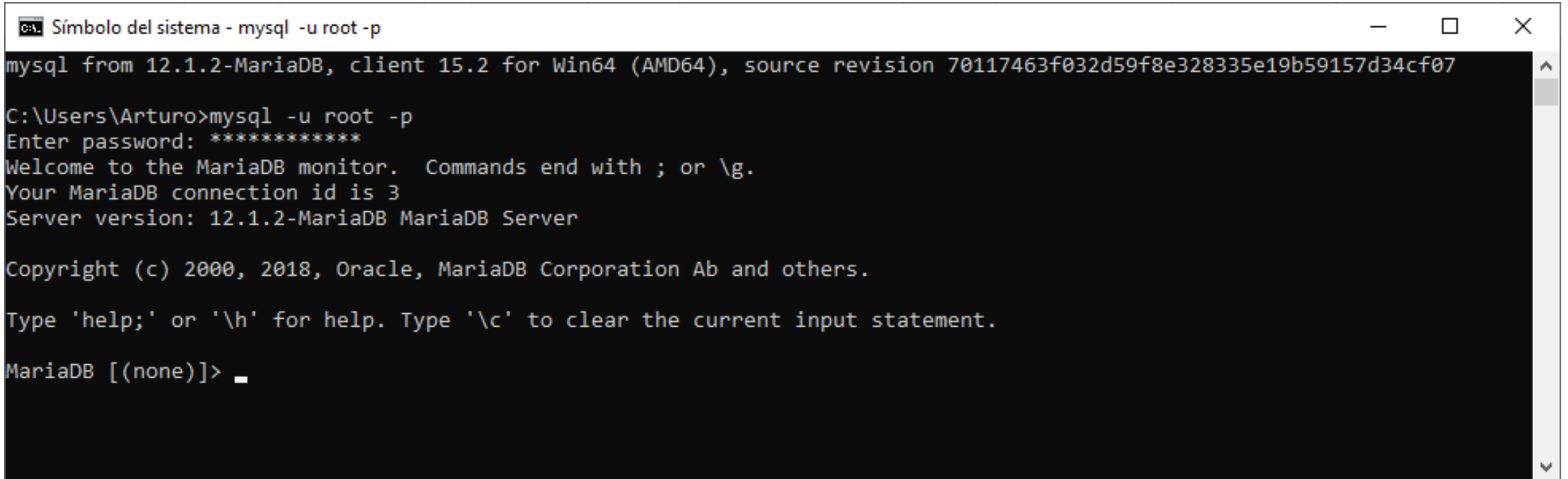


```
Seleccionar Símbolo del sistema
Microsoft Windows [Versión 10.0.19045.6466]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\Arturo>mysql --version
mysql from 12.1.2-MariaDB, client 15.2 for Win64 (AMD64), source revision 70117463f032d59f8e328335e19b59157d34cf07

C:\Users\Arturo>}
```

Ingresamos comando `mysql -u root -p` (Solicitará la contraseña que se asignó durante la instalación), si todo es correcto entraras al SMBD mariadb.



```
Símbolo del sistema - mysql -u root -p
mysql from 12.1.2-MariaDB, client 15.2 for Win64 (AMD64), source revision 70117463f032d59f8e328335e19b59157d34cf07
C:\Users\Arturo>mysql -u root -p
Enter password: *****
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 3
Server version: 12.1.2-MariaDB MariaDB Server

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> _
```

- Una vez dentro se pueden ejecutar los comandos del proceso realizado anteriormente en la **Raspberry Pi 5**.