



# COMUNICACIÓN MATLAB/ MYSQL; GUÍA PRÁCTICA CON EJEMPLOS.

## *MATLAB/ MYSQL COMMUNICATION; PRACTICAL GUIDE WITH EXAMPLES.*

<sup>1</sup> Universidad Estatal Península de Santa Elena. Facultad de Sistemas y Telecomunicaciones. Electrónica y Automatización. La Libertad-Santa Elena. Ecuador.

Autor: Arturo Santos Sánchez.

ORCID: <https://orcid.org/0009-0000-3107-4692>

Correo: [Arturo.santossanchez@upse.edu.ec](mailto:Arturo.santossanchez@upse.edu.ec)

<sup>2</sup> Universidad Estatal Península de Santa Elena. Facultad de Sistemas y Telecomunicaciones. Electrónica y Automatización. La Libertad-Santa Elena. Ecuador.

Autor: Marco Espinoza Arroyo

ORCID: <https://orcid.org/0000-0001-7122-994>

Correo: [marco.espinozaarroyo@upse.edu.ec](mailto:marco.espinozaarroyo@upse.edu.ec)

### Resumen

En el entorno de la programación y la gestión de bases de datos, la integración efectiva entre diferentes herramientas y sistemas es esencial para lograr un flujo de trabajo eficiente y una manipulación de datos sin problemas. Este conjunto de soluciones y ejemplos se centra en la conexión y comunicación entre MATLAB y “MySQL”, dos poderosas herramientas ampliamente utilizadas en la comunidad de desarrollo y análisis de datos. Este conjunto de ejemplos y aplicaciones ilustra cómo establecer conexiones entre MATLAB y “MySQL”, realizar operaciones de lectura y escritura en bases de datos, y aplicar estos conceptos en diversos contextos, desde el análisis de datos relacionados con “COVID-19” hasta la creación de calculadoras y graficadoras. Cada ejemplo demuestra cómo aprovechar la integración entre estas dos plataformas para mejorar la eficiencia en la manipulación de datos y desarrollar soluciones prácticas para diversas aplicaciones.

**Palabras clave:** programación; base de datos; conexión.

### Abstract

*In the environment of programming and database management, effective integration between different tools and systems is essential to achieve an efficient workflow and smooth data manipulation. This set of solutions and examples focuses on the connection and communication between MATLAB and MySQL, two powerful tools widely used in the data development and analysis community. This set of examples and applications illustrates how to establish connections between MATLAB and MySQL, perform read and write operations on databases, and apply these concepts in various contexts, from analyzing COVID-19-related data to creating calculators and graphers. Each example demonstrates how to leverage the integration between these two platforms to improve efficiency in data manipulation and develop practical solutions for various applications.*

**Keywords:** programming; database; connection.



## INTRODUCCIÓN

El programa MATLAB, un sistema de cómputo numérico y un lenguaje de programación, se destaca por su versatilidad en el manejo de matrices, la implementación de algoritmos y la creación de interfaces de usuario (The MathWorks, Inc., 2023). Por otro lado, “MySQL”, un reconocido sistema de gestión de bases de datos, ofrece una plataforma robusta para almacenar y recuperar datos de manera eficiente (Oracle, 2023). La combinación de estas dos herramientas permite una sinergia poderosa para manipular datos y facilitar tareas como análisis, visualización y almacenamiento (Gilat, 2006).

MySQL, en su esencia, es un sistema de gestión de bases de datos relacional (Suehring, 2001), lo que implica que organiza la información en tablas independientes en lugar de almacenar todos los datos en un único archivo (Casillas Santillán, Pérez Mora, & Gibert Ginestà, 2014). Esta arquitectura confiere notables ventajas en términos de velocidad y flexibilidad, permitiendo la rápida recuperación y manipulación de datos (Cambaudon, 2018). Además, estas tablas están intrínsecamente interconectadas, posibilitando la combinación de diversos datos en diferentes tablas y la consecuente creación de relaciones entre ellos.

Lo que hace a MySQL especialmente atractivo es su carácter de software de código abierto. Este enfoque otorga a cualquier individuo la capacidad de utilizar MySQL y personalizarlo según sus necesidades, incluso a través de la modificación del código fuente. Acompañando esta filosofía, se encuentra el GPL (General Public License), que establece los términos y condiciones para el uso y distribución del software (Arias, 2017). Aquellos que emplean MySQL deben acogerse a las directrices del GPL, lo que garantiza tanto la libertad de uso como la comunidad comprometida en torno a su desarrollo y mejora continua (Christudas, 2019).



## MATERIALES Y MÉTODOS

### **Materiales necesarios:**

- **MATLAB:** tener acceso a MATLAB instalado en el sistema del ordenador a trabajar.
- Base de datos MySQL: se debe tener una base de datos MySQL en funcionamiento.
- Controlador JDBC: se requiere un controlador JDBC (Java Database Connectivity) que permitirá conectar a MySQL desde MATLAB. Se puede descargar el controlador MySQL JDBC desde el sitio web de MySQL.

### **Métodos:**

#### **Guía para la Comunicación.**

1. Comunicación MATLAB/MySQL
  - Descargar el programa de la página oficial de MATLAB Y MySQL
2. Instalación de MySQL "Conector-ODBC".
  - Descargar el Programa MySQL "Conector-ODBC" (v. 5.3.14)
3. Creación de conexión MySQL.
4. Conexión entre MATLAB y MySQL.
5. Creación de base de datos MySQL.
6. Guardar las bases de datos en la conexión MySQL

#### **Tipos de ejemplos a mostrar:**

- Análisis de COVID-19 en MATLAB.
- Calculadora de derivadas en MATLAB.
- Calculadora operaciones básicas en MATLAB.
- Graficadora de ecuaciones.
- Facturación en MATLAB.
- Cálculos de Ingresos y Egresos en MATLAB.
- Calculadora de polinomio en MATLAB.



## RESULTADOS Y DISCUSIÓN

### **Ejemplos de comunicación MATLAB/MySQL:**

- Análisis de COVID-19 en MATLAB.

#### Paso 1: Creación de conexión MySQL

#### Paso 2: Crear la base de datos MySQL

#### Paso 3:

Se crean las tablas con el [Paso2](#) en MySQL con su tipo de dato y solo se ingresan los datos de las tablas llamadas (Datos\_1 y Datos\_2), porque en la tabla 3 (Registro) se registran datos para el programa que se elaborara en MATLAB.

#### Paso 4: Guardar la base de datos en la conexión MySQL

#### Paso 5: Establecemos la conexión con MATLAB y MySQL

#### Paso 6:

Una vez ya guardada la base de datos y establecido la conexión, nos dirigimos a abrir el programa ya elaborado en MATLAB. Que contiene ya el código listo para leer y escribir datos que se guardan en la base de datos.

Dentro del Código al principio se pone este código para que inicie la conexión entre MATLAB y MySQL.

```
clc,clear,close all;  
con=database('conexión','root','123456789')
```

El formato para otro tipo de máquina que desea utilizar el programa sería el siguiente:

```
conn=database ('Conexión','Usuario','Password')
```

Esto quiere decir lo siguiente:

- 'Conexión': Lo que se creó en [creación de conexión](#) MySQL el nombre puede cambiar, así mismo como cambiar tiene que ponerlo en el código de MATLAB.
- 'Usuario': Aquí va el usuario que se puso en la conexión de MySQL.
- 'Password': Así mismo como el usuario va la contraseña que se puso en la instalación de mysql.

Los códigos para Leer y Escribir En la Base de Datos.

Leer los datos de los pacientes infectados almacenados en SQL.

Va ubicado en el Case 1 para poder Graficar.



```
y1 = fetch(conn,'SELECT PortInfectados FROM programa_1.datos_2;');  
y1=table2cell(y1);  
ocup=cell2mat(y1(:,1));  
y2 = fetch(conn,'SELECT PortInfectados FROM programa_1.datos_1;');  
y2=table2cell(y2);  
ed=cell2mat(y2(:,1));
```

Guardar los datos de la probabilidad de contagio en MySQL

Va en el Case 2

```
call={'idRegistro','Edad','Ocupacion','ProbabilidadInfeccion'};  
insert(con,'programa_1.registro',call,{id,e,oc,pc});
```

Como último también va esta línea de código para cerrar la base de datos

```
dis(' ');  
close(con)
```

### Paso 7:

Se inicia el programa para verificar que la conexión se ha establecido correctamente, eso se puede observar en el “Command Window” y tranquilamente trabajaríamos con el programa.

```
Command Window  
  
conn =  
  
connection with properties:  
  
    DataSource: 'Conexion'  
    UserName: 'root'  
    Message: ''  
    Type: 'ODBC Connection Object'  
Database Properties:  
  
    AutoCommit: 'on'  
    ReadOnly: 'off'  
    LoginTimeout: 0  
    MaxDatabaseConnections: 0  
Catalog and Schema Information:  
  
    DefaultCatalog: 'programa_1'  
    Catalogs: {'information_schema', 'mydb', 'mysql' ... and 4 more}  
    Schemas: {}  
Database and Driver Information:  
  
    DatabaseProductName: 'MySQL'  
    DatabaseProductVersion: '8.0.21'  
    DriverName: 'myodbc6w.dll'  
    DriverVersion: '08.00.0021'  
  
CALCULO DE PROBABILIDAD DE CONTAGIO DE COVID-19  
  
Digite 1 para ver las graficas de contagio generales.  
Digite 2 para obtener su porcentaje de contagio.  
Digite 3 si desea salir.  
Escoja una opcion: |
```

Figura 1: Paso 7



## Paso 8: [Final](#)

Verificamos la lectura y escrituras de Datos

Leer los datos de los pacientes infectados almacenados en “SQL”.

GRAFICAS DE CONTAGIO GENERALES

Si desea continuar digite 1  
Que desea hacer:

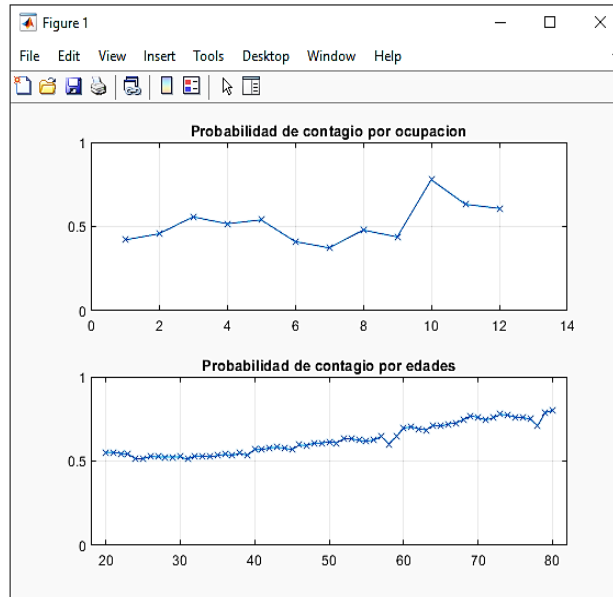


Figura2: Graficas de contagios generales

Guardar los datos de la probabilidad de contagio en MySQL

- Calculadora de derivadas en MATLAB.

Paso 1: [Creación de conexión MySQL.](#)

Paso 2: [Crear la base de datos MySQL.](#)

Paso 3:

Se crean las tablas con el [paso 2](#) en MySQL con su tipo de dato y solo se registran los datos en la tabla llamada (Datos\_1), para el programa que se elaborara en MATLAB.

Paso 4: [Guardar la base de datos en la conexión MySQL.](#)

Paso 5: [Establecemos la conexión con MATLAB y MySQL.](#)

Paso 6:

Una vez ya guardada la base de datos y establecido la conexión, nos dirigimos a abrir el programa ya elaborado en MATLAB. Que contiene ya el código listo para leer y escribir datos que se guardan en la base de datos.

Dentro del Código al principio se poner este código para que inicie la conexión entre MATLAB y MySQL.



```
clc,clear,close all;  
con=database('conexión','root','123456789')
```

El formato para otro tipo de máquina que desea utilizar el programa sería el siguiente:

```
conn=database ('Conexión','Usuario','Password')
```

Esto quiere decir lo siguiente:

- 'Conexión': Lo que se creó en [creación de conexión](#) MySQL el nombre puede cambiar, así mismo como cambiar tiene que ponerlo en el código de MATLAB.
- 'Usuario': Aquí va el usuario que se puso en la conexión de MySQL.
- 'Password': Así mismo como el usuario va la contraseña que se puso en la instalación de mysql.

Los códigos para Leer y Escribir En la Base de Datos.

Leer los datos guardados en MATLAB.

```
Case 2  
y1 = fetch(con,'SELECT * FROM prog_derivada.dato_1;');  
disp(y1)
```

Guardar los datos en la tabla creada en MySQL.

```
insert(con,'prog_derivadas.datos_1',col,val);
```

Como último también va esta línea de código para cerrar la base de datos

```
dis(' ');  
close(con)
```

Paso 7:

Se inicia el programa para verificar que la conexión se ha establecido correctamente, eso se puede observar en el "Command Window".



```
Command Window

conn =

  connection with properties:

      DataSource: 'Conexion'
      UserName: 'root'
      Message: ''
      Type: 'ODBC Connection Object'
Database Properties:

      AutoCommit: 'on'
      ReadOnly: 'off'
      LoginTimeout: 0
      MaxDatabaseConnections: 0

Catalog and Schema Information:

      DefaultCatalog: 'programa_1'
      Catalogs: {'information_schema', 'mydb', 'mysql' ... and 4 more}
      Schemas: {}

Database and Driver Information:

      DatabaseProductName: 'MySQL'
      DatabaseProductVersion: '8.0.21'
      DriverName: 'myodbc8w.dll'
      DriverVersion: '08.00.0021'

CALCULADORA DE DIFERENCIALES

Digite 1 PARA RESOLVER LA ECUACION DIFERENCIAL.
Digite 2 PARA VISUALIZAR LAS ECUACIONES REALIZADAS.
Digite 3 PARA SALIR.
fx Escoja una opcion: |
```

Figura3: Paso 7

### Paso 8: [Final](#)

Verificamos la lectura y Escrituras de Datos

La ecuación que Insertamos se guardó en la tabla y podemos visualizar en el mismo MATLAB.

- Calculadora operaciones básicas en MATLAB.

Paso 1: [Creación de conexión MySQL.](#)

Paso 2: [Crear la base de datos MySQL.](#)

Paso 3:

Se crean las tablas con el "Paso 2" en MySQL con su tipo de dato y solo se registran los datos en la tabla llamada (DT\_Operaciones), para el programa que se elaborara en MATLAB.

Paso 4: [Guardar la base de datos en la conexión MySQL.](#)





### Paso 5: [Establecemos la conexión con MATLAB y MySQL.](#)

#### Paso 6:

Una vez ya guardada la base de datos y establecido la conexión, nos dirigimos a abrir el programa ya elaborado en MATLAB. Que contiene ya el código listo para leer y escribir datos que se guardan en la base de datos.

Dentro del Código al principio se poner este código para que inicie la conexión entre MATLAB y MySQL.

```
clc,clear,close all;  
con=database('conexión','root','123456789')
```

El formato para otro tipo de máquina que desea utilizar el programa sería el siguiente:

```
conn=database ('Conexión','Usuario','Password')
```

Esto quiere decir lo siguiente:

- 'Conexión': Lo que se creó en [creación de conexión](#) MySQL el nombre puede cambiar, así mismo como cambiar tiene que ponerlo en el código de MATLAB.
- 'Usuario': Aquí va el usuario que se puso en la conexión de MySQL.
- 'Password': Así mismo como el usuario va la contraseña que se puso en la instalación de mysql.

En este Programa Utilizamos otro Código para Iniciar la conexión en el MATLAB, (Pero también se puede utilizar el código de arriba también) cabe recalcar que siempre al inicio del código.

```
clc,clear;  
bd = Conexión"; user = "root"; pass = "1213456789";%se establece los parámetros  
conn = database(bd,user,pass);% se establece la conexión con la base de datos
```

Los códigos para Escribir y Leer En la Base de Datos  
Guardar los datos en la tabla creada en MySQL

```
y1 = fetch (con,"SELECT ID_Operacion FROM bd_calculadora.dt_operaciones");  
id = (y1)+1;  
val = {id, "SUMA",A,B,Suma};%establece los valores de la columna  
insert(conn,'bd_calculadora.dt_operaciones', col, val);%ingresa datos en la datae
```

La misma, pero se cambia el nombre de suma por las otras operaciones  
Leer los datos guardados en MATLAB

```
y1 = select(conn,"SELECT * FROM bd_calculadora.dt_operaciones");
```

Como último también va esta línea de código para cerrar la base de datos

```
dis(' ');
```



close(con)

### Paso 7: Final

Verificamos la lectura y Escrituras de Datos.

Guardar los datos.

```
PROGRAMA QUE REALIZA LAS OPERACIONES BÁSICAS CON DOS NUMEROS
OPERACIONES

-> 1. SUMA
-> 2. RESTA
-> 3. MULTIPLICACION
-> 4. DIVISION
-> 5. VER OPERACIONES REALIZADAS
-> 6. SALIR

Escoja la operacion: 1

*** SUMA ***

Ingrese el primer valor: 2
Ingrese el segundo valor: 2

valores
2+2

Resultado =
4
```

Figura 4: Paso 7

Así con todas Las Operaciones para Guardarlas en la Tabla.  
Leer los datos

Como podemos observar las operaciones que insertamos se guardaron en la tabla y podemos visualizar en el mismo MATLAB y el MySQL.

- Graficadora de ecuaciones

Paso 1: Creación de conexión MySQL.

Paso 2: Crear la base de datos MySQL.

Paso 3:

Se crean las tablas con el "Paso 2" en MySQL con su tipo de dato y solo se registran los datos en la tabla llamada (DT\_ECUACIONES), para el programa que se elaborara en MATLAB.

Paso 4: Guardar la base de datos en la conexión MySQL.

Paso 5: Establecemos la conexión con MATLAB y MySQL.

Paso 6:



Una vez ya guardada la base de datos y establecido la conexión, nos dirigimos a abrir el programa ya elaborado en MATLAB. Que contiene ya el código listo para leer y escribir datos que se guardan en la base de datos.

Dentro del Código al principio se poner este código para que inicie la conexión entre MATLAB y MySQL.

```
clc,clear,close all;  
con=database('conexión','root','123456789')
```

El formato para otro tipo de máquina que desea utilizar el programa sería el siguiente:

```
conn=database ('Conexión','Usuario','Password')
```

Esto quiere decir lo siguiente:

- 'Conexión': Lo que se creó en [creación de conexión](#) MySQL el nombre puede cambiar, así mismo como cambiar tiene que ponerlo en el código de MATLAB.
- 'Usuario': Aquí va el usuario que se puso en la conexión de MySQL.
- 'Password': Así mismo como el usuario va la contraseña que se puso en la instalación de mysql.

En este Programa Utilizamos otro Código para Iniciar la conexión en el MATLAB, (Pero también se puede utilizar el código de arriba también) cabe recalcar que siempre al inicio del código

```
clc,clear;  
bd = Conexión"; user = "root"; pass = "1213456789";%se establece los parámetros  
conn = database(bd,user,pass);% se establece la conexión con la base de datos
```

Los códigos para Escribir y leer En la Base de Datos

Guardar los datos en la tabla creada en MySQL:

```
insert(conn,'bd_graficacion.dt_ecuaciones', col, val);%ingresa datos en la datae
```

Leer los datos guardados en MATLAB:

```
y1 = select(conn,"SELECT * FROM bd_graficacion.dt_ecuaciones");
```

Como último también va esta línea de código para cerrar la base de datos:

```
dis(' ');  
close(con)
```

Paso 7: Final.

Verificamos la lectura y Escrituras de Datos

Guardar los datos de una ecuación con una variable.

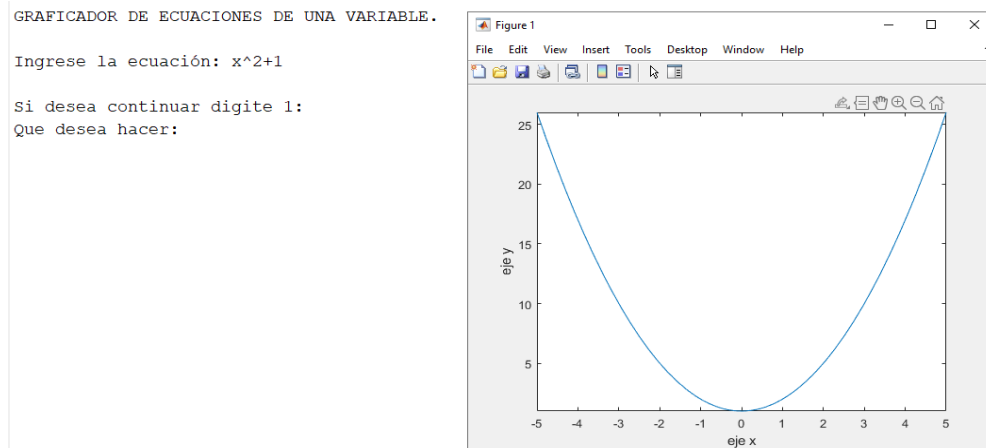


Figura 5: Gráfica de una variable

### Guardar los datos de una ecuación con dos variables

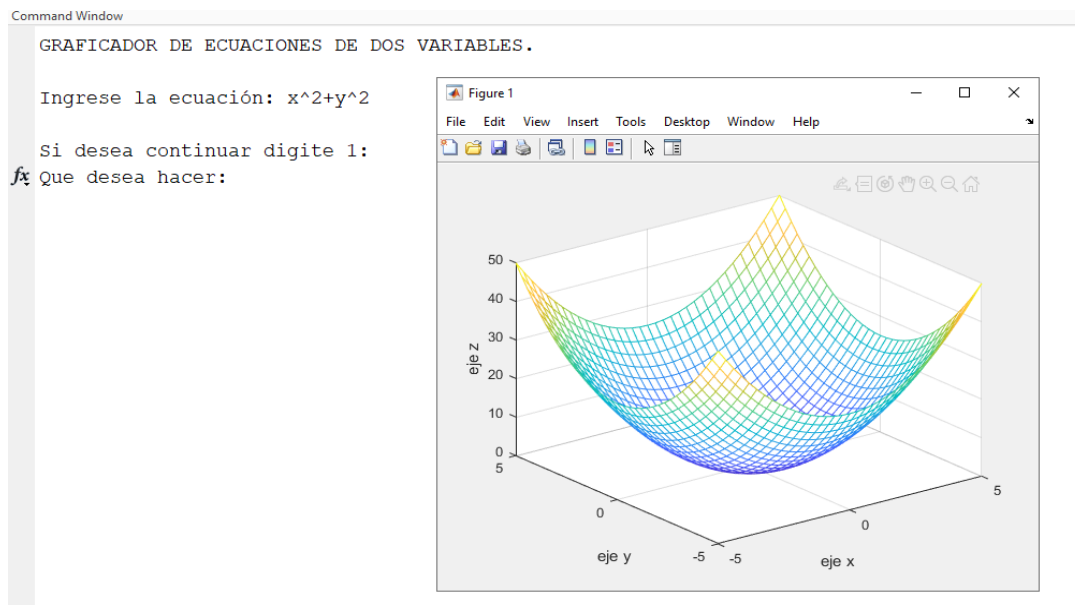


Figura 6: Gráfica de dos variables

Leer datos de la tabla, como se puede observar, muestra la Ecuaciones que se insertaron, se guardaron en la tabla y podemos visualizar en el mismo MATLAB y el MySQL.

- Facturación en MATLAB.

Paso 1: [Creación de conexión MySQL](#)

Paso 2: [Crear la base de datos MySQL](#)

Paso 3:



Se crean las tablas con el "Paso 2" en MySQL con su tipo de dato y solo se registran los datos en la tabla llamada (DT\_Factura), para el programa que se elaborara en MATLAB.

Paso 4: [Guardar la base de datos en la conexión MySQL.](#)

Paso 5: [Establecemos la conexión con MATLAB y MySQL.](#)

Paso 6:

Una vez ya guardada la base de datos y establecido la conexión, nos dirigimos a abrir el programa ya elaborado en MATLAB. Que contiene ya el código listo para leer y escribir datos que se guardan en la base de datos.

Dentro del código al principio se coloca lo siguiente para que inicie la conexión entre MATLAB y MySQL.

```
clc,clear,close all;  
con=database('conexión','root','123456789')
```

El formato para otro tipo de máquina que desea utilizar el programa sería el siguiente:

```
conn=database ('Conexión','Usuario','Password')
```

Esto quiere decir lo siguiente:

- 'Conexión': Lo que se creó en [creación de conexión](#) MySQL el nombre puede cambiar, así mismo como cambiar tiene que ponerlo en el código de MATLAB.
- 'Usuario': Aquí va el usuario que se puso en la conexión de MySQL.
- 'Password': Así mismo como el usuario va la contraseña que se puso en la instalación de MySQL.

En este programa utilizamos otro código para iniciar la conexión en el MATLAB (pero también se puede utilizar el código de arriba también) cabe recalcar que siempre al inicio del código.

```
clc,clear;  
bd = Conexión"; user = "root"; pass = "1213456789";%se establece los parámetros  
conn = database(bd,user,pass);% se establece la conexión con la base de datos
```

Los códigos para escribir y leer En la Base de Datos.

Guardar los datos en la tabla creada en MySQL:

```
val={id,prod,dcrp,prec,desc,total};%establece los valores de las columnas  
insert(con,'db_facturero.dt_factura',col, val);
```

Leer los datos guardados en MATLAB:

```
y1 = select(conn,"SELECT * FROM db_facturero.dt_factura");%lee los datos en la database
```



disp(y1)

Como último también va esta línea de código para cerrar la base de datos:

```
dis(' ');  
close(con)
```

### Paso 7: Final

Verificamos la lectura y escrituras de Datos

Command Window

CONSULTA DE VENTAS.

| id_Factura | Nombre_prod | Descripcion_prod | Precio_prod | Descuento_prod | Total_prod |
|------------|-------------|------------------|-------------|----------------|------------|
| 1          | {'Arroz'}   | {'Grano' }       | 2           | 10             | 1.8        |
| 2          | {'leche'}   | {'liquido'}      | 3           | 20             | 2.4        |

Si desea continuar digite 1:  
fx Que desea hacer: |

Figura 7: Lectura de datos

Como podemos observar las ventas que insertamos, se guardaron en la tabla y podemos visualizar en el mismo MATLAB y el MySQL.

- Cálculos de Ingresos y Egresos en MATLAB.

### Paso 1: Creación de conexión MySQL.

### Paso 2: Crear la base de datos MySQL.

### Paso 3:

Se crean las tablas con el "Paso 2" en MySQL con su tipo de dato y solo se registran los datos en la tabla llamada (DT\_), para el programa que se elaborara en MATLAB.

### Paso 4: Guardar la base de datos en la conexión MySQL.

### Paso 5: Establecemos la conexión con MATLAB y MySQL.

### Paso 6:

Una vez ya guardada la base de datos y establecido la conexión, nos dirigimos a abrir el programa ya elaborado en MATLAB. Que contiene ya el código listo para leer y escribir datos que se guardan en la base de datos.

Dentro del código al principio se poner lo siguiente para que inicie la conexión entre MATLAB y MySQL:



```
clc,clear,close all;  
con=database('conexión','root','123456789')
```

El formato para otro tipo de máquina que desea utilizar el programa sería el siguiente:

```
conn=database('Conexión','Usuario','Password')
```

Esto quiere decir lo siguiente:

- 'Conexión': Lo que se creó en [Creación de conexión](#) MySQL el nombre puede cambiar, así mismo como cambiar tiene que ponerlo en el código de MATLAB.
- 'Usuario': Aquí va el usuario que se puso en la conexión de MySQL.
- 'Password': Así mismo como el usuario va la contraseña que se puso en la instalación de mysql.

En este Programa Utilizamos otro Código para Iniciar la conexión en el MATLAB (pero también se puede utilizar el código de arriba) cabe recalcar que siempre al inicio del código:

```
clc,clear;  
bd = Conexión; user = "root"; pass = "1213456789";%se establece los parámetros  
conn = database.bd,user,pass);% se establece la conexión con la base de datos
```

Los códigos para Escribir y leer En la Base de Datos  
Guardar los datos en la tabla creada en MySQL:

```
col={'id_Ingresos','Ingresos'};%establece las columnas de las tablas  
val={id,ing};  
insert(conn,'db_ingresos_egresos.dt_ingreso' col, val);  
col={'id_Ingresos','Egresos'};%establece las columnas de las tablas  
val={id,ing};  
insert(conn,'db_ingresos_egresos.dt_Egreso' col, val);
```

Leer los datos guardados en MATLAB

```
y1 = fetch(con,'SELECT Ingresos FROM db_ingresos_egresos.dt_ingreso');  
y1 = table2cell (y1);  
y1 = cell2mat (y1);  
y1 = fetch(con,'SELECT Ingresos FROM db_ingresos_egresos.dt_ingreso');  
tam1 = size(y1);  
y2= fetch(con,'SELECT Ingresos FROM db_ingresos_egresos.dt_ingreso');  
y2 = table2cell (y2);  
y2 = cell2mat (y2);  
yy2 = fetch(con,'SELECT Ingresos FROM db_ingresos_egresos.dt_ingreso');  
tam2 = size(yy2);
```



Como último también va esta línea de código para cerrar la base de datos

```
dis(' ');  
close(con)
```

Paso 7: Final.

Verificamos la lectura y Escrituras de Datos  
Guardar los datos

```
Command Window  
CALCULO de Ingresos y Egresos  
  
Digite 1 Visualizar Ingresos y Egresos.  
Digite 2 Digitar Ingresos.  
Digite 3 Digitar Egresos.  
fx Escoja una opcion: |
```

Figura 8: Cálculo de ingresos y egresos

```
Command Window  
VENTANA DE INGRESOS  
  
Digite sus ingresos: 200  
  
Si desea continuar digite 1  
fx Que desea hacer: 1|
```

Figura 1: Ventana de ingresos

```
Command Window  
VENTANA DE EGRESOS  
  
Digite sus egresos: 300  
  
Si desea continuar digite 1  
fx Que desea hacer: 1|
```

Figura 10: Ventana de Egresos





Leer datos de la tabla, como podemos observar las gráficas de los valores que insertamos, se guardaron en la tabla y podemos visualizar en el mismo MATLAB y el MySQL.

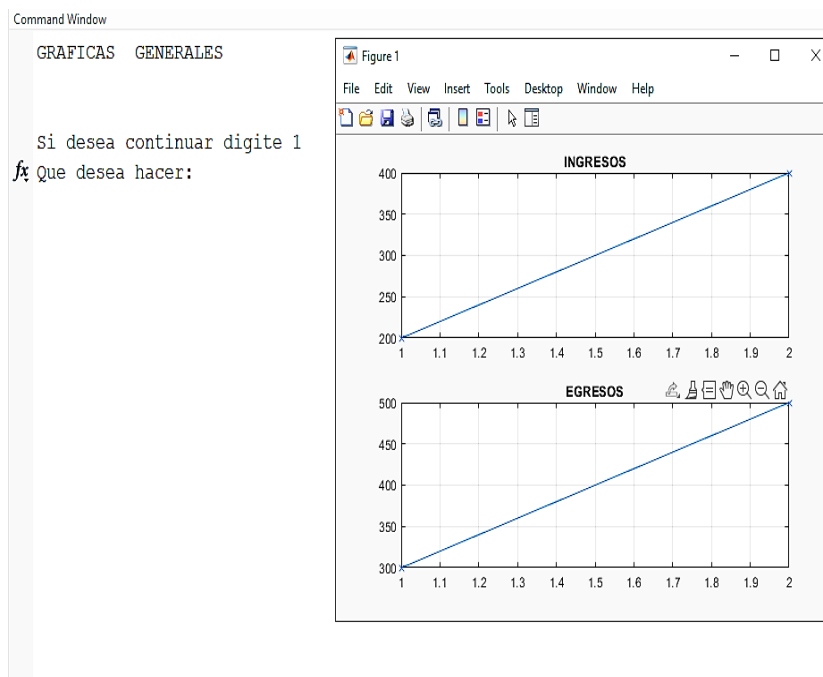


Figura 11: Gráfico de ingresos y egresos

- Calculadora de polinomio en MATLAB.

Paso 1: [Creación de conexión MySQL.](#)

Paso 2: [Crear la base de datos MySQL.](#)

Paso 3:

Se crean las tablas con el “Paso 2” en MySQL con su tipo de dato y solo se registran los datos en la tabla llamada (DT\_), para el programa que se elaborara en MATLAB.

Paso 4: [Guardar la base de datos en la conexión MySQL.](#)

Paso 5: [Establecemos la conexión con MATLAB y MySQL.](#)

Paso 6:

Una vez ya guardada la base de datos y establecido la conexión, nos dirigimos a abrir el programa ya elaborado en MATLAB. Que contiene ya el código listo para leer y escribir datos que se guardan en la base de datos

Dentro del Código al principio se poner este código para que inicie la conexión entre MATLAB y MySQL.

```
clc,clear,close all;
```



```
con=database('conexión','root','123456789')
```

El formato para otro tipo de máquina que desea utilizar el programa sería el siguiente:

```
conn=database ('Conexión','Usuario','Password')
```

Esto quiere decir lo siguiente:

- 'Conexión': Lo que se creó en [creación de conexión](#) MySQL el nombre puede cambiar, así mismo como cambiar tiene que ponerlo en el código de MATLAB.
- 'Usuario': Aquí va el usuario que se puso en la conexión de MySQL.
- 'Password': Así mismo como el usuario va la contraseña que se puso en la instalación de mysql.

En este Programa Utilizamos otro Código para Iniciar la conexión en el MATLAB, (Pero también se puede utilizar el código de arriba también) cabe recalcar que siempre al inicio del código

```
clc,clear;  
bd = Conexión"; user = "root"; pass = "1213456789";%se establece los parámetros  
conn = database(bd,user,pass);% se establece la conexión con la base de datos
```

Los códigos para Escribir y leer En la Base de Datos  
Guardar los datos en la tabla creada en MySQL

```
val = {id,a};  
insert(conn, 'db_polinomio.dt_polinomios',col,val);
```

Leer los datos guardados en MATLAB

```
y1 = fetch(con,"SELECT id_Polinomio FROM db_polinomio.dt_polinomio");  
id = size (y1) + 1;
```

Como último también va esta línea de código para cerrar la base de datos

```
dis(' ');  
close(con)
```

Paso 7: Final.

Verificamos la lectura y Escrituras de Datos.  
Guardar los datos.



```
Command Window
*** MENU ***

Digite 1 PARA RESOLVER RAICES DE UN POLINOMIO.
Digite 2 PARA VISUALIZAR LAS ECUACIONES REALIZADAS.
Digite 3 PARA SALIR.
Escoja una opcion: 1

*** RESOLVER RAICES DE UN POLINOMIO. ***

Ingrese la Ecuacion: x^4+3

Sus raices son:
2^(1/2)*3^(1/4)*(- 1/2 - 1i/2)
2^(1/2)*3^(1/4)*(- 1/2 + 1i/2)
2^(1/2)*3^(1/4)*(1/2 - 1i/2)
2^(1/2)*3^(1/4)*(1/2 + 1i/2)

Si desea continuar digite 1
fx Que desea hacer: |
```

Figura12: Paso 7

### Leer datos de la tabla

```
Command Window
*** MENU ***

Digite 1 PARA RESOLVER RAICES DE UN POLINOMIO.
Digite 2 PARA VISUALIZAR LAS ECUACIONES REALIZADAS.
Digite 3 PARA SALIR.
Escoja una opcion: 2

*** TABLA DE ECUACIONES REALIZADAS. ***

   id_Polinomios   polinomios
   -----
           1      {'x^2 + 4'}
```

Figura 13: Lectura de datos

Como podemos observar los datos que Insertamos, se guardaron en la tabla y podemos visualizar en el mismo MATLAB y el MySQL.



## Conclusiones

Una conclusión importante sobre la comunicación entre MATLAB y MySQL es que esta integración permite a los usuarios aprovechar las capacidades de procesamiento y análisis avanzados de MATLAB en conjunción con la potencia de almacenamiento y gestión de datos de MySQL. Esto es especialmente valioso en aplicaciones científicas y de ingeniería, donde se requiere el análisis de datos complejos y la generación de informes basados en bases de datos MySQL.

Al conectar MATLAB y MySQL de manera efectiva, los usuarios pueden:

- Importar datos desde bases de datos MySQL a MATLAB para análisis y visualización.
- Realizar operaciones de procesamiento y modelado de datos en MATLAB y almacenar los resultados nuevamente en MySQL.
- Automatizar tareas de recopilación, análisis y generación de informes de datos en tiempo real desde MySQL.
- Mantener una conexión bidireccional para actualizar y sincronizar datos entre MATLAB y MySQL de manera eficiente.

La comunicación entre MATLAB y MySQL es esencial para optimizar el flujo de datos entre estas dos herramientas, lo que permite a los usuarios aprovechar al máximo las ventajas de ambas plataformas en sus proyectos de análisis de datos y modelado.



## REFERENCIAS

Arias, M. Á. (2017). *Aprende programación web con php y mysql: 2ª edición*. IT campus Academy.

Cambaudon, S. (2018). *MySQL 5.7: administración y optimización*. Barcelona: Ediciones Eni.

Casillas Santillán, L. A., Pérez Mora, Ó., & Gibert Ginestà, M. (2014). *Bases de datos en MySQL*. Cataluña: Universitat oberta de Catalunya.

Christudas, B. (2019). *MySQL. In: Practical Microservices Architectural Patterns*. Apress, Berkeley. Apress.

Gilat, A. (2006). *Matlab: Una introducción con ejemplos prácticos*. Ohio: Reverté.

Oracle. (2023). *MySQL*. Obtenido de <https://www.mysql.com/>

Suehring, S. (2001). *MySQL Bible*. New York: Wiley Publishing, Inc.

The MathWorks, Inc. (2023). *Mathworks*. Obtenido de [la.mathworks.com: https://la.mathworks.com/products/matlab.html](https://la.mathworks.com/products/matlab.html)