

Universidad Mariano Gálvez de Guatemala

Facultad de ingeniería en sistemas

Campus Villa nueva, Guatemala

Curso: Programación I

Docente: Inge. Carlos Arias



Actividad: CRUD

Nombre: Stephanie Cristina Sabán Cárcamo

Carnet: 5090- 23-11167

Sección: "A"

Fecha: 29/04/24

## INTRODUCCIÓN

En el desarrollo de aplicaciones modernas, el manejo y la gestión de datos es un aspecto fundamental. Las operaciones CRUD (Crear, Leer, Actualizar y Eliminar) son la base de cualquier sistema que involucre el almacenamiento y manipulación de información persistente. Estas operaciones permiten interactuar con bases de datos o sistemas de archivos de manera efectiva, brindando a los usuarios la capacidad de agregar, consultar, modificar y eliminar registros de acuerdo a sus necesidades.

Visual Basic, un lenguaje de programación ampliamente utilizado en el desarrollo de aplicaciones de escritorio y web, ofrece una sólida integración con diversos sistemas de gestión de bases de datos, incluyendo MySQL. Esta combinación de tecnologías permite crear soluciones robustas y escalables para una amplia gama de escenarios, desde pequeñas aplicaciones hasta sistemas empresariales complejos.

El presente trabajo tiene como objetivo explorar y ejemplificar la implementación de operaciones CRUD en Visual Basic, utilizando MySQL como sistema de gestión de bases de datos. Se abordarán los conceptos fundamentales, las técnicas y las mejores prácticas para establecer una conexión segura con la base de datos, ejecutar consultas SQL y manipular los datos de manera eficiente.

Además, se analizarán los diferentes componentes y herramientas disponibles en Visual Basic para facilitar el acceso a datos, como los objetos de conexión, comandos y adaptadores de datos. Se presentarán ejemplos prácticos que ilustren la implementación de las operaciones CRUD en el contexto de una aplicación de escritorio o web, utilizando controles de interfaz de usuario adecuados.








MySQL es un sistema de gestión de bases de datos relacional ampliamente utilizado que permite almacenar y administrar datos de manera eficiente. Visual Studio Code, por otro lado, es un editor de código ligero y altamente personalizable que ofrece una amplia gama de extensiones para diferentes propósitos de desarrollo. Al integrar MySQL con Visual Studio Code mediante extensiones especializadas, los desarrolladores pueden aprovechar las funcionalidades de ambos entornos para interactuar con bases de datos MySQL de forma conveniente y eficaz directamente desde su entorno de desarrollo, lo que facilita la creación, gestión y optimización de aplicaciones basadas en bases de datos.

## ¿QUE ES MYSQL WORKBENCH?

MySQL Workbench es una herramienta visual y un entorno integrado de desarrollo diseñado para trabajar con bases de datos MySQL. Desarrollado por Oracle, este software se ha convertido en la elección preferida de desarrolladores y administradores de bases de datos gracias a su interfaz intuitiva y sus robustas funcionalidades.

## ¿PARA QUE ES UTIL MYSQL?

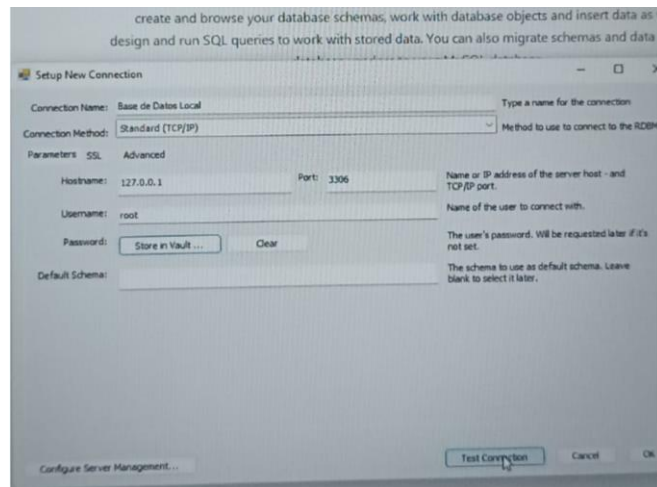
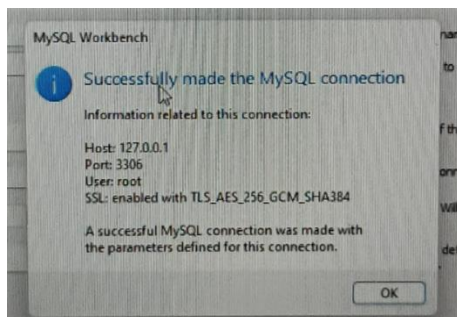
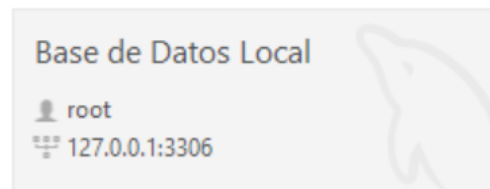
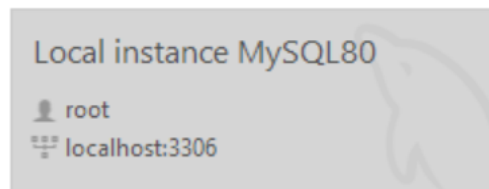
MySQL tiene varias funcionalidades, como las siguientes:

-  **Diseño y modelado de bases de datos:** Una de las funciones estrella de MySQL Workbench es su capacidad para diseñar y modelar bases de datos de manera visual. Los usuarios pueden crear esquemas, tablas y relaciones de manera gráfica.
-  **Desarrollo de consultas SQL:** MySQL Workbench facilita el desarrollo y la ejecución de consultas SQL directamente desde su interfaz.
-  **Administración de usuarios y privilegios:** La herramienta le permite a los administradores de bases de datos gestionar usuarios y sus privilegios de manera centralizada.
-  **Migración de datos:** MySQL Workbench simplifica el proceso de migración de datos entre diferentes bases de datos. Facilita la transferencia de datos de un origen a otro, asegurando una migración sin problemas y sin pérdida de información.
-  **Monitoreo y optimización del rendimiento:** La herramienta ofrece capacidades de monitoreo en tiempo real que permiten a los usuarios supervisar el rendimiento de la base de datos.
-  **Generación de informes visuales:** MySQL Workbench ofrece la generación de informes visuales que facilitan la comprensión y el análisis de datos.
-  **Automatización de tareas:** Con funciones de automatización, como la planificación de copias de seguridad y tareas administrativas recurrentes, simplifica la gestión diaria de bases de datos, lo que le permite a los usuarios centrarse en el desarrollo y la mejora continua.

## CONEXIONES DE MYSQL A VISUAL BASIC CODE

Para empezar creamos una conexión desde mysql workbench como la siguiente donde nos pedirá nombre host, contraseña y ruta.

## MySQL Connections



Luego desde el CMD ponemos algunos comandos dirigiéndose para una carpeta en específico, para crear una contraseña con su usuario desde allí y luego ya en visual hacer la conexión.

```
General-20240424_084253-Mee: x +
Simbolo del sistema - mysql x + v
C:\Program Files\MySQL\MySQL Server 8.0\bin>mysql -u root -p
Enter password: *****
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 21
Server version: 8.0.36 MySQL Community Server - GPL

Copyright (c) 2000, 2024, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

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

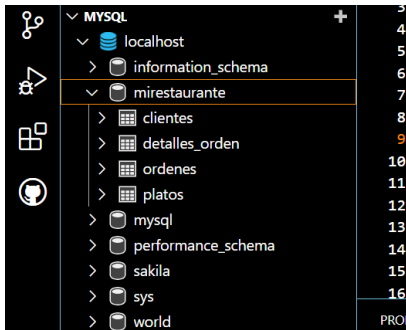
mysql> CREATE USER 'sqluser'@'%' IDENTIFIED WITH mysql_native_password BY 'password';
Query OK, 0 rows affected (0.02 sec)

mysql> GRANT ALL PRIVILEGES ON . TO 'sqluser'@'%';
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version
for the right syntax to use near '. TO 'sqluser'@'%' at line 1
mysql> CREATE USER 'sqluser'@'%' IDENTIFIED WITH mysql_native_password BY 'password';
ERROR 1396 (HY000): Operation CREATE USER failed for 'sqluser'@'%'
mysql> GRANT ALL PRIVILEGES ON *.* TO 'sqluser'@'%';
Query OK, 0 rows affected (0.01 sec)

mysql> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.01 sec)

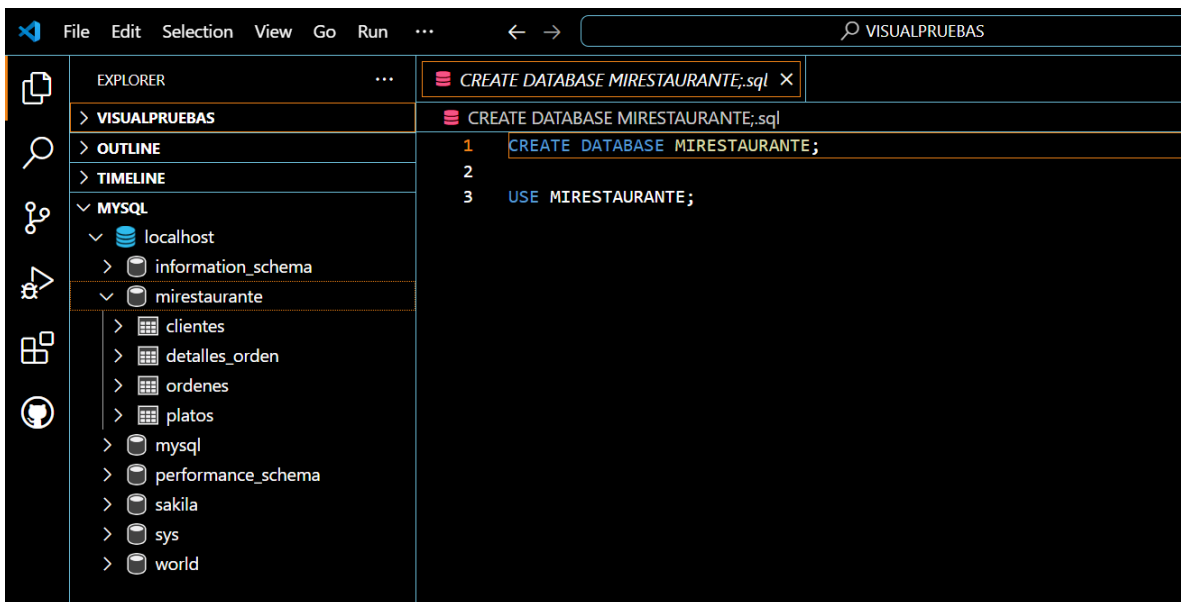
mysql> |
```

Luego ya ingresamos el usuario y la contraseña que acabamos de crear y luego ya tendremos la conexión completada.

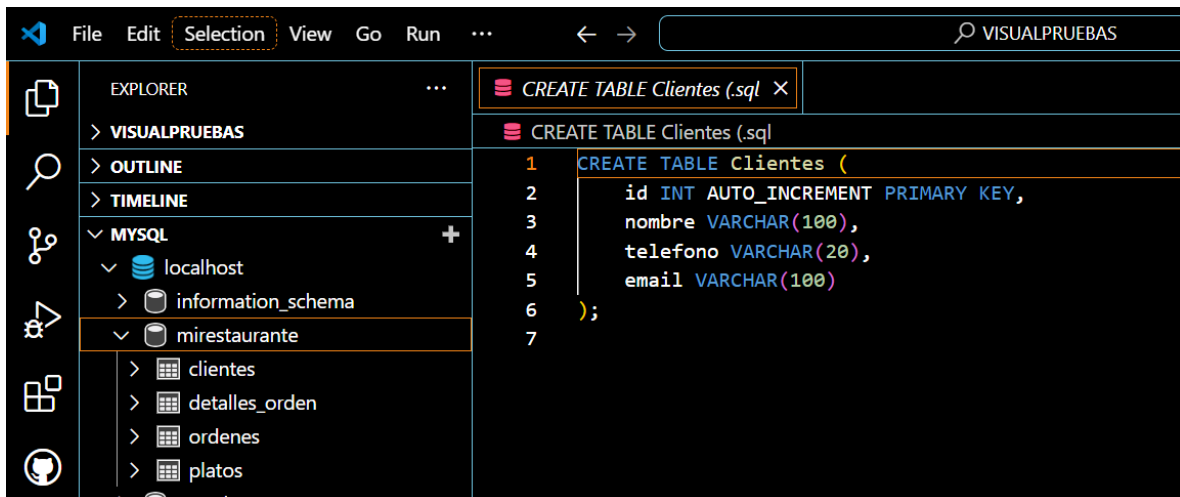


## CRUD

### Crear



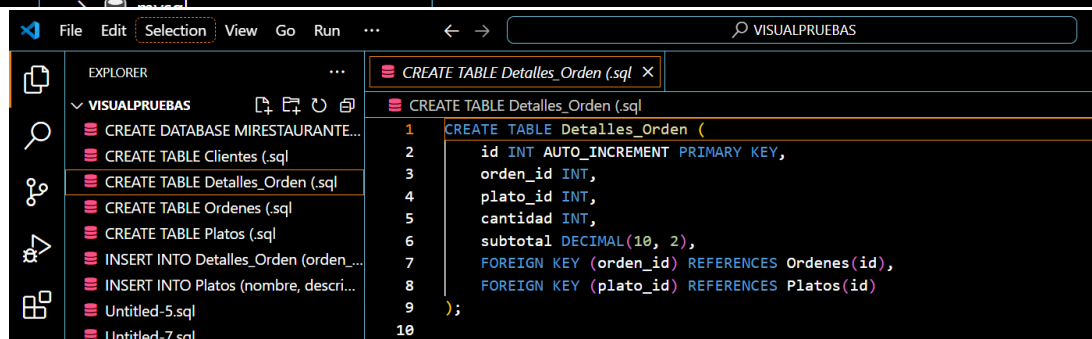
Creación de información como la creación de database y las tablas correspondientes para un restaurante como clientes, sus ordenes, el detalle de las ordenes y los platos.



```
File Edit Selection View Go Run ... VISUALPRUEBAS

EXPLORER
  > VISUALPRUEBAS
  > OUTLINE
  > TIMELINE
  > MYSQL
    > localhost
      > information_schema
      > mirestaurante
        > clientes
        > detalles_order
        > ordenes
        > platos

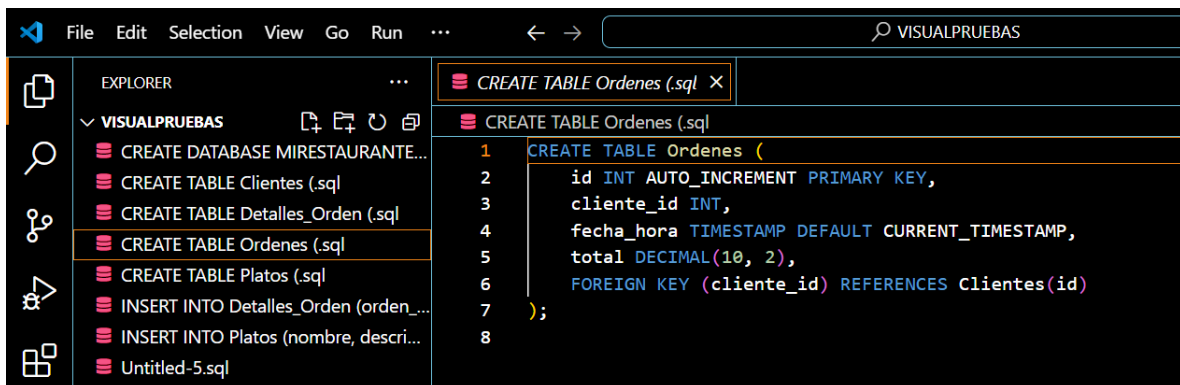
CREATE TABLE Clientes (.sql X)
CREATE TABLE Clientes (.sql)
1 CREATE TABLE Clientes (
2     id INT AUTO_INCREMENT PRIMARY KEY,
3     nombre VARCHAR(100),
4     telefono VARCHAR(20),
5     email VARCHAR(100)
6 );
7
```



```
File Edit Selection View Go Run ... VISUALPRUEBAS

EXPLORER
  > VISUALPRUEBAS
    CREATE DATABASE MIRESTaurante...
    CREATE TABLE Clientes (.sql)
    CREATE TABLE Detalles_Order (.sql)
    CREATE TABLE Ordenes (.sql)
    CREATE TABLE Platos (.sql)
    INSERT INTO Detalles_Order (orden_...
    INSERT INTO Platos (nombre, descri...
    Untitled-5.sql
    Untitled-7.sql

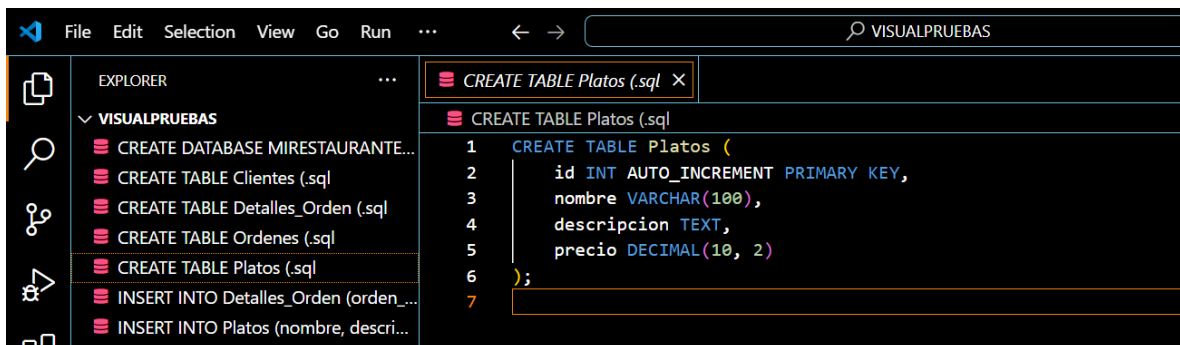
CREATE TABLE Detalles_Order (.sql X)
CREATE TABLE Detalles_Order (.sql)
1 CREATE TABLE Detalles_Order (
2     id INT AUTO_INCREMENT PRIMARY KEY,
3     orden_id INT,
4     plato_id INT,
5     cantidad INT,
6     subtotal DECIMAL(10, 2),
7     FOREIGN KEY (orden_id) REFERENCES Ordenes(id),
8     FOREIGN KEY (plato_id) REFERENCES Platos(id)
9 );
10
```



```
File Edit Selection View Go Run ... VISUALPRUEBAS

EXPLORER
  > VISUALPRUEBAS
    CREATE DATABASE MIRESTaurante...
    CREATE TABLE Clientes (.sql)
    CREATE TABLE Detalles_Order (.sql)
    CREATE TABLE Ordenes (.sql)
    CREATE TABLE Platos (.sql)
    INSERT INTO Detalles_Order (orden_...
    INSERT INTO Platos (nombre, descri...
    Untitled-5.sql

CREATE TABLE Ordenes (.sql X)
CREATE TABLE Ordenes (.sql)
1 CREATE TABLE Ordenes (
2     id INT AUTO_INCREMENT PRIMARY KEY,
3     cliente_id INT,
4     fecha_hora TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
5     total DECIMAL(10, 2),
6     FOREIGN KEY (cliente_id) REFERENCES Clientes(id)
7 );
8
```



```
File Edit Selection View Go Run ... VISUALPRUEBAS

EXPLORER
  > VISUALPRUEBAS
    CREATE DATABASE MIRESTaurante...
    CREATE TABLE Clientes (.sql)
    CREATE TABLE Detalles_Order (.sql)
    CREATE TABLE Ordenes (.sql)
    CREATE TABLE Platos (.sql)
    INSERT INTO Detalles_Order (orden_...
    INSERT INTO Platos (nombre, descri...

CREATE TABLE Platos (.sql X)
CREATE TABLE Platos (.sql)
1 CREATE TABLE Platos (
2     id INT AUTO_INCREMENT PRIMARY KEY,
3     nombre VARCHAR(100),
4     descripcion TEXT,
5     precio DECIMAL(10, 2)
6 );
7
```

Luego insertamos los datos en cada tabla

```
View Go Run ... VISUALPRUEBAS

... INSERT INTO Detalles_Orden (orden_id, pl.sql X
INSERT INTO Detalles_Orden (orden_id, pl.sql
1 INSERT INTO Detalles_Orden (orden_id, plato_id, cantidad, subtotal) VALUES
2 (1, 1, 2, 17.98),
3 (1, 3, 1, 5.49),
4 (2, 2, 1, 6.49),
5 (3, 3, 1, 5.99),
6 (4, 4, 1, 9.99),
7 (4, 5, 1, 12.99),
8 (5, 1, 1, 8.99),
9 (5, 2, 1, 6.49);
10
```

```
View Go Run ... VISUALPRUEBAS

... INSERT INTO Platos (nombre, descripcion,.sql X
INSERT INTO Platos (nombre, descripcion,.sql
1 INSERT INTO Platos (nombre, descripcion, precio) VALUES
2 ('Pizza Margarita', 'Pizza con salsa de tomate, mozzarella y albahaca', 8.99),
3 ('Hamburguesa Clásica', 'Hamburguesa de carne de res con lechuga, tomate y cebolla', 6.49),
4 ('Ensalada César', 'Ensalada de lechuga romana con aderezo César y crutones', 5.99),
5 ('Pasta Alfredo', 'Pasta con salsa Alfredo y pollo a la parrilla', 9.99),
6 ('Sushi Variado', 'Sushi de salmón, atún y camarón', 12.99);
7
```

```
... VISUALPRUEBAS

Untitled-5.sql X
Untitled-5.sql
1 INSERT INTO Clientes (nombre, telefono, email) VALUES
2 ('Juan Pérez', '123456789', 'juan@example.com'),
3 ('María García', '987654321', 'maria@example.com'),
4 ('Pedro López', '555123456', 'pedro@example.com'),
5 ('Laura Martínez', '666789456', 'laura@example.com'),
6 ('Carlos Rodríguez', '999888777', 'carlos@example.com');
7
```

```
Run ... VISUALPRUEBAS

Untitled-7.sql X
Untitled-7.sql
1 INSERT INTO Ordenes (cliente_id, total) VALUES
2 (1, 25.47),
3 (2, 18.98),
4 (3, 11.50),
5 (4, 32.75),
6 (5, 22.60);
7
```

Luego podemos leer los datos y visualizarlos usando SELECT y colocando de donde vienen los datos en este caso cada tabla.

```
9  (5, 2, 1, 6.49);
10
11  -- ACA CONSULTAMOS LOS DATOS
12  SELECT * FROM CLIENTES;
13
14  SELECT * FROM ORDENES;
15
16  SELECT * FROM PLATOS;
17
18  SELECT * FROM DETALLES_ORDEN;
19
```

Results

id	nombre	telefono	email
1	Juan Pérez	123456789	juanperez@gmail.com
2	María García	987654321	maria@example.com
3	Pedro López	555123456	pedro@example.com
4	Laura Martínez	666789456	lauramartinez@gmail.com

Results

id	cliente_id	fecha_hora	total
1	1	Mon Apr 29 2024 09:23:37 GMT-0600 (Central Standard Time)	25.47
2	2	Mon Apr 29 2024 09:23:37 GMT-0600 (Central Standard Time)	18.98
3	3	Mon Apr 29 2024 09:23:37 GMT-0600 (Central Standard Time)	11.5
4	4	Mon Apr 29 2024 09:23:37 GMT-0600 (Central Standard Time)	32.75

Visualizar

SELECT \* FROM `mirestaurante`.`detalles\_orden`

Results

id	orden_id	plato_id	cantidad	subtotal
1	1	1	2	17.98
2	1	3	1	5.49
3	2	2	1	6.49
4	3	3	1	5.99
5	4	4	1	9.99
6	4	5	1	12.99



Results			
id	nombre	descripcion	precio
1	Pizza Margarita	Pizza con salsa de tomate, mozzarella y albahaca	8.99
2	Hamburguesa Clásica	Hamburguesa de carne de res con lechuga, tomate y cebolla	6.49
3	Ensalada César	Ensalada de lechuga romana con aderezo césar y crutones	5.99
4	Pasta Alfredo	Pasta con salsa Alfredo y pollo a la parrilla	9.99
5	Sushi Variado	Sushi de salmón, atún y camarón	12.99

Por ultimo actualizar y eliminar datos, para eliminar datos es necesario ver que esos datos no estenconectados con otra tabla o con otros datos importantes.

```

20  -- ACA ACTUALIZAMOS LOS DATOS
21
22  update clientes set email= 'juanperez@gmail.com' where id=1;
23
24  update clientes set email= 'lauramartinez@gmail.com' where id=4;
25
26  -- ACA BORRAMOS DATOS DE LA TABLA
27
28  delete from DETALLES_ORDEN where id=7;
29  delete from DETALLES_ORDEN where id=8;
30  delete from ORDENES where id=5;
31  delete from clientes where id=5;

```

## CONCLUSIÓN

A lo largo de este trabajo, hemos explorado la implementación de operaciones CRUD (Crear, Leer, Actualizar y Eliminar) en aplicaciones desarrolladas con Visual Basic y MySQL. Esta combinación de tecnologías ha demostrado ser una solución poderosa y flexible para el manejo de datos persistentes en diversos escenarios.

Mediante ejemplos prácticos, hemos visto cómo Visual Basic proporciona una amplia gama de herramientas y componentes para facilitar la interacción con bases de datos MySQL. Desde establecer conexiones seguras hasta ejecutar consultas SQL y manipular los datos, el marco de trabajo de .NET ofrece una experiencia de desarrollo fluida e intuitiva.

Si bien este trabajo ha cubierto los conceptos básicos y las implementaciones esenciales, existen numerosas áreas de exploración adicionales, como la optimización de consultas, el manejo de transacciones, la implementación de patrones de diseño y la integración con tecnologías web modernas. Estas áreas brindan oportunidades emocionantes para seguir profundizando y ampliando los conocimientos en el desarrollo de aplicaciones de manejo de datos con Visual Basic y MySQL.

## REFERENCIA

<https://github.com/StephanieSabann/CRUD.git>