

## 1. TÍTULO

**OPTIMIZACIÓN DEL DISEÑO DE LA BASE DE DATOS DE LA  
EMPRESA AGROINSUMOS DEL SUR S.A**

**OPTIMIZATION OF THE DESIGN OF THE DATABASE OF THE  
COMPANY AGROINSUMOS DEL SUR S.A**

## 2. AUTORES

**MADELEMNE MARIANA MAFARE CALVA**

**Estudiante de la Carrera de Desarrollo de Software**

[marymafare@gmail.com](mailto:marymafare@gmail.com)

**MYRIAM ARACELY CAIZA ANALUISA**

**Ingeniería en Sistemas Informáticos**

[acaiza@istmas.edu.ec](mailto:acaiza@istmas.edu.ec)

**NANCY MARITZA MONTOYA RAMÍREZ**

**Magister en Informática Educativa**

[nmontoya@istmas.edu.ec](mailto:nmontoya@istmas.edu.ec)

## 3. RESUMEN

El objetivo del trabajo de investigación es optimizar el diseño de la base de datos de la empresa AGROINSUMOS DEL SUR S.A, a partir de la normalización y definición de relaciones, que se alineen a las necesidades y operaciones de la misma.

El enfoque de investigación es cualitativo ya que a través de entrevistas y reuniones de trabajo se obtuvieron datos que permiten una mejor comprensión de los requerimientos, en cuanto al manejo de stock, clientes



098 81 83 081



03-2961680



Loja 31-15 y Buenos Aires, Riobamba

y proveedores.

En cuanto a la generación de consultas, permite a los usuarios obtener información específica sobre los stock, clientes y proveedores, la retroalimentación de los usuarios ha facilitado la toma de decisiones informadas y ágiles.

### **Palabras Claves**

BASE DE DATOS, MySQL, DDL, DML, INNER JOIN

### **Keywords**

DATABASE, MySQL, DDL, DML, INNER JOIN

## **4. INTRODUCCIÓN**

En la actualidad, la eficiente gestión de datos se ha convertido en un factor decisivo para el éxito de una empresa; en este contexto, AGROINSUMOS DEL SUR S.A, busca ser líder en el sector agroindustrial y reconoce la importancia de contar con una base de datos personalizada y optimizada que cubra sus necesidades.

Este proyecto inicia con la identificación de los requisitos únicos de la empresa hasta el diseño y la implementación de una base de datos personalizada en cuanto al manejo de stock, clientes y proveedores. También se generan consultas, que buscan proporcionar a la empresa una herramienta que no solo almacene datos de manera eficiente, sino que también permita acceder a información de manera rápida y precisa, sentando las bases para una toma de decisiones dentro de AGROINSUMOS DEL SUR S.A.



## **5. DESARROLLO**

### **Marco teórico**

#### **Base de Datos**

Según Moreira Centeno et al.(2022),” El término base de datos hace referencia a un conjunto de datos multidimensionales en el sentido de que los enlaces internos existentes entre los distintos elementos hacen que se pueda acceder a la información con diversas perspectivas”.

Una base de datos es un sistema organizado que permite recopilar, almacenar y gestionar la información. Esta información se encuentra alojada en diversas tablas que forman parte de la estructura de la base de datos.

#### **Modelo Entidad Relación**

El modelo entidad-relación, también conocido como diseño conceptual, es un enfoque para diseñar bases de datos relacionales que permite visualizar de manera gráfica la estructura mediante entidades, atributos y relaciones. Este modelo facilita la interacción entre los elementos, siendo el primer proceso que se desarrolla al pretender diseñar una base de datos.

También, en opinión de Moreira Centeno et al.(2022) menciona que “Un diagrama relacional es una representación de las entidades de la base de datos relacional, los atributos dentro de esas entidades y las relaciones entre esas entidades.”

#### **Gestor de Base de Datos MySQL**

Desde el punto de vista de Pilicita Garrido , Borja Lopez , & Gutiérrez Constante (2020) citado por Cornejo Manzo (2022), “Un sistema de gestión de base de datos es un software que oculta la complejidad de gestión de los datos al desarrollador.”

Para la presente investigación el gestor seleccionado fue MYSQL que como



sostiene Flores( 2018) citado por (Cornejo Manzo, 2022), "Es la base de datos Open Source más usada a nivel mundial, se caracteriza por su rapidez en las operaciones y su buen rendimiento, posee una baja probabilidad de corromper los datos."

MySQL es uno de los gestores de bases de datos más utilizados en el desarrollo de aplicaciones web y otras aplicaciones. Gracias a su administrador, PHPMyAdmin, la gestión resulta menos compleja para el programador debido a su interfaz gráfica y la posibilidad de utilizar comandos SQL. MySQL es un software gratuito y fácil de instalar, especialmente a través del servidor local XAMPP.

## **Comandos SQL(Structured Query Language)**

SQL es un lenguaje de consulta estructurado que accede a bases de datos relacionales ejecutando diferentes tipos de operaciones tanto en la estructura como en la manipulación de los datos. Es decir, en ciertos casos, se pueden realizar consultas que permiten recuperar de forma sencilla la información de interés en un conjunto de datos e incluso efectuar modificaciones en ciertos datos de manera rápida. (docplayer, 2023)

## **Comandos DDL**

El Lenguaje de Definición de Datos (DDL por sus siglas en inglés, Data Definition Language) es un conjunto de instrucciones que permite la creación de la base de datos y la definición de la estructura de los objetos dentro de la base de datos. Esto incluye la creación de tablas junto con sus campos e índices. Además, abarca las operaciones de modificación y eliminación de tablas, donde se almacenan los datos de la base de datos. En la investigación, se emplean cuatro operaciones básicas de comandos DDL, a saber: CREATE, ALTER y DROP. A continuación, se explica la función de cada uno de ellos.

## **CREATE**

El comando create permite crear la base de datos, tablas, trigger, vistas y procedimientos almacenados.



Ejemplo:

```
CREATE TABLE 'NOMBRE_TABLA';
```

### **ALTER**

Este comando permite modificar la estructura de una tabla, agregando o quitando campos, modificar el tipo de dato del campo, modificar los índices e incluso modificar un trigger.

Ejemplo

```
ALTER TABLE 'NOMBRE_TABLA' ADD NUEVO_CAMPO TIPO_DATO;
```

### **DROP**

Con el comando drop se elimina un objeto de la base de datos tales como: tablas, vistas, índices, trigger, procedimientos almacenados incluso si se esta administrando de forma local puede eliminar con el DROP la base de datos.

```
DROP TABLE 'NOMBRE_TABLA';
```

### **Comando DML**

Lenguaje de manipulación de datos (Data Manipulation Language, o DML en inglés) es un lenguaje para base de datos relacional que permite a los usuarios manipular los datos , ejecutar diferentes consultas de interés para los usuarios de una manera rápida y organizada. (docplayer, 2023)

Lenguaje de manipulación de datos (Data Manipulation Language, o DML en inglés) es un lenguaje para base de datos relacional que permite a los usuarios manipular los datos , ejecutar diferentes consultas de interés para los usuarios de una manera rápida y organizada. (docplayer, 2023)

### **SELECT**

Con este comando permite consultar los datos almacenados en una tabla de la base de datos sea ciertos campos o todos los campos de la tabla.



Forma

```
SELECT `columna1`, `columna3`, `columna4` FROM `NOMBRE_TABLA`  
`WHERE `columna1`>0
```

## **INSERT**

La sentencia INSERT de SQL agrega uno o varios registros a una tabla .

Forma

```
INSERT INTO 'NOMBRE_TABLA'('columna1','columna2,...')  
VALUES('valor1',['valor2,...'])
```

## **UPDATE**

La sentencia UPDATE permite modificar los valores de los registros de la base de datos

Forma

```
UPDATE NOMBRE_TABLA SET Columna1='NUEVO VALOR' WHERE  
Columna2= 'VALOR'
```

## **DELETE**

El comando DELETE borra uno o varios registros de la tabla.

Forma

```
DELETE FROM TABLA WHERE COLUMNA1 ='valor1'
```

## **INNER JOIN**

Se diferencia de todas las demás declaraciones JOIN en que muestra un conjunto de resultados mínimo, ya que en el producto cruzado solo aparecen los registros que satisfacen los criterios de selección y se visualiza en una tabla de resultados llamada "View" que no contiene valores NULL. (Iñón, 2019). Es decir cuando se realiza una consulta con INNER JOIN que une dos o más tablas por medio de un enlace (Clave foránea) el resultado de la consulta se visualiza en una tabla que no se crea dentro de la base de datos



sino que tan solo permite visualizar la consulta resultante como si fuera una sola tabla es por ello el nombre a la tabla de "VIEW".

## Ejemplo

```
SELECT aviones.codigo , id_vuelos.destino from aviones INNER JOIN  
id_vuelos ON aviones.id = id_vuelos.avion_id;
```

## Objetivos

### Objetivo General

Optimizar el diseño de la base de datos de la empresa AGROINSUMOS DEL SUR S.A, que se alinee a sus necesidades y operaciones.

### Objetivos Específicos

Identificar los requerimientos de la empresa AGROINSUMOS DEL SUR S.A para un análisis detallado del manejo de stock.

Diseñar una base de datos personalizada que cumpla con los requerimientos de la empresa.

Generar consultas en la base de datos de acuerdo a las necesidades de la empresa.

### Materiales y Metodología

El enfoque de investigación es cualitativo ya que, a través de entrevistas, se obtuvo datos que permitieron una comprensión más amplia para el diseño de la base de datos, en cuanto al diseño de investigación fue una investigación aplicada, ya que se resolvió un problema específico.

A través de reuniones de trabajo con los propietarios se realizó la requerimentación de las necesidades de la empresa, en cuanto al manejo de stock, clientes y proveedores; después se procedió al diseño del modelo entidad - relación de la base de datos.

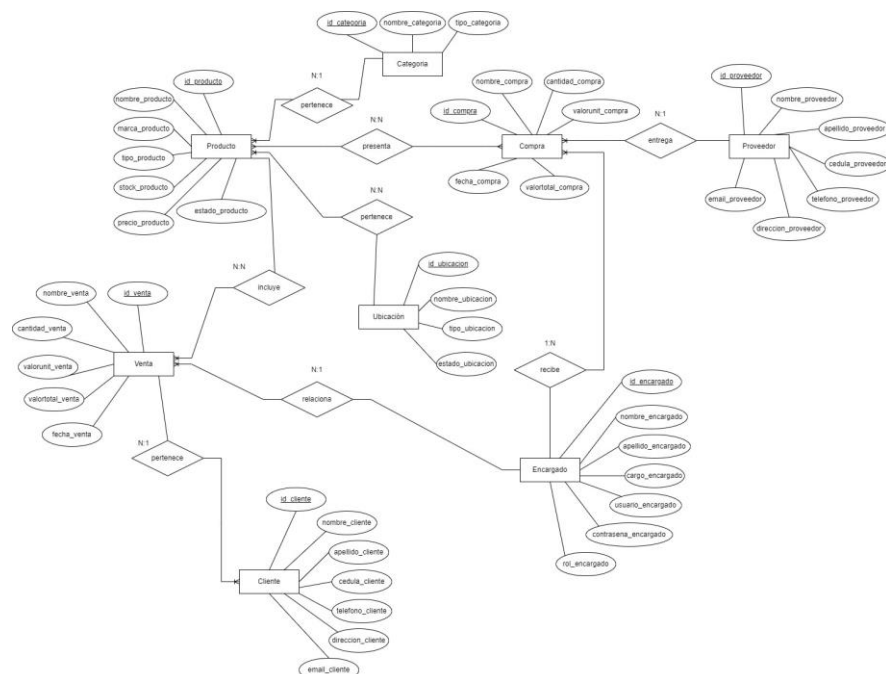


AGROINSUMOS DEL SUR S.A es una pequeña empresa que se especializa en la venta de insumos agrícolas, como semillas, fertilizantes, pesticidas y herramientas para agricultores locales. Trabajando en estrecha colaboración con los clientes para proporcionar soluciones agrícolas efectivas y mejorar la producción de cultivos en la región.

Existen inconvenientes en el control de stock de los insumos agrícolas, así como también con los clientes y proveedores, ya que dicha información se lleva manualmente y ha existido pérdida de información que repercute en el servicio que la empresa brinda a sus clientes.

Una vez obtenida la información se procedió al diseño del diagrama entidad - relación de la base de datos, el mismo que se muestra a continuación:

**Figura 1: Diagrama Entidad Relación**

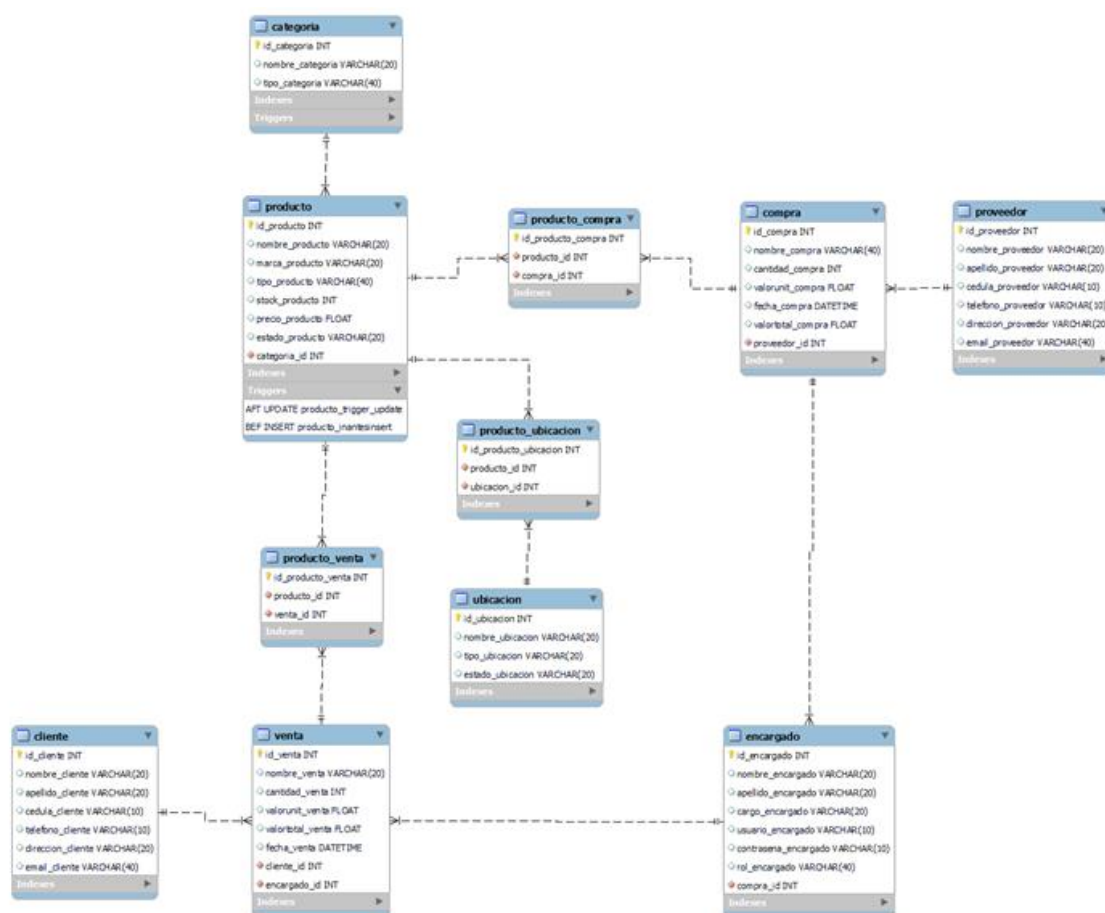


Nota. Elaboración propia



Una vez obtenido el diagrama entidad - relación, se utilizó el gestor de base de datos MySQL, generando 11 tablas, con las respectivas consultas que permitieron verificar el buen funcionamiento y optimización de la base de datos. A continuación, se presenta las mencionadas tablas:

**Figura 2: Diseño Físico**



Nota. Elaboración propia

Asimismo, se generaron las consultas considerando los comandos DDL y DML.

**Comandos DDL.** - Se coloca cada uno de los comandos DDL utilizados para la estructura de la base de datos en el gestor de base de datos MYSQL.

## CREATE

La siguiente sentencia permite crear la base de datos de manera local



098 81 83 081



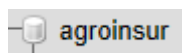
03-2961680



Loja 31-15 y Buenos Aires, Riobamba

CREATE DATABASE agroinsur;

**Figura 3:** Ícono de la base de datos

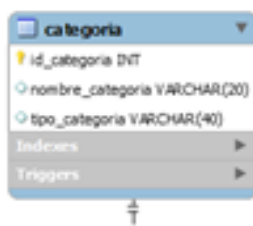


Nota. Elaboración propia

Una vez creada la base de datos, se inicia con una sentencia SQL para crear tablas, como la que se muestra a continuación para la creación de la tabla "categoría".

```
CREATE TABLE `categoria` (  
  `id_categoria` int(11) NOT NULL,  
  `nombre_categoria` varchar(20) DEFAULT NULL,  
  `tipo_categoria` varchar(40) DEFAULT NULL  
);
```

**Figura 4:** Tabla Categoría



Nota. Elaboración propia

## DROP

La siguiente sentencia elimina una tabla .

```
DROP TABLE categoria;
```

## ALTER

Con el siguiente comando se puede añadir un campo a la tabla



098 81 83 081



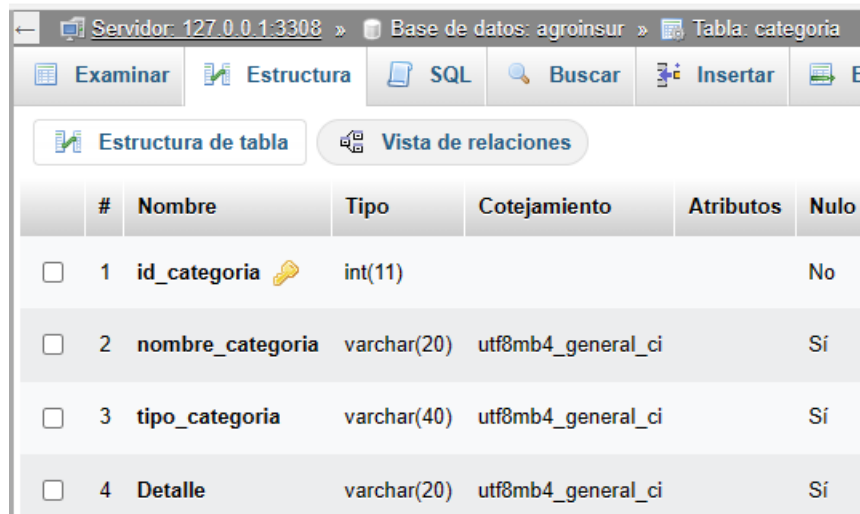
03-2961680



Loja 31-15 y Buenos Aires, Riobamba

ALTER TABLE `categoria` ADD `Detalle` varchar(20);

**Figura 5:** Campos de la tabla categoria



| # | Nombre           | Tipo        | Cotejamiento       | Atributos | Nulo |
|---|------------------|-------------|--------------------|-----------|------|
| 1 | id_categoria     | int(11)     |                    |           | No   |
| 2 | nombre_categoria | varchar(20) | utf8mb4_general_ci |           | Sí   |
| 3 | tipo_categoria   | varchar(40) | utf8mb4_general_ci |           | Sí   |
| 4 | Detalle          | varchar(20) | utf8mb4_general_ci |           | Sí   |

Nota. Elaboración propia

Las siguientes sentencias como el trigger, vistas y procedimientos almacenados abarca los dos grupos de comandos DDL y DML pero al tener su término inicial CREATE se coloca dentro del bloque de grupo de comando DDL con lo fundamentado en el marco teórico.

## TRIGGER O DISPARADORES

Este tipo de comandos permiten actualizar el stock o manejar un proceso de autoría como se visualiza en el siguiente comando. Cada vez que se inserte un producto se registrar la hora exacta en que se ingreso dicha información en una tabla denominada auditoria\_producto, esto permite tener control sobre el registro de los productos al inventario ya que todos deben pasar por un proceso de contabilidad antes de ser registrado y por lo cual su registro se encontrará dentro de los horarios administrativos según políticas de la institución.

CREATE TRIGGER producto\_inantesinsert BEFORE INSERT ON producto FOR EACH ROW



098 81 83 081



03-2961680



Loja 31-15 y Buenos Aires, Riobamba

```
INSERT INTO auditoria_producto (nombre_producto, precio_producto,  
tiempo_registro) VALUES (NEW.nombre_producto, NEW.precio_producto,  
NOW())
```

**Figura 6:** Tabla auditoria\_producto



Nota. Elaboración propia

**Figura 7:** Datos registrados en la tabla auditoria\_producto

| nombre_producto | precio_producto | tiempo_registro     |
|-----------------|-----------------|---------------------|
| Motosierra      | 100.1           | 2023-10-31 20:10:34 |

Nota. Elaboración propia

## VISTAS

La sentencia vista es creada principalmente para ciertos reportes que la empresa requiere de una respuesta rápida para la toma de decisiones. La siguiente sentencia permite visualizar la cédula y nombre de los clientes.

### Creación de la vista

```
CREATE VIEW vista_refore_cliente AS SELECT  
cedula_cliente, nombre_cliente FROM bd_inventario_agroinsur.cliente WHERE  
id_cliente > 1
```

Llamado a la vista

```
SELECT * FROM `vista_refore_cliente`;
```

La visualización de la cédula y nombre de los clientes es un reporte muy común por lo que debe ser creado por medio de una vista; por lo tanto, cuando se desee visualizar este tipo de reporte tan sólo se debe llamar a la vista y no digitar una



serie de líneas de comando SQL.

**Figura 8:** Reporte de la vista `vista\_refore\_cliente`

| cedula_cliente | nombre_cliente |
|----------------|----------------|
| 0804756693     | Jhon Henry     |
| 172536894      | Daniela Martha |

Nota. Elaboración propia

## PROCEDIMIENTOS ALMACENADOS

Los procedimientos almacenados son como funciones los cuales pueden enviar o no ciertos parámetros para que se ejecute una operación. El siguiente procedimiento almacenado permite a la empresa visualizar un reporte de la cédula y nombre del cliente donde el id del cliente que se desea consultar se envía por parámetro.

Creación

```
CREATE PROCEDURE procedimiento_reporte_cliente_dinamic (id_cliente_p int)
SELECT cedula_cliente, nombre_cliente FROM cliente WHERE id_cliente =
id_cliente_p;
```

Llamado al procedimiento almacenado

```
CALL procedimiento_reporte_cliente_dinamic (1);
```

**Figura 9:** Reporte del procedimiento almacenado  
“procedimiento\_reporte\_cliente\_dinamic”

| cedula_cliente | nombre_cliente |
|----------------|----------------|
| 1804751863     | Esther Faviola |

Nota. Elaboración propia

## Comandos DML

Los siguientes comandos permiten la manipulación de los datos desde insertar información, modificarlo o consultarlo.

### INSERT

El comando INSERT permite un nuevo registro en cada una de las tablas. El siguiente comando permite ingresar una nueva categoría.

```
INSERT INTO `categoria` (`id_categoria`, `nombre_categoria`, `tipo_categoria`)
VALUES
(1, 'Bomba de Motor', 'Equipos de Riego')
```

**Figura 10:** Resultado de la ejecución de la sentencia INSERT

| ←T→                                                                                                                                                                                                                                                                                                       | id_categoria | nombre_categoria | tipo_categoria   | Detalle |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------|------------------|------------------|---------|
| <input type="checkbox"/>  Editar  Copiar  Borrar | 1            | Bomba de Motor   | Equipos de Riego | NULL    |

Nota. Elaboración propia

### UPDATE

El comando UPDATE permite la modificación de los datos en cada una de las tablas de la base de datos.

**Figura 11:** Reporte de la tabla cliente

| id_cliente | nombre_cliente | apellido_cliente | cedula_cliente | telefono_cliente | direccion_cliente  | email_cliente              |
|------------|----------------|------------------|----------------|------------------|--------------------|----------------------------|
| 1          | Esther Faviola | Moran Gutierrez  | 0803467788     | 0974334466       | Esmeraldas/Mirador | esthermafare2004@gmail.com |

Nota. Elaboración propia

El siguiente comando permite la modificación del número de cédula del cliente con id= 1

```
UPDATE `cliente` SET `cedula_cliente`='1804751863' WHERE `id_cliente`=1;
```

**Figura 12:** Resultado de la ejecución de la sentencia UPDATE

| id_cliente | nombre_cliente | apellido_cliente | cedula_cliente | telefono_cliente | direccion_cliente  | email_cliente              |
|------------|----------------|------------------|----------------|------------------|--------------------|----------------------------|
| 1          | Esther Faviola | Moran Gutierrez  | 1804751863     | 0974334466       | Esmeraldas/Mirador | esthermafare2004@gmail.com |

Nota. Elaboración propia

Es una sentencia muy utilizada por la empresa debido a que los clientes a menudo proporcionan un número de cédula incorrecto.

## DELETE

La siguiente sentencia permite eliminar un registro. La empresa lo utiliza para eliminar productos que no deben estar dentro de la base de datos.

```
DELETE FROM `producto` WHERE `id_producto`=2
```

Con la sentencia anterior se elimina de la base de datos el producto con id=2

## SELECT

En cuanto a la sentencia select la empresa lo utiliza para visualizar los diferentes reportes de interés.

Se tiene al SELECT \*FROM el cual reporta todos los campos de la tabla. La siguiente sentencia reporta todos los campos(columnas)de la tabla proveedor.

```
SELECT * FROM `proveedor` ;
```

**Figura 13:** Resultado de la ejecución de la sentencia SELECT \*

| id_proveedor | nombre_proveedor | apellido_proveedor | cedula_proveedor | telefono_proveedor | direccion_proveedor |
|--------------|------------------|--------------------|------------------|--------------------|---------------------|
| 1            | Dario Xavier     | Taisa Analuisa     | 1714698937       | 0952347651         | Quito/Chillo Gallo  |
| 2            | Martin Leonardo  | Bravo Moran        | 0805472261       | 0936345877         | Guayaquil/Pascuales |

Nota. Elaboración propia

Sin embargo se tiene el comando SELECT para elegir ciertos campos de la tabla que se desea conocer.

La siguiente sentencia muestra el nombre del proveedor seguido del apellido del proveedor y teléfono del proveedor cuyo número de cédula es =0805472261

```
SELECT `nombre_proveedor`, `apellido_proveedor`, `telefono_proveedor` FROM
`proveedor` WHERE `cedula_proveedor`="0805472261"
```

**Figura 14:** Resultado de la ejecución de la sentencia SELECT

| nombre_proveedor | apellido_proveedor | telefono_proveedor |
|------------------|--------------------|--------------------|
| Martin Leonardo  | Bravo Moran        | 0936345877         |

Nota. Elaboración propia

De esta manera el personal encargado del manejo de la base de datos , localiza los datos principales de un proveedor con tan solo disponer el número de cédula.

## INNER JOIN

El INNER JOIN utiliza la empresa para conocer donde está ubicado un producto teniendo en cuenta que en una tabla se encuentra los productos, en otra las ubicaciones dentro de la empresa para el inventario y una tabla resultante de la relación entre productos y ubicación, por lo tanto, al estar los campos que necesita

en su reporte en diferentes tablas es necesario la utilización de un inner join que permite unir en una tabla virtual.

La siguiente sentencia permite visualizar un reporte con el nombre del producto seguido de la ubicación teniendo como código de búsqueda el id=1(Id del producto)

```
SELECT producto.nombre_producto, ubicacion.nombre_ubicacion from producto  
INNER JOIN producto_ubicacion ON producto.id_producto  
=producto_ubicacion.producto_id INNER JOIN ubicacion ON  
producto_ubicacion.ubicacion_id = ubicacion.id_ubicacion where  
producto.id_producto=1;
```

**Figura 15:** Resultado de la ejecución de la sentencia INNER JOIN

| nombre_producto      | nombre_ubicacion |
|----------------------|------------------|
| Fumigadora Motor 26L | Deposito BMotor  |

Nota. Elaboración propia

Son consultas bastante recurrentes por parte del personal que maneja el inventario.

## 6. ANÁLISIS Y DISCUSIÓN

La gestión de stock, clientes y proveedores en AGROINSUMOS DEL SUR S.A exigió un cuidadoso diseño de la base de datos; el análisis de los requerimientos visibilizó la necesidad de una estructura que abarcara los distintos datos, desde detalles específicos de productos hasta la información detallada sobre clientes y proveedores. La normalización y la definición de relaciones se presentaron como elementos esenciales para garantizar un buen diseño de la base de datos.

El diseño se centró en la eficiencia operativa y flexibilidad, en el contexto de la gestión de inventarios la velocidad de acceso es importante ya que la información debe influir en las decisiones de reabastecimiento y en la atención al cliente; para (Párraga, 2023) MySQL tiene una capacidad de adaptación a grandes y variables



volúmenes de datos, lo que ha permitido la optimización del rendimiento, que han sido elementos clave para garantizar la utilidad práctica de la base de datos.

Por otra parte, la generación de consultas ha permitido la eficiencia operativa de la base de datos; la flexibilidad incorporada en las consultas permite a los usuarios obtener información específica sobre el stock, clientes y proveedores de manera rápida; la retroalimentación de los usuarios ha permitido recoger sus necesidades operativas diarias, facilitando la toma de decisiones informadas y ágiles; las sentencias SQL se optimizan de acuerdo al gestor de base de datos donde se ejecutan (Vicuña, Ponce, & Erazo, 2018).

## 7. CONCLUSIONES

- Se identificaron los requerimientos específicos de la empresa, cumpliendo con sus expectativas, ya que necesitan una base de datos que no solo almacene información en cuanto al stock, clientes y proveedores, sino que permita recuperación rápida de datos.
- La implementación de entidades, relaciones entre ellas, la normalización de tablas y la consideración de los tipos de datos permitieron el diseño de una base de datos eficiente.
- Las consultas generadas han sido efectivas ya que han proporcionado respuestas rápidas, permitiendo a los usuarios la obtención de información relevante.

## 8. REFERENCIAS BIBLIOGRÁFICAS

Cornejo Manzo, L. (2022). *Estudio Comparativo De Los Gestores De Bases De Datos MYSQL Y María Db: Caso Práctico Migración De Bases De Datos MYSQL A María Db*.  
<http://dspace.utb.edu.ec/bitstream/handle/49000/11715/E-UTB-FAFI-SIST-INF-000011.pdf?sequence=1&isAllowed=y>



Docplayer. (2023). *SQL(Structured Query Language)*.  
<https://docplayer.es/5730903-Sql-structured-query-language.html>

Ionos. (2019). *INNER JOIN: definición y aplicación*.  
<https://www.ionos.es/digitalguide/hosting/cuestiones-tecnicas/inner-join/>

Moreira Centeno, R., Almeida Zambrano, E., Mendoza Rodríguez, H., San Andrés Laz, E., & Mendoza Muñoz, K. (2022). *Análisis y Diseño de Base de Datos*. <https://munayi.uleam.edu.ec/wp-content/uploads/2022/07/Ana%CC%81lisis-y-disen%CC%83o-de-base-de-datos.pdf>

Párraga, C. (2023). *Análisis comparativo de los sistemas de gestión de base de datos MySQL y Oracle*. Obtenido de Universidad Técnica de Babahoyo: <http://dspace.utb.edu.ec/handle/49000/15036>

Vicuña, A., Ponce, J., & Erazo, O. (Julio de 2018). *Análisis de la optimización de sentencias SQL del Lenguaje de Consulta Estructurado usando grandes volúmenes de datos*. Obtenido de Revista Publicando: <chrome-extension://efaidnbmnnnibpcajpcgiclfindmkaj/https://core.ac.uk/download/pdf/236644673.pdf>

