

Proyecto final

BASE DE DATOS

ANDRADE SALAZAR, IGNACIO

GUERRERO SANTANA, EDGAR GEOVANNY

ORTIZ JIMENEZ, VLADIMIR

Contenido

[Definición del problema 2](#_Toc133929027)

[Objetivos 3](#_Toc133929028)

[Objetivo General: 3](#_Toc133929029)

[Objetivos específicos: 3](#_Toc133929030)

[Especificación de Entidades y atributos 4](#_Toc133929031)

[Diagrama Entidad-Relación 13](#_Toc133929032)

[Modelo Relacional 14](#_Toc133929033)

[Tablas 15](#_Toc133929035)

[Normalización 19](#_Toc133929036)

[Enunciados para las sentencias SQL 27](#_Toc133929037)

[Conclusiones 31](#_Toc133929038)

[Referencias 31](#_Toc133929039)

[Avances del sistema 32](#_Toc133929040)

## Definición del problema

El presente proyecto pretende desarrollar una herramienta con el uso de la tecnología que hoy en día se encuentra en todos lados y en cualquier actividad que hagamos, en la administración de los negocios ha resultado muy favorable el uso de ella para el control y manejo de recursos, así como para la reducción de problemas, es por ello que este proyecto está dirigido a los pequeños y medianos negocios que busquen una herramienta que les ayude a administrar sus entradas y salidas de mercancías, así como el cálculo neto de su utilidad. Al ser el manejo de recursos el principal problema a resolver nos permite proporcionar soluciones a otras situaciones que puedan presentarse como el robo hormiga, la pérdida de tiempo en inventariar y la facilitación de información de tu cartera de clientes y sus productos favoritos al momento.



## Objetivos

### Objetivo General:

Proporcionar al usuario una estructura de almacenamiento de datos correctamente estructurada para la administración de mercancías de su negocio.

### Objetivos específicos:

* Proporcionar al usuario la información necesaria para conocer la situación actual de su negocio, por ejemplo:
* Ventas totales en un periodo de tiempo determinado
* Cantidad en stock de los productos
* Mercancías a punto de agotarse
* Top de productos más vendidos o menos vendidos
* Utilidad neta en un periodo de tiempo determinado
* Clientes potenciales
* Proveedores con mejores precios
* Entre otras.
* Evitar robos por parte del personal o posibles alteraciones en los inventarios.
* Facilitación en el cálculo de utilidades y otros conceptos contables.



## Especificación de Entidades y atributos

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **VENTAS** | | | | | |
| **Descripción** | Almacena los datos del cliente y el monto total de la venta. | | | | |
| **Atributos** | **Descripción** | **Tipo de dato** | |
| **id** | Es la llave primaria. | | Int autoincrementable (PK) | |
| **clientes\_id** | Es la llave foránea que relaciona a la entidad clientes. | | (FK) | |
| **estatus\_compras\_ventas\_id** | Es la llave foránea que relaciona a la entidad estatus\_compras\_ventas. | | (FK) | |
| **categoria\_facturas** | Es la llave foránea que relaciona la entidad categoría\_facturas. | | (FK) | |
| **monto\_total** | Almacena el valor monetario total de la venta. | | Decimal(10,2) | |
| **descuentos\_id** | Es la llave foránea que relaciona la entidad con los descuentos. | | (FK) | |

|  |  |  |
| --- | --- | --- |
| **COMPRAS** | | |
| **Descripción** | Almacena los datos del proveedor y el monto total de la compra. | |
| **Atributos** | **Descripción** | **Tipo de dato** |
| **id** | Es la llave primaria. | Int autoincrementable(PK) |
| **proveedores\_id** | Es la llave foránea que relaciona a la entidad proveedores. | (FK) |
| **estatus\_compras\_ventas\_id** | Es la llave foránea que relaciona a la entidad estatus\_compras\_ventas. | (FK) |
| **monto\_total** | Almacena el valor monetario total de la compra. | Decimal(10,2) |
| **descuentos\_id** | Es la llave foránea que relaciona la entidad descuentos. | (FK) |

|  |  |  |
| --- | --- | --- |
| **CLIENTES** | | |
| **Descripción** | Entidad que almacena los datos personales del cliente y sus datos para la aplicación móvil. | |
| **Atributos** | **Descripción** | **Tipo de dato** |
| **id** | Es la llave primaria. | Int autoincrementable(PK) |
| **nombre** | Guarda el nombre completo del cliente. | String(99) |
| **telefono** | Guarda el teléfono del cliente 10 dígitos | String(10) |
| **calle** | Guarda el nombre de la calle de la dirección del cliente. | String(99) |
| **numero** | Guarda el número del domicilio del cliente. | Int |
| **codigo\_postal** | Guarda el código postal de la dirección del cliente. | Int |
| **email** | Guarda un email. | String(45) |
| **password** | Guarda una contraseña generada por el cliente para acceder a la aplicación de pedidos y se guarda encriptada. | String(99) |

|  |  |  |
| --- | --- | --- |
| **PROVEEDORES** | | |
| **Descripción** | Entidad que almacena la información de contacto del proveedor. | |
| **Atributos** | **Descripción** | **Tipo de dato** |
| **id** | Es la llave primaria. | Int auto\_increment(PK) |
| **nombre** | Guarda el nombre completo del proveedor. | String(99) |
| **telefono** | Guarda el teléfono del proveedor 10 dígitos | String(10) |
| **calle** | Guarda la dirección del proveedor. | String(99) |
| **numero** | Guarda el número del domicilio del proveedor. | Int |
| **codigo\_postal** | Guarda el código postal de la dirección del proveedor. | Int |
| **email** | Guarda un email. | String(45) |

|  |  |  |
| --- | --- | --- |
| **INGRESOS** | | |
| **Descripción** | Entidad que contiene los ingresos provenientes de las ventas realizadas al público | |
| **Atributo** | **Descripción** | **Tipo de dato** |
| **id** | Es la llave primaria. | INT autoincrement(PK) |
| **ingreso\_total** | total, del ingreso | DECIMAL(10, 2) |
| **fecha** | fecha en la que se registró el ingreso | TIMESTAMP(2) |

|  |  |  |
| --- | --- | --- |
| **EGRESOS** | | |
| **Descripción** | Almacena el egreso total asociado al número de compra correspondiente. | |
| **Atributo** | **Descripción** | **Tipo de dato** |
| **id** | Es la llave primaria. | INT autoincrement(PK) |
| **egreso\_total** | total, del egreso | DECIMAL(10, 2) |
| **fecha** | fecha en la que se registró el egreso | TIMESTAMP(2) |

|  |  |  |
| --- | --- | --- |
| **UTILIDADES** | | |
| **Descripción** | Almacena la utilidad bruta diaria restando los ingresos y los egresos. | |
| **Atributo** | **Descripción** | **Tipo de dato** |
| **id** | Es la llave primaria. | INT autoincrement(PK) |
| **utilidad** | total, del egreso | DECIMAL(10, 2) |
| **fecha\_cierre** | Fecha del cierre de caja y cálculo de utilidad bruta | TIMESTAMP(2) |
| **ingresos\_id** | Es la llave foránea que relaciona la entidad ingresos con utilidades | FK |
| **egresos\_id** | Es la llave foránea que relaciona la entidad egresos con utilidades | FK |

**NOTA:** Solo se tendrá un registro de utilidad por día ya que esta entidad está relacionada uno a uno con ingresos y egresos.

|  |  |  |
| --- | --- | --- |
| **ÓRDENES DE VENTA** | | |
| **Descripción** | Relaciona la entidad ventas con productos y almacena la cantidad de cada producto asociado a un número de venta. | |
| **Atributo** | **Descripción** | **Tipo de dato** |
| **id** | identificador de la venta | INT(PK) |
| **productos\_id** | identificador de los productos que fueron vendidos en cierto número de venta | FK |
| **ventas\_id** | identificador del número de venta que cuenta con sus datos | FK |
| **cantidad** | cantidad total de la venta | INT |

|  |  |  |
| --- | --- | --- |
| **ÓRDENES DE COMPRA** | | |
| **Descripción** | Relaciona la entidad compras con productos y almacena la cantidad de cada producto asociado a un número de compra. | |
| **Atributo** | **Descripción** | **Tipo de dato** |
| **id** | identificador de la compra | INT(PK) |
| **productos\_id** | identificador de los productos que fueron comprados en cierto número de compra | FK |
| **ventas\_id** | identificador del número de compra que cuenta con sus datos | FK |
| **cantidad** | cantidad total de la compra | INT |

|  |  |  |
| --- | --- | --- |
| **PRODUCTOS** | | |
| **Descripción** | Entidad que almacena la información de cada producto, podría considerarse como la tabla más importante de la base de datos, almacena tanto costo como precio de cada producto. | |
| **Atributo** | **Descripción** | **Tipo de dato** |
| **id** | número identificador del producto | INT autoincrement(PK) |
| **categorias\_productos\_id** | Es la llave foránea que relaciona la entidad productos con categoria\_productos | FK |
| **nombre\_producto** | nombre del producto | VARCHAR(45) |
| **precio** | precio al público | DECIMAL(10, 2) |
| **costo** | costo que tuvo con proveedor | DECIMAL(10, 2) |
| **fecha\_alta** | fecha en la que se dio de alta el producto | TIMESTAMP(2) |

|  |  |  |
| --- | --- | --- |
| **INVENTARIOS** | | |
| **Descripción** | Relaciona la entidad productos con almacenes y almacena la cantidad en stock de cada producto y su valor monetario. | |
| **Atributo** | **Descripción** | **Tipo de dato** |
| **id** | número identificador del inventario | INT autoincrement(PK) |
| **productos\_id** | Es la llave foránea que relaciona la entidad productos con la de inventarios | FK |
| **almacenes\_id** | Es la llave foránea que relaciona la entidad almacenes con la de inventarios | FK |
| **stock** | Cantidad en unidades inventariadas | INT |
| **valorMonetario** | Valor en dinero del producto inventariado | DECIMAL(10, 2) |
| **fecha\_registro** | fecha en la que se registró el inventario | TIMESTAMP(2) |
| **fecha\_caducidad** | Fecha de caducidad del lote de productos inventariado | TIMESTAMP(2) |

|  |  |  |
| --- | --- | --- |
| **ALMACENES** | | |
| **Descripción** | Guarda un conjunto de almacenes virtuales que sirven para identificar el estado del producto, por ejemplo, el almacén 0001 es el almacén de productos disponibles, mientras que el almacén 0002 es el almacén de productos próximos a caducar. | |
| **Atributo** | **Descripción** | **Tipo de dato** |
| **id** | número identificador del inventario | INT autoincrement(PK) |
| **almacen** | Nombre del almacén | VARCHAR(45) |
| **descripcion** | Características que tienen los productos almacenados aquí. | VARCHAR(255) |

|  |  |  |
| --- | --- | --- |
| **CATEGORIAS PRODUCTOS** | | |
| **Descripción** | Define la clasificación de productos para su mejor manejo e identificación. | |
| **Atributo** | **Descripción** | **Tipo de dato** |
| **id** | identificador de la categoría | INT autoincrement(PK) |
| **categoria\_productos** | nombre de la categoría | VARCHAR(45) |

|  |  |  |
| --- | --- | --- |
| **ESTATUS COMPRA-VENTA** | | |
| **Descripción** | Define el estado actual de la compra o venta, se utiliza para llevar un control de solicitudes finalizadas, canceladas, en proceso, etc. | |
| **Atributo** | **Descripción** | **Tipo de dato** |
| **id** | identificador del estatus | INT autoincrement(PK) |
| **estatus** | estado de la compra/venta | VARCHAR(45) |

|  |  |  |
| --- | --- | --- |
| **DESCUENTOS** | | |
| **Descripción** | Define la categoría de descuentos que aplicamos a los clientes o que los proveedores nos apliquen y el porcentaje aplicado del mismo. | |
| **Atributo** | **Descripción** | **Tipo de dato** |
| **id** | identificador del descuento | INT autoincrement(PK) |
| **descuento** | categoría del descuento | VARCHAR(45) |
| **porcentaje** | porcentaje que representa el descuento del monto total | DECIMAL(10,2) |

|  |  |  |
| --- | --- | --- |
| **CATEGORÍAS FACTURA** | | |
| **Descripción** | Define una serie de categorías que nos permite emitir facturas a los clientes según sea el concepto o motivo de venta. | |
| **Atributo** | **Descripción** | **Tipo de dato** |
| **id** | identificador de la categoría | INT autoincrement(PK) |
| **categoria** | estado de la compra/venta | VARCHAR(45) |

|  |  |  |
| --- | --- | --- |
| **USUARIOS** | | |
| **Descripción** | Son los encargados del uso del sistema final, podrán realizar actividades de registro, modificación, consultas a otras entidades según sea su rol de usuario. | |
| **Atributo** | **Descripción** | **Tipo de dato** |
| **id** | identificador del usuario | INT autoincrement(PK) |
| **nombre** | nombre real del usuario | VARCHAR(45) |
| **nombre\_usuario** | username o alias del usuario se utiliza para entrar al sistema | VARCHAR(45) |
| **email** | correo electronico | VARCHAR(45) |
| **password** | contraseña | VARCHAR(45) |
| **roles\_id** | identificador del rol al que pertenece el usuario | FK |

|  |  |  |
| --- | --- | --- |
| **ROLES** | | |
| **Descripción** | Define la jerarquía de privilegios de la entidad usuarios. | |
| **Atributo** | **Descripción** | **Tipo de dato** |
| **id** | identificador del rol | INT autoincrement(PK) |
| **nombre\_rol** | nombre representativo del rol | VARCHAR(45) |

## Diagrama, Esquemático Descripción generada automáticamenteDiagrama Entidad-Relación

## Modelo Relacional

## 

## Tablas

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ventas | | | | | | | | |
| id  (pk) | clientes\_id(fk) | montoTotal | estatus\_compras\_ventas\_id(fk) | categoria\_facturas\_id(fk) | descuentos\_id(fk) | fecha | categorias\_facturas\_id(fk) | ingresos\_id(fk) |
|  |  |  |  |  |  |  |  |  |

|  |  |  |
| --- | --- | --- |
| descuentos | | |
| id (pk) | descuento | porcentaje |
|  |  |  |

|  |  |
| --- | --- |
| Categoria\_facturas | |
| id(pk) | categoria |
|  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| clientes | | | | | | |
| id  (pk) | nombre\_cliente | telefono | calle | password | numero | codigo\_postal |
|  |  |  |  |  |  |  |

|  |  |  |
| --- | --- | --- |
| ingresos | | |
| id(pk) | ingreso\_total | fecha |
|  |  |  |

|  |  |
| --- | --- |
| estatus\_compras\_ventas | |
| id(pk) | estatus |
|  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| utilidades\_total | | | | |
| id(pk) | utilIdad | fecha\_cierre | ingresos\_id(fk) | egresos\_id(fk) |
|  |  |  |  |  |

|  |  |  |
| --- | --- | --- |
| egresos | | |
| id(pk) | egreso\_total | fecha |
|  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| ordenes\_de\_venta | | | |
| id(pk) | productos\_id(fk) | ventas\_id(fk) | cantidad |
|  |  |  |  |

|  |  |
| --- | --- |
| categoria\_productos | |
| id(pk) | categoria\_productos |
|  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| productos | | | | | |
| id  (pk) | nombre\_producto | precio | costo | categorias\_producto\_id(fk) | fecha\_alta |
|  |  |  |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| ordenes\_de\_compra | | | |
| id(pk) | productos\_id(fk) | compras\_id(fk) | cantidad |
|  |  |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| compras | | | | | | |
| id  (pk) | estatus\_compras\_ventas(fk) | proveedores  (fk) | monto\_total | descuentos\_id  (fk) | fecha | egresos\_id(fk) |
|  |  |  |  |  |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| usuarios | | | | | |
| id  (pk) | nombre | nombre\_usuario | email | password | roles\_id(fk) |
|  |  |  |  |  |  |

|  |  |
| --- | --- |
| roles | |
| id(pk) | nombre\_rol |
|  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| inventarios | | | | | |
| id  (pk) | productos\_id(fk) | almacenes\_id(fk) | stock | valorMonetario | fecha\_registro |
|  |  |  |  |  |  |

|  |  |  |
| --- | --- | --- |
| almacenes | | |
| id(pk) | almacen | descripcion |
|  |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| proveedores | | | | | | |
| id  (pk) | nombre\_proveedor | telefono | calle | numero | codigo\_postal | email |
|  |  |  |  |  |  |  |

## Normalización

Las siguientes 6 tablas (almacenes, categorias\_facturas, categorías\_productos, estatus\_compra\_venta, descuentos y roles).

Cumplen con la primera forma normal ya que tiene un valor simple en los atributos, en descripción de la tabla almacenes como el nombre lo dice es una descripción así que es válido.

Cumplen con la segunda forma normal ya que los atributos son dependientes de la llave.

Cumplen con la tercera forma normal ya que no hay relación de los atributos entre ellos por lo que no habrá que hacer otra tabla.

**Almacenes**



**categorias\_facturas**

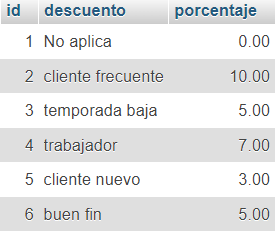
Tabla

Descripción generada automáticamente con confianza media

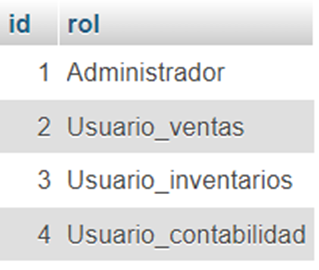
**categorias\_productos**



**estatus\_compras\_ventas descuentos**



**roles**



**Tabla clientes**

Tiene un valor simple en los atributos así que cumple con la primera forma normal, la calle cuenta como un valor.

Cumplen con la segunda forma normal ya que los atributos son dependientes de la llave.

No existe relación de los atributos entre ellos por lo que no habrá que hacer otra tabla, por lo tanto, cumple con la tercera forma.

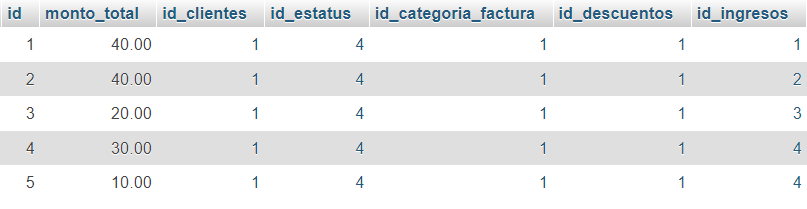
**clientes**



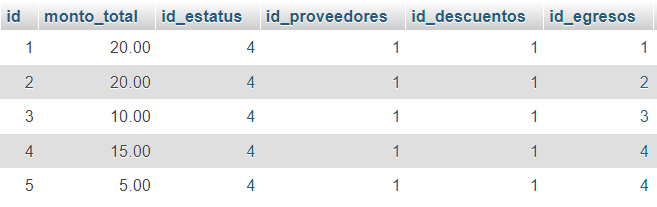
**La tabla Ventas y Compras**

Las tablas ventas y compras son una relación con otras entidades, como las id dentro de las tablas ventas se pueden repetir entonces se agrega un id de ventas para que no se repitan, así se está cumpliendo la primera forma normal, cumple la segunda forma normal ya que las entidades tienen dependencia con la llave primaria, cumple la tercera forma normal porque las entidades no tienen relación con otra llave que no sea la primaria.

**ventas**



**compras**



**La tabla productos**

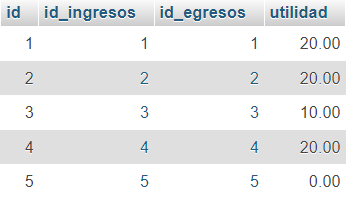
Tiene un único dato dentro de las entidades (1FN), las entidades tienen dependencia con la llave primaria(2FN), las entidades no tienen relación entre ellas o con la llave foránea(3FN).

**productos**



Las tablas Utilidades, ventas, compras, inventarios y son una relación con otras entidades, como las id dentro de las tablas ventas se pueden repetir entonces se agrega un id de ventas para que no se repitan, así se está cumpliendo la primera forma normal, cumple la segunda forma normal ya que las entidades tienen dependencia con la llave primaria, la tercera forma normal se cumple porque las entidades no tienen relación con otra llave que no sea la primaria.

**utilidades**



**ordenesdecompra**  **ordenesdeventa**  



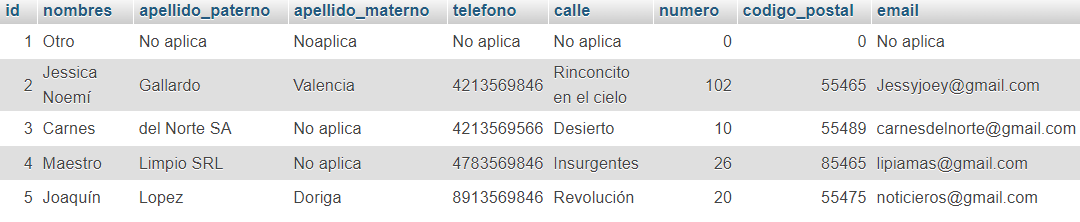
**inventarios**



**La tabla proveedores**

La tabla proveedores tiene un único valor dentro de los atributos, la calle cuenta como un único elemento ya que se trata del mismo dato por lo tanto es válido(1FN), los atributos tienen dependencia con la llave primaria(2FN), los atributos no son dependientes de otros atributos por lo que separar la tabla no es necesario(3FN).

**proveedores**



La tabla users tiene un único valor dentro de los atributos, created\_at y updated\_at el valor que almacenan cuenta como un único elemento(1FN),.

Los atributos tienen dependencia con la llave primaria(2FN).

Los atributos no son dependientes de otros atributos ni de la llave foránea (roles\_id) por lo que separar la tabla no es necesario(3FN).

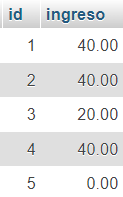
**Users**



**La tabla ingresos**

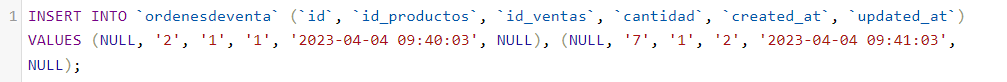
La tabla ingresos tiene un único valor almacenado dentro de los atributos(1FN), las tres entidades tienen dependencia con la llave primaria(2FN), no hay una relación fuerte entre los atributos como para hacer otra tabla(3FN).

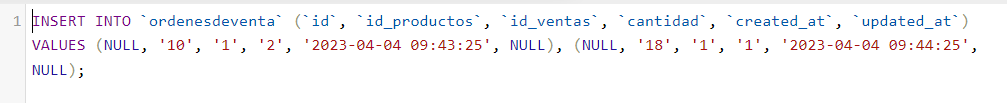
**ingresos**

****

## Enunciados para las sentencias SQL

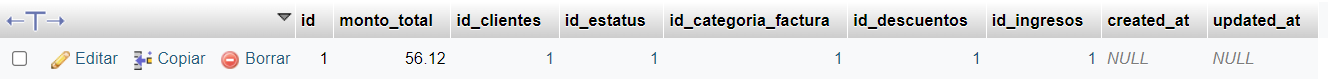
1. **Un nuevo día comienza, llega nuestro primer cliente y con él, nuestra primera venta. Procedemos a crear la nueva venta y agregar cada uno de los productos dentro de la orden asignada a la venta.**

****

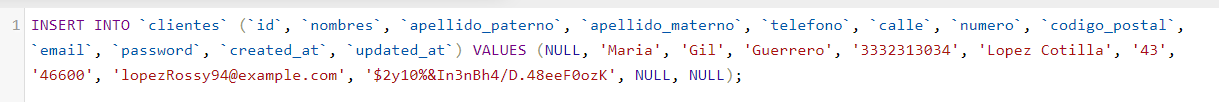
****

**Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente**

****

1. **Un nuevo cliente ha llegado, le ofrecemos el ser cliente registrado y este acepta, procedemos a registrarlo en el sistema**

****

****

1. **El gerente del supermercado necesita una reimpresión del ticket de compra de un cliente ya registrado llamado Juan, el cliente dice que hizo la compra el día 10 de abril del 2023**

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

1. **Ha llegado el fin de mes, el súper requiere de abastecimiento, necesitamos los datos de uno de nuestros proveedores para contactarlo y realizar el pedido**

**Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente**

1. **para una documentación necesaria para un reporte mensual necesitamos la cantidad total generada por las ventas dentro del mes de abril del 2023**

**Interfaz de usuario gráfica, Aplicación

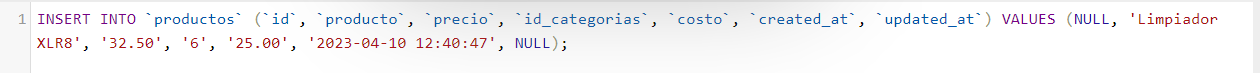
Descripción generada automáticamente**

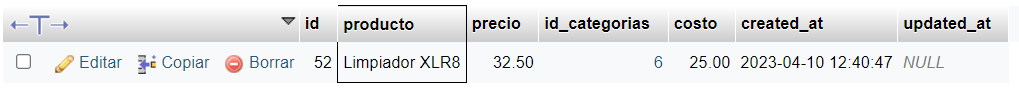
1. **Un cliente quiere abrir una tienda de abarrotes y quiere que nosotros seamos su proveedor. Le entregaremos nuestro catálogo de productos dentro de la categoría de abarrotes.**

**Interfaz de usuario gráfica, Aplicación

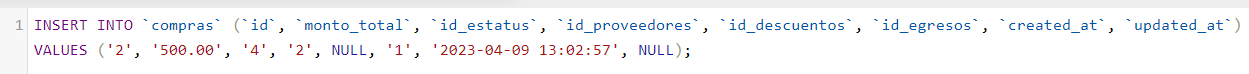
Descripción generada automáticamente**

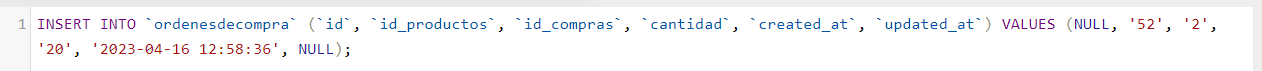
1. **Uno de nuestros proveedores nos recomendó un producto que ha estado en tendencia estas últimas semanas, después de verificarlo nos ha convencido, procederemos a hacer el registro de este**

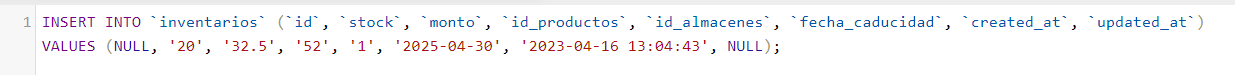
****

****

1. **A la semana siguiente que nuestro proveedor que nos recomendó el producto nos trae el primer cargamento para ver cómo se comportan las ventas respecto a este, haremos la compra y el inventariado de este**

****

****

****

****

1. **Ha pasado temporada vacacional y requerimos saber solamente la cantidad de ingresos que hemos obtenido en las dos semanas referentes a semana santa y pascua**

**Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente**

1. **Un producto está próximo a caducar, por tanto, procedemos a agregarlo a dicho almacén**

Imagen que contiene Texto

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

## Conclusiones

El presente proyecto nos brinda las herramientas necesarias para la correcta administración de entradas y salidas de mercancías, a nivel base de datos, es decir se tiene la estructura relacionada del proyecto mas no se ha concluido la programación del sistema como tal.

Durante el desarrollo de esta nos encontramos con problemas como por ejemplo el almacenamiento masivo de registros en la tabla de ventas, ya que por cada producto se almacenaba un registro, por lo anterior se optó por incluir una tabla llamada orden de ventas donde se almacenarán todos los productos de una venta en específico, evitando así la saturación de información en la tabla ventas y mejor organización en la Base de datos.

Por decisión del equipo se optó por elaborar una base de datos que comprenda un punto de venta y un sistema de inventarios, proporcionando información tanto de compras como de ventas, y teniendo en cuenta estos dos factores se pueden obtener utilidades brutas.

Este proyecto tiene como finalidad desarrollar varios sistemas que se complementen entre sí, proporcionando a los usuarios un control total de sus recursos, ya sean mercancías, personal, maquinaria, equipos, etcétera.

# Referencias

|  |  |
| --- | --- |
| [1] | A. E. y. S. B. Navathe, Sistemas de bases de datos, 5ta Ed Addison Wesley, 2007. |
| [2] | H. F. K. y. A. Silverschatz, Fundamentos de bases de datos, 5ta Ed McGrawHill, 2006. |
| [3] | T. M. C. y. C. E. Begg, Sistemas de bases de datos, 4ta Ed Pearson, 2005. |

# Avances del sistema

El sistema de punto de venta se encuentra en desarrollo actualmente el proyecto estará desarrollado en un fremework de php llamado laravel en su versión 9, esta es una poderosa librería que nos ofrece un gran número de herramientas para el desarrollo web además de sistemas criptográficos que ayudan a la seguridad, se opto por usar este fremework para el desarrollo backend y utilizar un framework de JavaScript conocido como vue para el desarrollo frontend permitiendo la interacción del usuario con el sistema en tiempo real sin necesidad de recargar página después de una petición a la base de datos.

Además de un sistema web, contaremos con una aplicación móvil para el cliente la cual le brindará la facilidad de realizar pedidos desde la comodidad de su hogar, esto pensado para negocios de comida rápida mas que nada.

Para poder comunicar vue con la base de datos se desarrollaron apis con laravel las cuales permiten el acceso a la base de datos por medio de peticiones get y post, añadiendo a esta un Jason web token para mayor seguridad en el manejo de datos.

Actualmente se tienen algunas apis desarrolladas por ejemplo la siguientes:

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

Este api nos arroja todos los productos registrados en la tabla productos, con un formato json como se muestra a continuación:

Texto

Descripción generada automáticamente

Como podemos observar gracias al ORM de php llamado eloquent el cual se utilizó para desarrollar las sentencias sql con código php podemos traer el producto con su relación a la tabla categorías\_productos con su id y su nombre, lo que permite un manejo mas simple a la hora de mostrar datos al usuario.

Este es el modelo que se utilizo para todas las tablas, permite un mejor acomodo de los datos, la comunicación de vue con la base de datos y una doble protección de las peticiones gracias al Json Web Token.

Por parte del frontend solo se ha trabajado en la implementación de una platilla para el sistema y una estructura de bienvenida para el usuario:

Interfaz de usuario gráfica, Aplicación, Sitio web

Descripción generada automáticamente

Se pretende mostrar información importante al usuario al inicio de sesión dependiendo del rol que desempeñe en el negocio, por ejemplo, si es un usuario de ventas les mostraríamos estadísticas del producto que este en oferta, de los productos que falten ventas entre otras, si se tratará el caso de un usuario de inventarios, podríamos mostrarle estadísticas de productos próximos a agotarse, próximos a caducar entre otros.

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Dentro del panel de ventas nos aparecerán las ordenes de venta que han sido encargadas a través de la aplicación o las ventas que aun no tienen estatus de terminada, en el botón rojo podremos dar clic para realizar una nueva venta, la cual permitirá registrar productos por nombre, código de barras, id, etc.