BASE DE DATOS

OSCAR DAVID RUIZ SAEZ

BRAYAN ARCOS BURBANO

INSTITUTO TECNOLOGICO DEL PUTUMAYO

DESARROLLO DE BASE DE DATOS

MOCOA – PUTUMAYO

2024

Contenido

[RESUMEN EJECUTIVO 3](#_Toc176537762)

[INTRODUCCION 4](#_Toc176537763)

[CONTEXTO Y MOTIVACION 4](#_Toc176537764)

[IPORTANCIA DE EL TEMA 4](#_Toc176537765)

[ALCANCE DE EL INFORME 4](#_Toc176537766)

[DISEÑO DE LA BASE DE DATOS 4](#_Toc176537767)

[OPTIMIZACIÓN 4](#_Toc176537768)

[CONSULTAS 5](#_Toc176537769)

[OBJETIVOS 5](#_Toc176537770)

[METODOLOGÍA 5](#_Toc176537771)

[HERRAMIENTAS UTILIZADAS 5](#_Toc176537772)

[PROCEDIMIENTO 5](#_Toc176537773)

[DESARROLLO DEL INFORME 6](#_Toc176537774)

[DESCRIPCIÓN DE LA BASE DE DATOS 6](#_Toc176537775)

[CONSULTAS SQL 8](#_Toc176537776)

[Consulta general 8](#_Toc176537777)

[Consulta usando joins 9](#_Toc176537778)

[Consulta usando subconsultas 10](#_Toc176537779)

[Consultas usando filtros 11](#_Toc176537780)

[DISEÑO DE BASE DE DATOS 12](#_Toc176537781)

[ANALISIS Y DISCUSIONES 12](#_Toc176537782)

[OBJETIVOS 12](#_Toc176537783)

[CONCLUSION 12](#_Toc176537784)

[Recomendaciones 13](#_Toc176537785)

[Referencias 13](#_Toc176537786)

# RESUMEN EJECUTIVO

Contexto y Motivación: Decidí crear una base de datos para una ferretería local porque me intrigaba cómo una herramienta así podría transformar la gestión del negocio. Quería ver cómo una base de datos puede organizar toda la información de productos y clientes de manera estructurada, facilitando encontrar rápidamente lo que se necesita y tomar decisiones más informadas.

Importancia: Una base de datos bien estructurada no solo mejora la eficiencia operativa, sino que también proporciona una ventaja competitiva. Permite adaptarse rápidamente a las necesidades del mercado, ofrecer un mejor servicio al cliente y tomar decisiones basadas en datos concretos. Es una herramienta esencial para cualquier ferretería que quiera crecer y prosperar.

Objetivo: Mi objetivo es transformar la gestión de la ferretería local mediante la implementación de una base de datos eficiente y estructurada. Quiero facilitar el acceso rápido a la información de productos y clientes, mejorando así la toma de decisiones y la calidad del servicio al cliente. Al hacerlo, busco no solo optimizar las operaciones diarias, sino también proporcionar una ventaja competitiva que permita a la ferretería adaptarse a las necesidades del mercado y prosperar en el largo plazo.

Mejoras: La base de datos localHardwareStore ha sido mejorada de manera significativa para optimizar su rendimiento y funcionalidad. Hemos añadido nuevas tablas y refinado la estructura existente para gestionar mejor los datos. Entre las mejoras más destacadas se incluyen la adición de tablas como roll, paymentData y invoiceProducts, así como la simplificación y optimización de otras tablas existentes. Estas modificaciones facilitan la gestión de la información y permiten realizar consultas más eficientes y obtener informes detallados.

Conclusión: Crear esta base de datos ha sido una experiencia increíblemente gratificante. No solo he podido organizar y gestionar la información de productos y clientes de manera más eficiente, sino que también he visto cómo esta herramienta ha mejorado la calidad del servicio al cliente y ha facilitado la toma de decisiones informadas. Al ver cómo la ferretería puede adaptarse rápidamente a las necesidades del mercado y prosperar, me siento orgulloso de haber contribuido a su crecimiento y éxito. Este proyecto no solo ha optimizado las operaciones diarias, sino que también ha proporcionado una ventaja competitiva esencial para el futuro de la ferretería. Saber que mi trabajo ha tenido un impacto positivo en el negocio y en la vida de las personas que dependen de él es realmente satisfactorio.

Recomendaciones:

1. Escucha a tus clientes: Mantén una comunicación abierta para entender sus necesidades y preferencias.
2. Capacita a tu personal: Asegúrate de que estén bien capacitados en el uso de la base de datos y en la atención al cliente.
3. Mantén tu base de datos actualizada: Revisa y actualiza regularmente la información.
4. Analiza los datos: Utiliza la información para identificar tendencias y tomar decisiones informadas.
5. Ofrece promociones personalizadas: Aprovecha los datos para ofrecer promociones y descuentos personalizados.
6. Invierte en tecnología: Considera integrar la base de datos con otras herramientas tecnológicas.
7. Solicita retroalimentación: Pide opiniones para mejorar la base de datos y los procesos relacionados.

# INTRODUCCION

## CONTEXTO Y MOTIVACION

Decidí crear una base de datos para una ferretería local porque me intrigaba cómo una herramienta así podría transformar la gestión del negocio.

Quería ver cómo una base de datos puede organizar toda la información de productos y clientes de manera estructurada. Esto facilita encontrar rápidamente lo que se necesita y tomar decisiones más informadas.

### ImPORTANCIA DE EL TEMA

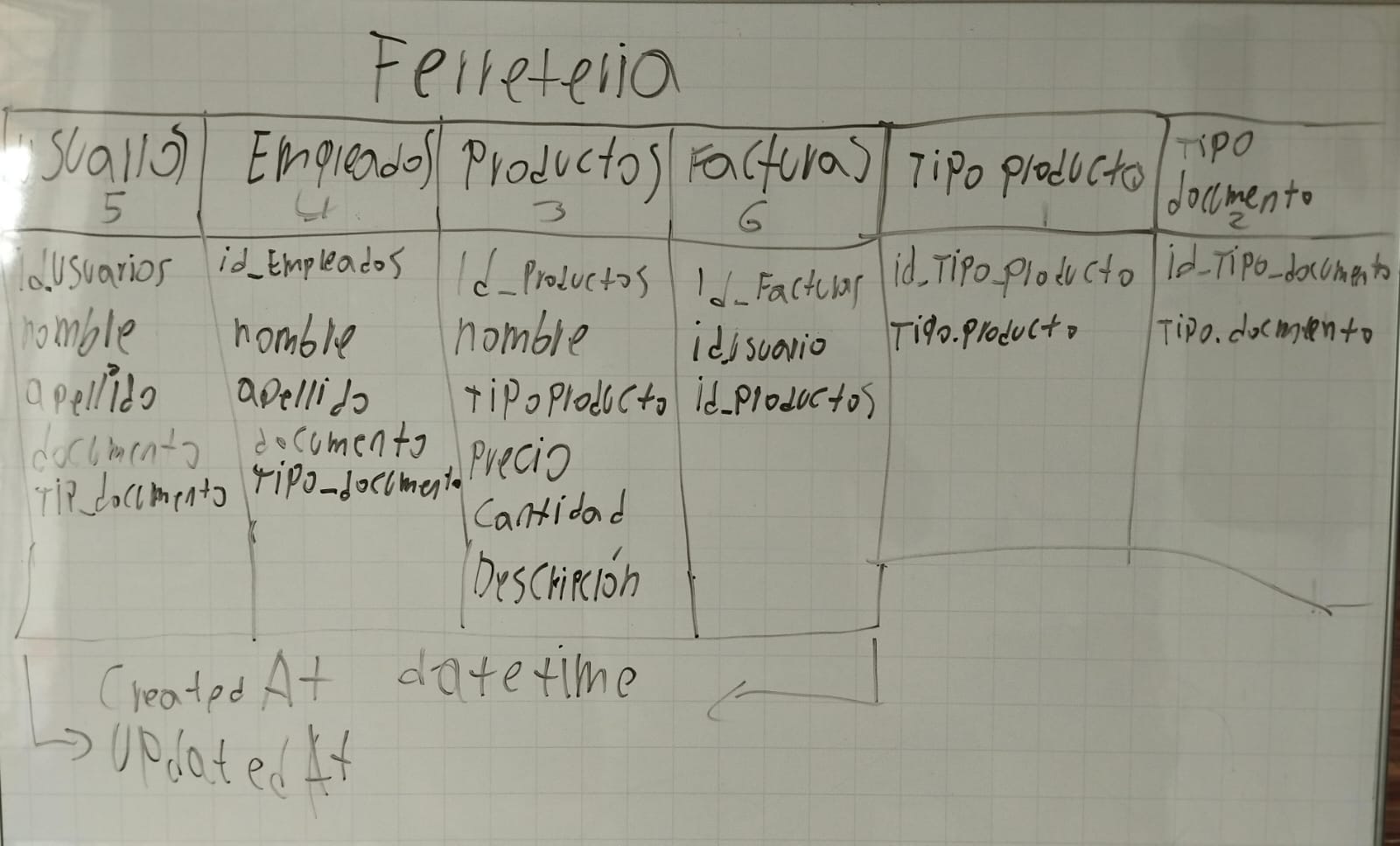
crear una base de datos para una ferretería local no solo mejora la eficiencia operativa, sino que también proporciona una ventaja competitiva. Permite adaptarse rápidamente a las necesidades del mercado, ofrecer un mejor servicio al cliente y tomar decisiones basadas en datos concretos. Es una herramienta esencial para cualquier ferretería que quiera crecer y prosperar.

## ALCANCE DE EL INFORME

Los aspectos que se implementaron en MYSQL WORKBENCH son los siguientes:

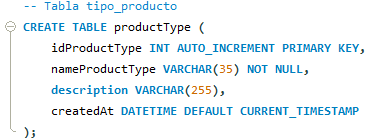
### DISEÑO DE LA BASE DE DATOS

La base de datos de datos se planteo mediante un boceto de tablero



Aquí planteamos como deben quedar las tablas y los números representan el orden en el que se deben crear las tablas en MYSQL.

### OPTIMIZACIÓN

La base de datos localHardwareStore originalmente consistía en tablas básicas que gestionaban productos, usuarios y facturas. La estructura permitía el almacenamiento de datos esenciales, pero presentaba limitaciones en términos de relaciones y detalles específicos.

### CONSULTAS

Se implementaron unas consultas sencillas en la base de datos las cuales se mostrarán después en CONSULTAS SQL del tema DESARROLLO DEL INFORME

## OBJETIVOS

El objetivo de esta base de datos es analizar el flujo de datos en nuestra ferretería, identificar problemas y encontrar patrones para mejorar la eficiencia del negocio.

Principio del formulario

Final del formulario

# METODOLOGÍA

## HERRAMIENTAS UTILIZADAS

Las herramientas que se usaron para crear esta base de datos son las siguientes:

Tablero, IA (Chat GPT, COPILOT), MySQL Workbench, Paint, Excel, tutoriales de YouTube, clase grabada.Principio del formulario

Final del formulario

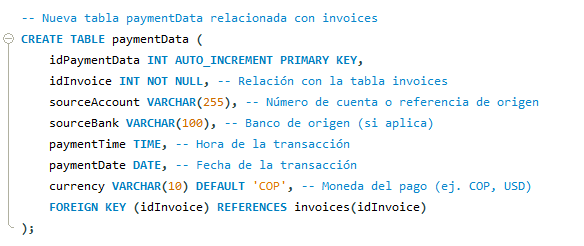
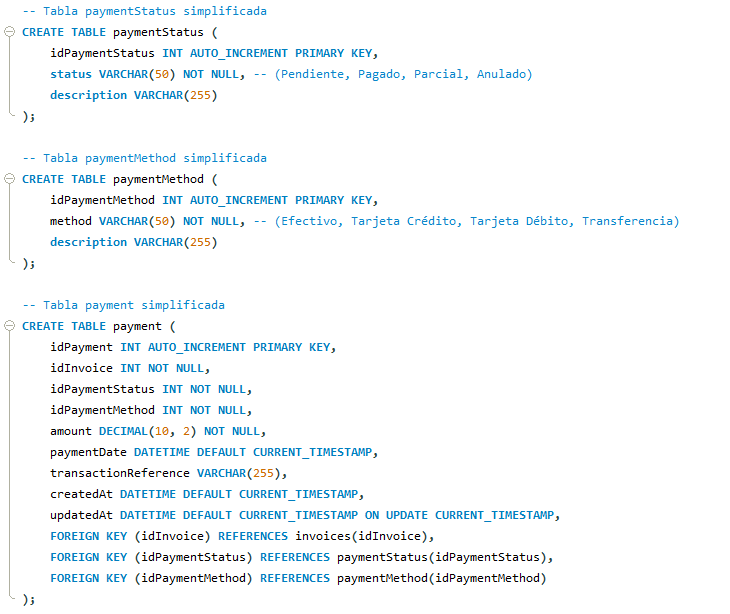
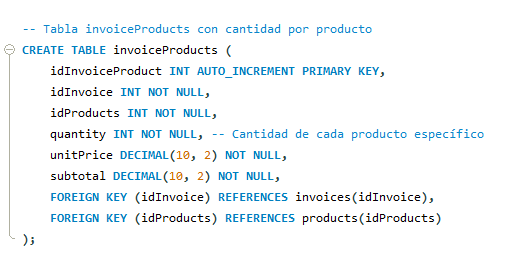
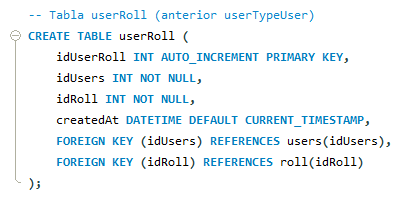
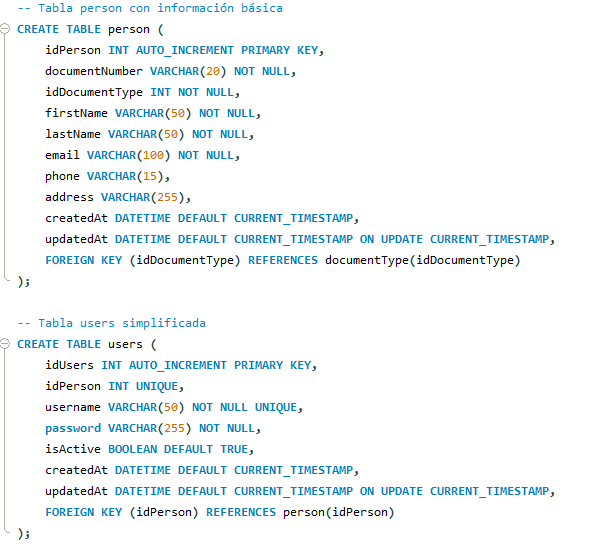
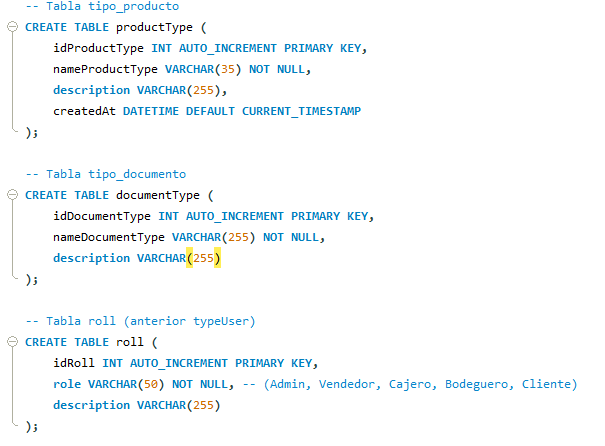
## PROCEDIMIENTO

1. Creamos una base de datos llamada ferretería local.
2. Iniciamos centrándome en una ferretería local pequeña en la cual note que había 4 entidades esenciales la cuales son USUARIOS, EMPLEADOS, PRODUCTOS, FACTURAS.
3. Continuamos con el análisis de cada uno de los posibles atributos necesarios de cada tabla.
4. Definimos que tipo de dato es realmente necesario para cada atributo dentro de la tabla.
5. Realizamos un proceso lógico para definir muy las cardinalidades.

# DESARROLLO DEL INFORME

## DESCRIPCIÓN DE LA BASE DE DATOS

#### TABLAS



#### RELACIONES

Anteriores:

Las relaciones entre las tablas eran básicas, lo que limitaba la capacidad de realizar consultas complejas y mantener la integridad de los datos.

-  
 **1. Tipo\_Producto y productos:**

* + **Tipo de relación: muchos a uno**

1. **Tipo\_documentos y empleados:**
   * **Tipo de relación: Uno a muchos**
2. **Tipo\_documento y usuario:**
   * **Tipo de relación: Uno a muchos**
3. **Usuarios y facturas:**
   * **Tipo de relación: Uno a muchos**
4. **Factura Productos y Productos:**
   * **Tipo de relación: Muchos a muchos**

Actual:

Se ha mejorado significativamente la estructura de las relaciones en la base de datos, permitiendo una gestión más eficiente y segura de la información.

1. **productType y products**: productType tiene una relación uno a muchos con products, donde idProductType en productType es clave primaria y se utiliza como clave foránea en products.
2. **documentType y person**: documentType tiene una relación uno a muchos con person, donde idDocumentType en documentType es clave primaria y se utiliza como clave foránea en person.
3. **roll y userRoll**: roll tiene una relación uno a muchos con userRoll, donde idRoll en roll es clave primaria y se utiliza como clave foránea en userRoll.
4. **person y users:** person tiene una relación uno a uno con users, donde idPerson en person es clave primaria y se utiliza como clave foránea en users.
5. **users y userRoll:** users tiene una relación uno a muchos con userRoll, donde idUsers en users es clave primaria y se utiliza como clave foránea en userRoll.
6. **invoices y users**: users tiene una relación uno a muchos con invoices, donde idUsers en users es clave primaria y se utiliza como clave foránea en invoices.
7. **invoices y invoiceProducts**: invoices tiene una relación uno a muchos con invoiceProducts, donde idInvoice en invoices es clave primaria y se utiliza como clave foránea en invoiceProducts.
8. **products y invoiceProducts**: products tiene una relación uno a muchos con invoiceProducts, donde idProducts en products es clave primaria y se utiliza como clave foránea en invoiceProducts.
9. **invoices y payment**: invoices tiene una relación uno a muchos con payment, donde idInvoice en invoices es clave primaria y se utiliza como clave foránea en payment.
10. **paymentStatus y payment:** paymentStatus tiene una relación uno a muchos con payment, donde idPaymentStatus en paymentStatus es clave primaria y se utiliza como clave foránea en payment.
11. **paymentMethod y payment:** paymentMethod tiene una relación uno a muchos con payment, donde idPaymentMethod en paymentMethod es clave primaria y se utiliza como clave foránea en payment.
12. **invoices y paymentData**: invoices tiene una relación uno a uno con paymentData, donde idInvoice en invoices es clave primaria y se utiliza como clave foránea en paymentData.

#### CLAVES PRINCIPALES

Las claves principales antiguas eran:  
**id\_tipo\_documento | id\_empleados | id\_usuarios | id\_facturas | id productos |id\_tipo\_producto |**

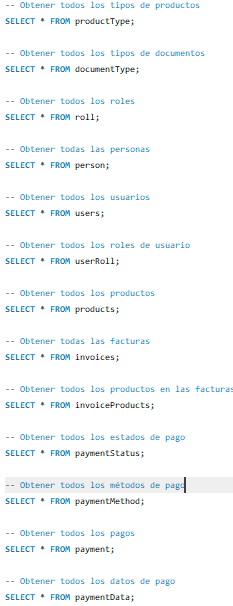
Las claves principales actuales son:

* productType: idProductType
* documentType: idDocumentType
* roll: idRoll person: idPerson
* users: idUsers
* userRoll: idUserRoll
* products: idProducts
* invoices: idInvoice
* invoiceProducts: idInvoiceProduct
* paymentStatus: idPaymentStatus
* paymentMethod: idPaymentMethod
* payment: idPayment
* paymentData: idPaymentData

## CONSULTAS SQL

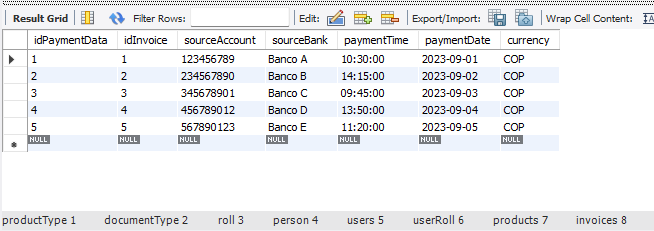
### Consulta general

Esta consulta realiza una obtención en general de todos los datos



##### resultado

no trae todas las tablas y cada una con sus datos solo debemos darle click al nombre de la tabla abajo y listo.

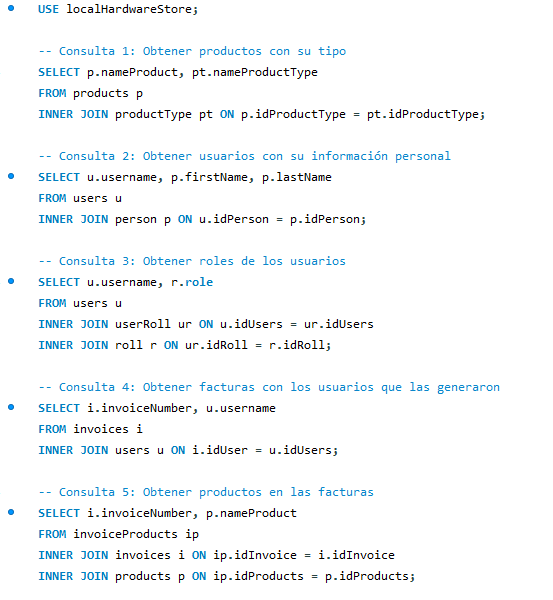


##### Logica

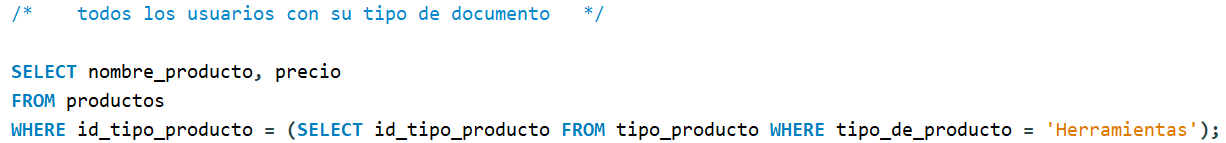
La lógica de esta consulta consiste en que select \* from “nombre de la tabla a traer” nos trae todos los valores de una tabla en específico, pero si hacemos eso con todas las tablas obtenemos todas las tablas.

### Consulta usando joins

Creamos una consulta la cual nos entregue solamente los usuarios con sus nombres, apellidos y los productos que compraron

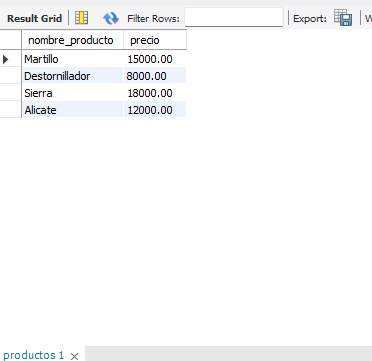


### Consulta usando subconsultas

Creamos una consulta la cual nos entregue los productos que pertenezcan únicamente al tipo de productos llamado “herramientas “

##### Resultado

Esta consulta devuelve una lista de productos que pertenecen al tipo de producto “Herramientas”

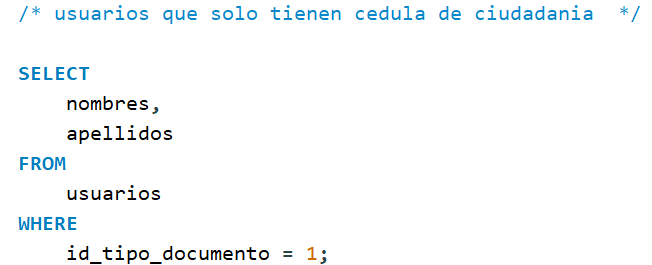


##### Lógica

Esta subconsulta busca el (id\_tipo\_producto) en la tabla (tipo\_producto) donde el (tipo\_de\_producto) es ‘Herramientas’. ya que la lógica de where funciona para limitar datos que no cumplen los requisitos específicos que se solicita

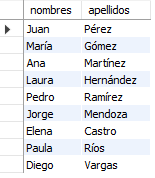
### Consultas usando filtros

Creamos una consulta la cual nos dará los usuarios que tiene “cedula de ciudadanía” como tipo de documento



##### Resultado

Los resultados serán únicamente los usuarios con cedula de ciudadanía

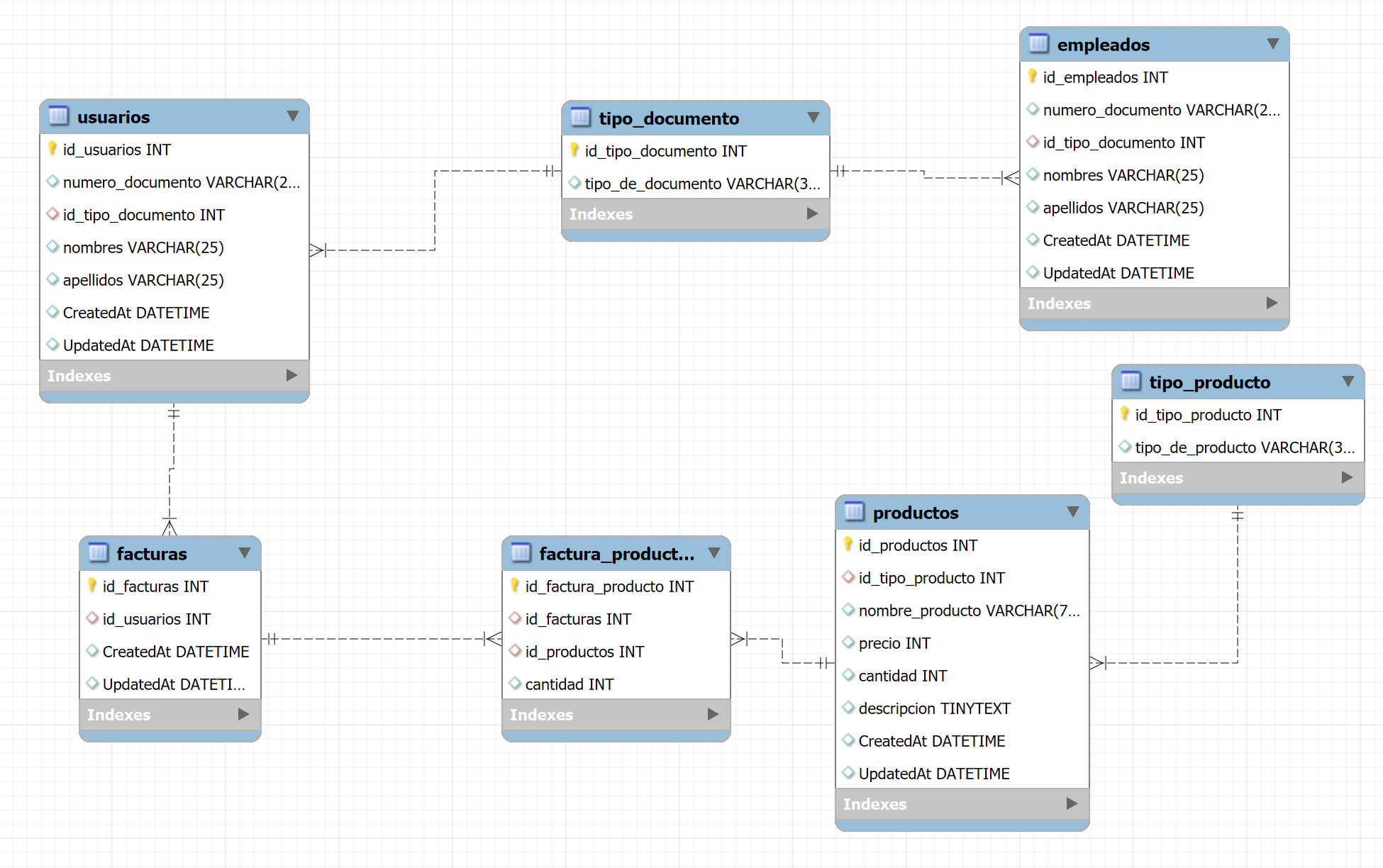


##### Lógica

* Especificamos las columnas que queremos obtener: nombres y apellidos.
* Indicamos que queremos obtener los datos de la tabla usuarios.
* Aplicamos un filtro para obtener solo los usuarios cuyo id\_tipo\_documento es 1, que corresponde a “cedula de ciudadania”.

## DISEÑO DE BASE DE DATOS

Este es el antiguo diseño de la base de datos:



1. Entidades y Atributos:

• Tipo de Producto (Entidad fuerte)

id\_tipo\_producto (Llave primaria)

nombre (Atributo simple)

• Tipo de Documento (Entidad fuerte)

id\_tipo\_documento (Llave primaria)

nombre(Atributo simple)

• Productos (Entidad fuerte)

id\_productos (Llave primaria)

id\_tipo\_producto (Llave foránea a Tipo de Producto)

nombre\_producto (Atributo simple)

precio (Atributo simple)

cantidad (Atributo simple)

descripcion (Atributo simple)

created\_at (Atributo simple, timestamp automático)

updated\_at (Atributo simple, timestamp automático)

• Empleados (Entidad fuerte)

id\_empleados (Llave primaria)

numero\_documento (Atributo simple)

id\_tipo\_documento (Llave foránea a Tipo de Documento)

nombres (Atributo simple)

apellidos (Atributo simple)

created\_at (Atributo simple, timestamp automático)

updated\_at (Atributo simple, timestamp automático)

• Usuarios (Entidad fuerte)

id\_usuarios (Llave primaria)

numero\_documento (Atributo simple)

id\_tipo\_documento (Llave foránea a Tipo de Documento)

nombres (Atributo simple)

apellidos (Atributo simple)

created\_at (Atributo simple, timestamp automático)

updated\_at (Atributo simple, timestamp automático)

• Facturas (Entidad fuerte)

id\_facturas (Llave primaria)

id\_usuarios (Llave foránea a Usuarios)

created\_at (Atributo simple, timestamp automático)

updated\_at (Atributo simple, timestamp automático)

cantidad (Atributo simple)

• Factura Productos (Entidad débil)

id\_factura\_producto (Llave primaria)

id\_facturas (Llave foránea a Facturas)

id\_productos (Llave foránea a Productos)

2. Relaciones:

• Relación entre Productos y Tipo de Producto (1)

id\_tipo\_producto (Llave foránea en Productos)

• Relación entre Empleados y Tipo de Documento (1)

id\_tipo\_documento (Llave foránea en Empleados)

• Relación entre Usuarios y Tipo de Documento (1)

id\_tipo\_documento (Llave foránea en Usuarios)

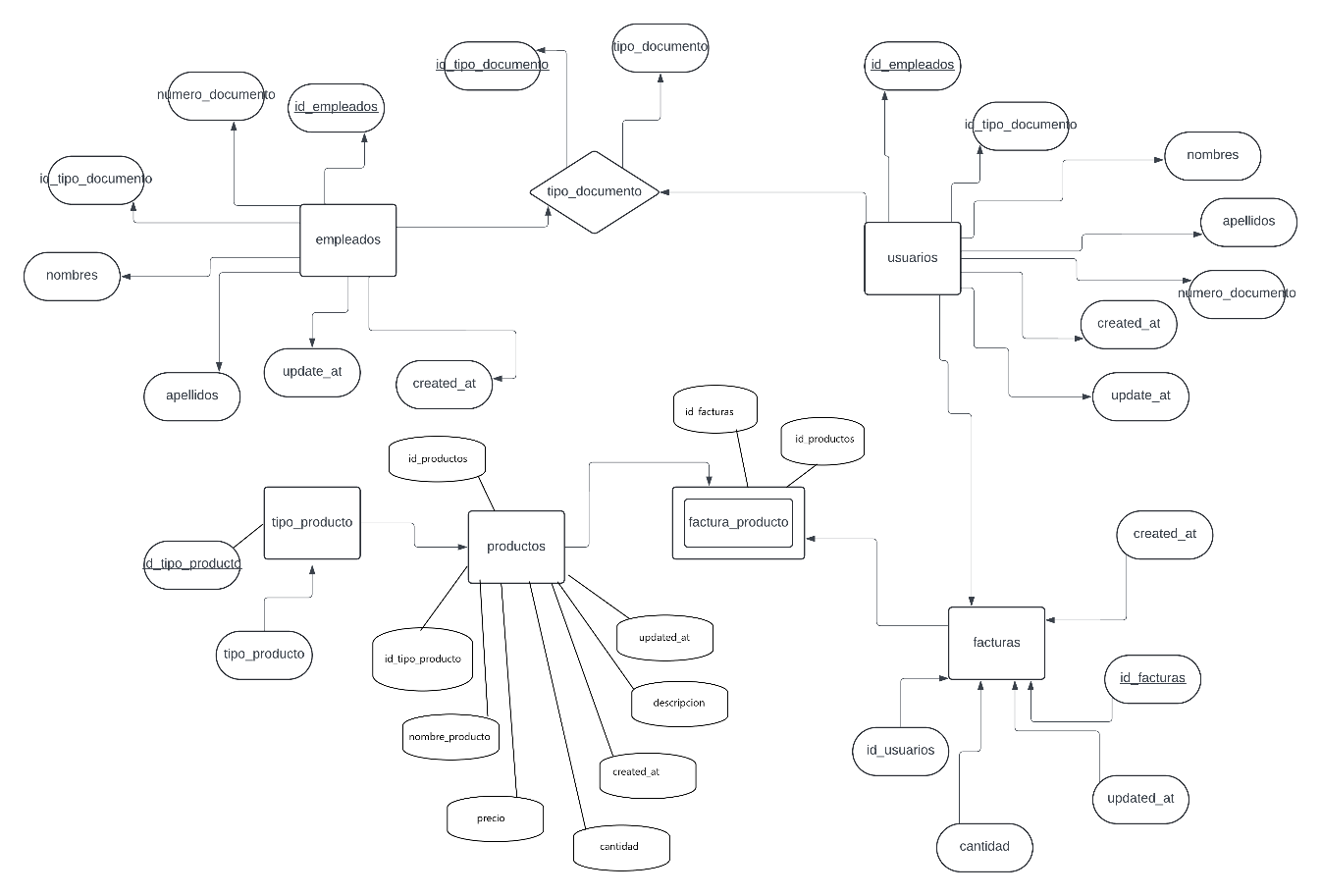
• Relación entre Facturas y Usuarios (1)

id\_usuarios (Llave foránea en Facturas)

• Relación entre Factura Productos y Facturas (N)

id\_facturas (Llave foránea en Factura Productos)

id\_productos (Llave foránea en Factura Productos)



3. Normalización:

Primera Forma Normal (1FN):

Todos los atributos contienen valores atómicos, es decir, no hay atributos multivalorados o repetitivos.

Segunda Forma Normal (2FN):

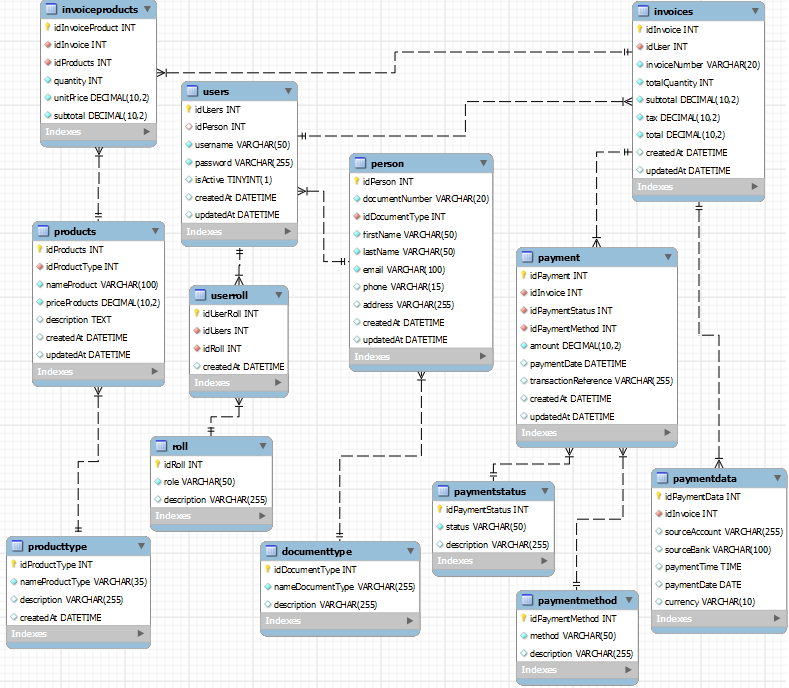
Cada tabla con clave compuesta tiene dependencias completas de todos sus atributos respecto a la clave. En este caso, la tabla factura\_productos está normalizada, ya que sus claves foráneas dependen completamente de la clave primaria compuesta por id\_facturas y id\_productos.

Tercera Forma Normal (3FN):

No existen dependencias transitivas, es decir, todos los atributos no clave dependen únicamente de la clave primaria de cada tabla.

Este modelo de datos está normalizado hasta la Tercera Forma Normal (3FN), lo que asegura que los datos no redundan y se minimizan las anomalías de inserción, actualización y eliminación.

Este es el diseño de la base de datos actual:



Entidades y atributos:

1. productType

* idProductType
* nameProductType
* description
* createdAt

1. documentType

* idDocumentType
* nameDocumentType
* description

1. roll

* idRoll
* role
* description

1. person

* idPerson
* documentNumber
* idDocumentType
* firstName
* lastName
* email
* pone
* address
* createdAt
* updatedAt

1. users

* idUsers
* idPerson
* username
* password
* isActive
* createdAt
* updatedAt

1. userRoll

* idUserRoll
* idUsers
* idRoll
* createdAt

1. products

* idProducts
* idProductType
* nameProduct
* priceProducts
* description
* createdAt
* updatedAt

1. invoices

* idInvoice
* idUser
* invoiceNumber
* totalQuantity
* subtotal
* tax
* total
* createdAt
* updatedAt

1. invoiceProducts

* idInvoiceProduct
* idInvoice
* idProducts
* quantity
* unitPrice
* subtotal

1. paymentStatus

* idPaymentStatus
* status
* description

1. paymentMethod

* idPaymentMethod
* method
* description

1. payment

* idPayment
* idInvoice
* idPaymentStatus
* idPaymentMethod
* amount
* paymentDate
* transactionReference
* createdAt
* updatedAt

1. paymentData

* idPaymentData
* idInvoice
* sourceAccount
* sourceBank
* paymentTime
* paymentDate
* currency

# ANALISIS Y DISCUSIONES

De momento en el análisis ya de ha documentado y no he tenido discusiones claras ya que le he preguntado a la mayoría de personas las cuales no saben al respecto del tema y se enfocan mas en la parte humana que en la parte lógica.

# OBJETIVOS

Mi objetivo es transformar la gestión de la ferretería local mediante la implementación de una base de datos eficiente y estructurada. Quiero facilitar el acceso rápido a la información de productos y clientes, mejorando así la toma de decisiones y la calidad del servicio al cliente. Al hacerlo, busco no solo optimizar las operaciones diarias, sino también proporcionar una ventaja competitiva que permita a la ferretería adaptarse a las necesidades del mercado y prosperar en el largo plazo.

# CONCLUSION

Crear una base de datos para la ferretería local ha sido una experiencia increíblemente gratificante. No solo he podido organizar y gestionar la información de productos y clientes de manera más eficiente, sino que también he visto cómo esta herramienta ha mejorado la calidad del servicio al cliente y ha facilitado la toma de decisiones informadas.

Al ver cómo la ferretería puede adaptarse rápidamente a las necesidades del mercado y prosperar, me siento orgulloso de haber contribuido a su crecimiento y éxito. Este proyecto no solo ha optimizado las operaciones diarias, sino que también ha proporcionado una ventaja competitiva esencial para el futuro de la ferretería. Saber que mi trabajo ha tenido un impacto positivo en el negocio y en la vida de las personas que dependen de él es realmente satisfactorio.

# Recomendaciones

1. Las siguientes recomendaciones me las han regalado personas al azar a las cuales les he comentado lo que estoy haciendo:  
   **Escucha a tus clientes**: Mantén una comunicación abierta con tus clientes para entender sus necesidades y preferencias. Esto te permitirá ajustar tu inventario y servicios para satisfacer mejor sus expectativas.
2. **Capacita a tu personal**: Asegúrate de que tus empleados estén bien capacitados en el uso de la base de datos y en la atención al cliente. Un equipo bien informado y amable puede marcar una gran diferencia en la experiencia del cliente.
3. **Mantén tu base de datos actualizada**: Regularmente revisa y actualiza la información en tu base de datos. Esto incluye agregar nuevos productos, actualizar precios y cantidades, y mantener los datos de los clientes al día.
4. **Analiza los datos**: Utiliza la información almacenada en tu base de datos para identificar tendencias y patrones. Esto te ayudará a tomar decisiones informadas sobre qué productos mantener en stock, cuándo hacer promociones y cómo mejorar tus servicios.
5. **Ofrece promociones personalizadas**: Aprovecha los datos de tus clientes para ofrecer promociones y descuentos personalizados. Esto no solo aumentará las ventas, sino que también fortalecerá la lealtad del cliente.
6. **Invierte en tecnología**: Considera la posibilidad de integrar tu base de datos con otras herramientas tecnológicas, como sistemas de punto de venta (POS) y plataformas de comercio electrónico. Esto puede mejorar la eficiencia y ampliar tu alcance de mercado.
7. **Solicita retroalimentación**: Pide a tus clientes y empleados que te den su opinión sobre cómo mejorar la base de datos y los procesos relacionados. La retroalimentación constructiva puede ayudarte a identificar áreas de mejora que quizás no habías considerado.

# Referencias

De momento no porto referencias claras