
<Company Name>



Ingeniería de Si

<Authors>

Fraider Tarazona Morales

Ferremax.

Fraider Tarazona Morales

Base de Datos

Universidad francisco de paula santander.

Ocaña, Norte de Santander

2025

<Project Name>	Versión: <1.0>
Documento de especificación del diseño e implementación de la BD	Fecha: <dd/mm/yy>
Bases de datos relacionales	Docente: Byron Cuesta Quintero

Control de cambios

Fecha	Versión	Descripción	Autor
16 / 02 / 2025	1.0	Primera versión sobre el diseño de una base de datos para el negocio ferremax.	Fraider Tarazona
20/ 04 / 2025	1.1	Cambios en las variables de los atributos para las tablas	Fraider Tarazona
22/04/2025	1.2	Ajustes en el MER y datos de las tablas	Fraider Tarazona

<Project Name>	Versión: <1.0>
Documento de especificación del diseño e implementación de la BD	Fecha: <dd/mm/yy>
Bases de datos relacionales	Docente: Byron Cuesta Quintero

Tabla de contenido.

Portada

1. Introducción

2. Estrategia y Análisis

2.1 Negocio y misión

2.2 Requisitos de información

2.3 Reglas del negocio

2.4 Supuestos

2.5 Modelo conceptual (Diagrama de ER)

2.6 Modelo lógico (Diagrama de ER)

3. Diseño

3.1 Especificaciones de diseño

3.2 Transformación del modelo lógico a Diagrama de tablas o grafico de instancias de tablas.

4. Creación

4.1 Modelo físico.

4.2 Script DDL y DML

4.3 Modelo C4 (Nivel 1: Diagrama contexto sistema, Nivel 2: Diagrama contenedor)

4.4 Desarrollo web

4.5 Repositorio Github

5. Transición

5.1 Pruebas de aceptación

6. Producción

6.1 Desplegar el modelo físico en la nube

6.2 Colocar en funcionamiento el sistema

7. Conclusión

8. Recomendaciones

9. Glosario

10. Bibliografía

<Project Name>	Versión: <1.0>
Documento de especificación del diseño e implementación de la BD	Fecha: <dd/mm/yy>
Bases de datos relacionales	Docente: Byron Cuesta Quintero

Portada

1. Introducción

La Ferretería de la Rutina ha decidido mejorar su sistema de inventarios, que actualmente se gestiona de manera manual y es propenso a errores. Este sistema manual presenta problemas como fallas en el registro de productos, pérdidas (no son frecuentes pero si sean presentando), por eso han decidido automatizar este sistema. El objetivo es contar con una solución tecnológica que permita gestionar el inventario de forma más eficiente, asegurando que las actualizaciones se realicen en tiempo real y facilitando tanto las compras como las ventas.

La propuesta consiste en desarrollar un sistema automatizado de gestión de inventarios que utilice una base de datos para hacer un seguimiento en tiempo real de los productos disponibles, mejorando el control y reduciendo los errores en las operaciones.

2. Estrategia y Análisis

2.1 Negocio y misión

Productos ofrecidos:

La Ferretería de la Rutina ofrece una amplia lista de productos de ferretería, que incluyen herramientas, materiales de construcción, pinturas, artículos de plomería y electricidad, entre otros..

Misión:

Brindarle a los clientes productos de calidad para el hogar y la construcción, con un excelente servicio al cliente, contribuyendo a la mejora continua de los procesos y el desarrollo del sector.

Los objetivos del negocio son: la optimización de inventarios para evitar la falta de productos o pérdida de productos.

Mejorar la eficiencia en las compras y ventas mediante un sistema automatizado.

Tipo de negocio que desea ser:

La ferretería busca ser una empresa moderna y eficiente, que aproveche la tecnología para mejorar sus operaciones internas y la experiencia del cliente.

Percepción de los servicios: Los servicios de La Ferretería de la Rutina se perciben como confiables y de calidad, con un fuerte enfoque en el servicio al cliente y la disponibilidad de productos adecuados a las necesidades.

<Project Name>	Versión: <1.0>
Documento de especificación del diseño e implementación de la BD	Fecha: <dd/mm/yy>
Bases de datos relacionales	Docente: Byron Cuesta Quintero

2.2 Requisitos de información

ID	Requerimiento
1	Se requiere información actualizada sobre los productos, incluyendo nombre, descripción, cantidad disponible, precios, proveedores, ubicación en el almacén y códigos de barras.
2	El sistema debe permitir la actualización de inventarios en tiempo real tras cada venta o compra, reflejando los cambios inmediatamente.
3	Es necesario registrar el historial de precios de productos, ya que pueden cambiar con el tiempo.

2.3 Reglas del negocio

ID	Regla	Tipo
1	Cada vez que se realice una venta, el inventario debe actualizarse automáticamente.	P
2	El sistema no debe permitir la venta de productos que no estén registrados en el inventario.	E
3	Los productos deben estar etiquetados con códigos de barras, los cuales deben ser escaneados al realizar una venta o compra..	E
4	Cuando un producto su punto de reorden, se debe generar una alerta de reabastecerlo.	P
5	Los precios de los productos pueden ser modificados solo por el personal autorizado, y cualquier cambio debe quedar registrado en el sistema.	E
6	La información del proveedor debe ser actualizada en caso de cambio de proveedor para un artículo.	P

2.4 Supuestos

ID	Supuesto
----	----------

<Project Name>	Versión: <1.0>
Documento de especificación del diseño e implementación de la BD	Fecha: <dd/mmm/yy>
Bases de datos relacionales	Docente: Byron Cuesta Quintero

1	Se asume que el personal de la ferretería estará capacitado para utilizar el sistema automatizado de inventarios.
2	Se espera que todos los productos sea etiquetados con códigos de barras.
3	Los proveedores de productos están dispuestos a enviar datos actualizados sobre sus precios y disponibilidad cuando se les solicite.

2.5 Modelo conceptual (Diagrama de ER)

Modelo conceptual.

Descripción de como se realizo el siguiente modelo conceptual:

Lo primero que hice fue hallar las entidades, para ya luego empezar a ver que relación tienen cada una de ellas, Las entidades encontradas fueron las siguientes:

- **PRODUCTO.**
- **PROVEEDOR.**
- **INVENTARIO.**
- **COMPRA.**
- **CLIENTE.**
- **VENTA.**

Relación entre producto y proveedor:

- El producto tiene un proveedor (1).
- El proveedor puede suministrar muchos productos (M).

Relación entre producto y compra:

- Una compra está relacionada con varios productos (M).
- Un producto puede ser parte de muchas compras (M).

Relación entre producto y inventario:

- Un producto tiene único registro (1).
- El inventario se actualiza con cada compra de un producto (1).

<Project Name>	Versión: <1.0>
Documento de especificación del diseño e implementación de la BD	Fecha: <dd/mm/yy>
Bases de datos relacionales	Docente: Byron Cuesta Quintero

Relación entre producto y venta:

- Un producto puede ser parte de muchas ventas(M)
- Una venta puede tener varios productos. (M)

Relación entre proveedor y compra:

- Un proveedor realiza muchas compras (M).
- Cada compra la hace un único proveedor (1).

Relación entre Compra y Inventario:

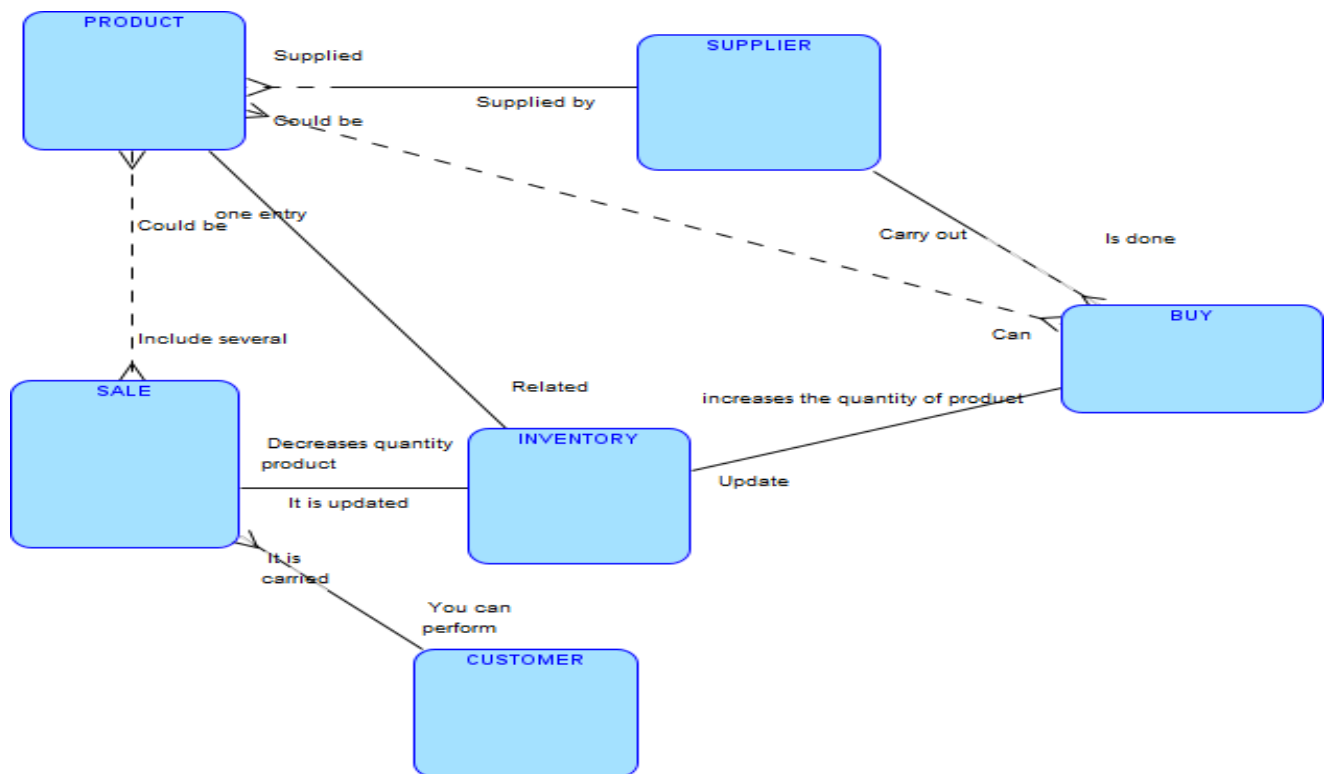
- El inventario se actualiza con cada compra (1).
- Incrementa el producto comprado en el inventario (1).

Relación entre Inventario y Venta:

- Con cada venta, el inventario se actualiza (1).
- El producto se reduce en el inventario (1).

Relación entre Inventario y Venta:

- Un cliente puede realizar muchas ventas.
- Una venta es realizada por un único cliente.



2.6 Modelo lógico (Diagrama de ER)

<Project Name>	Versión: <1.0>
Documento de especificación del diseño e implementación de la BD	Fecha: <dd/mm/yy>
Bases de datos relacionales	Docente: Byron Cuesta Quintero

- Incluye el ERD de la solución propuesta a nivel lógico para el problema de negocio. Tenga en cuenta las convenciones de nomenclatura y dibujo adecuadas para crear el ERD.
- Aplicar formas normales (1FN, 2FN, 3FN). La normalización es un proceso que se utiliza para eliminar la redundancia y así garantizar que la información se guarde en un solo lugar y en el mejor lugar posible.
- Análisis CRUD (Asegurar que todas las funciones CRUD especificadas por el caso del negocio y las reglas del negocio están representadas en el ERD).
- Refinamiento del modelo lógico (socializar a los stakeholders y luego hacer ajustes si es necesario).

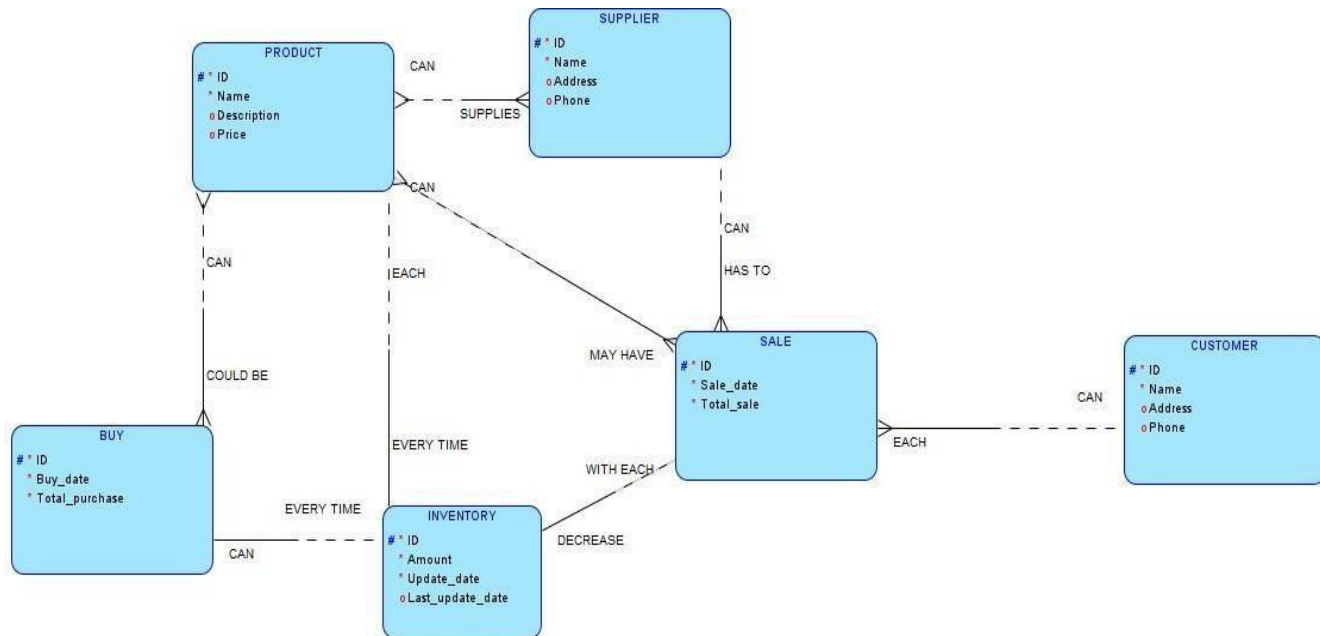


Figura 2. Modelo lógico

3. Diseño

3.1 Especificaciones de diseño

3.2 Transformación del modelo lógico a Diagrama de tablas.

Incluye las tablas generadas a partir de las entidades del modelo lógico. Es útil incluir datos de ejemplo del negocio. El uso de gráficos de instancia de tabla ayudara a detectar errores de asignación o problemas antes de que se creen las tablas de la base de datos.

<Project Name>	Versión: <1.0>
Documento de especificación del diseño e implementación de la BD	Fecha: <dd/mmm/yy>
Bases de datos relacionales	Docente: Byron Cuesta Quintero

Nombre de la tabla	Nombre abreviado de la tabla			
SALE	SL			
TIPO DE CLAVE	OPCIONALIDAD	NOMBRE DE LA COLUMNA	TIPO DE DATO	TAMAÑO
PK	*	ID_SALE	VARCHAR	15
	*	Sale_Date	DATE	
FK	*	ID_CUSTOME R	VARCHAR	15
FK	*	ID_EMPLOYEE	VARCHAR	15

Datos de tabla de ejemplo

Consecutive	Sale_date	Total	Id_employee	DNI
1	2024-10-10	350.750	1	1023456789
2	2024-10-11	425.000.	5	1122334455
3	2025-01-12	180.900	2	2233445566
4	2025-04-13	799.990	2	3344556677
5	2025-02-14	299.500	2	4455667788

Nombre de la tabla	Nombre abreviado de la tabla			
CUSTOMER	CSTR			
TIPO DE CLAVE	OPCIONALIDAD	NOMBRE DE LA COLUMNA	TIPO DE DATO	TAMAÑO
PK	*	ID	VARCHAR	15
	*	NAME	VARCHAR	30
	*	ADDRESS	VARCHAR	50
	*	PHONE	VARCHAR	20
	°	EMAIL	VARCHAR	50

<Project Name>	Versión: <1.0>
Documento de especificación del diseño e implementación de la BD	Fecha: <dd/mm/yy>
Bases de datos relacionales	Docente: Byron Cuesta Quintero

Id_customer	First_name	Last_name	First_surname	Last_surname	dni
1	Anthony	Brown	Evans	Blackburn	4468131643
2	Robin	Hoffman	Nolan	Owens	4374247664
3	Douglas	Chang	Stevens	Simpson	2827925383

Nombre de la tabla	Nombre abreviado de la tabla			
SUPPLIER	SPLR			
TIPO DE CLAVE	OPCIONALIDAD	NOMBRE DE LA COLUMNA	TIPO DE DATO	TAMAÑO
PK	*	ID	VARCHAR	15
	*	NAME	VARCHAR	30
	*	ADDRESS	VARCHAR	50
	*	PHONE	VARCHAR	20

<Project Name>				Versión: <1.0>	
Documento de especificación del diseño e implementación de la BD				Fecha: <dd/mmm/yy>	
Bases de datos relacionales				Docente: Byron Cuesta Quintero	
	*	DNI	VARCHAR	30	

Nombre de la tabla	Nombre abreviado de la tabla			
SALE_BUY	SBY			
TIPO DE CLAVE	OPCIONALIDAD	NOMBRE DE LA COLUMNA	TIPO DE DATO	TAMAÑO
PK	*	CODE	VARCHAR	15
	*	QUANTITY	NUMBER	10,2
	*	PRICE	NUMBER	10,2
	*	SUB_TOTAL	NUMBER	10,2
FK	*	ID_BUY	VARCHAR	15
FK	*	ID_PRODUCT	VARCHAR	15

Nombre de la tabla	Nombre abreviado de la tabla			
BUY				
TIPO DE CLAVE	OPCIONALIDAD	NOMBRE DE LA COLUMNA	TIPO DE DATO	TAMAÑO
PK	*	ID	VARCHAR	15
	*	BUY_DATE	DATE	
	*	TOTAL_PURCHASE	NUMBER	10,2
FK	*	ID_SUPPLIER	VARCHAR	15

Nombre de la tabla	Nombre abreviado de la tabla			
EMPLOYEE	EPLYE			
TIPO DE CLAVE	OPCIONALIDAD	NOMBRE DE LA COLUMNA	TIPO DE DATO	TAMAÑO
PK	*	ID	VARCHAR	15
	*	ID_MANAGER	VARCHAR	15

<Project Name>				Versión: <1.0>
Documento de especificación del diseño e implementación de la BD				Fecha: <dd/mm/yy>
Bases de datos relacionales				Docente: Byron Cuesta Quintero
	*	TYPE	VARCHAR	20
	*	NAME	VARCHAR	20
	*	PHONE	VARCHAR	20
	*	ADDRESS	VARCHAR	30

Datos de tabla de ejemplo

Id employee	First name	Last name	First surname	Last surname	DNI	Telephone	Address	Email
1	Carlos	Rivera	López	Hernández	1023456789	3114567890	Calle 10 #23-45	carlos.rivera@email.com
2	Laura	Gómez	Martínez	Díaz	1122334455	3209876543	Carrera 45 #12-34	laura.gomez@email.com
3	Andrés	Pérez	González	Torres	2233445566	3001234567	Avenida 7 #56-78	andres.perez@email.com
4	Juliana	Ríos	Fernández	Mejía	3344556677	3187654321	Calle 85 #7-32	juliana.rios@email.com
5	Sebastián	Ortiz	Ramírez	Salazar	4455667788	3019988776	Carrera 11 #9-99	sebastian.ortiz@email.com

Nombre de la tabla	Nombre abreviado de la tabla			
SALE_DETAIL	SLDT			
TIPO DE CLAVE	OPCIONALIDAD	NOMBRE DE LA COLUMNA	TIPO DE DATO	TAMAÑO
PK	*	ID	VARCHAR	15
	*	QUANTITY	NUMBER	10,2
	*	PRICE	NUMBER	10,2
	*	SUB_TOTAL	NUMBER	10,2
	*	TOTAL_SALE	NUMBER	10,2
FK	*	ID_PRODUCT	VARCHAR	15
FK	*	ID_SALE	VARCHAR	15

Datos de tabla de ejemplo

CONSECUTIVE	ORDINAL	ID_PRODUCT	QUANTITY	SALE_PRICE	SUBTOTAL
1	01	P001	2	300.000	600.000
1	02	P003	1	50.000	50.000
2	01	P002	4	80.000	320.000
3	01	P004	1	400.000	400.000
3	02	P005	1	500.000	500.000

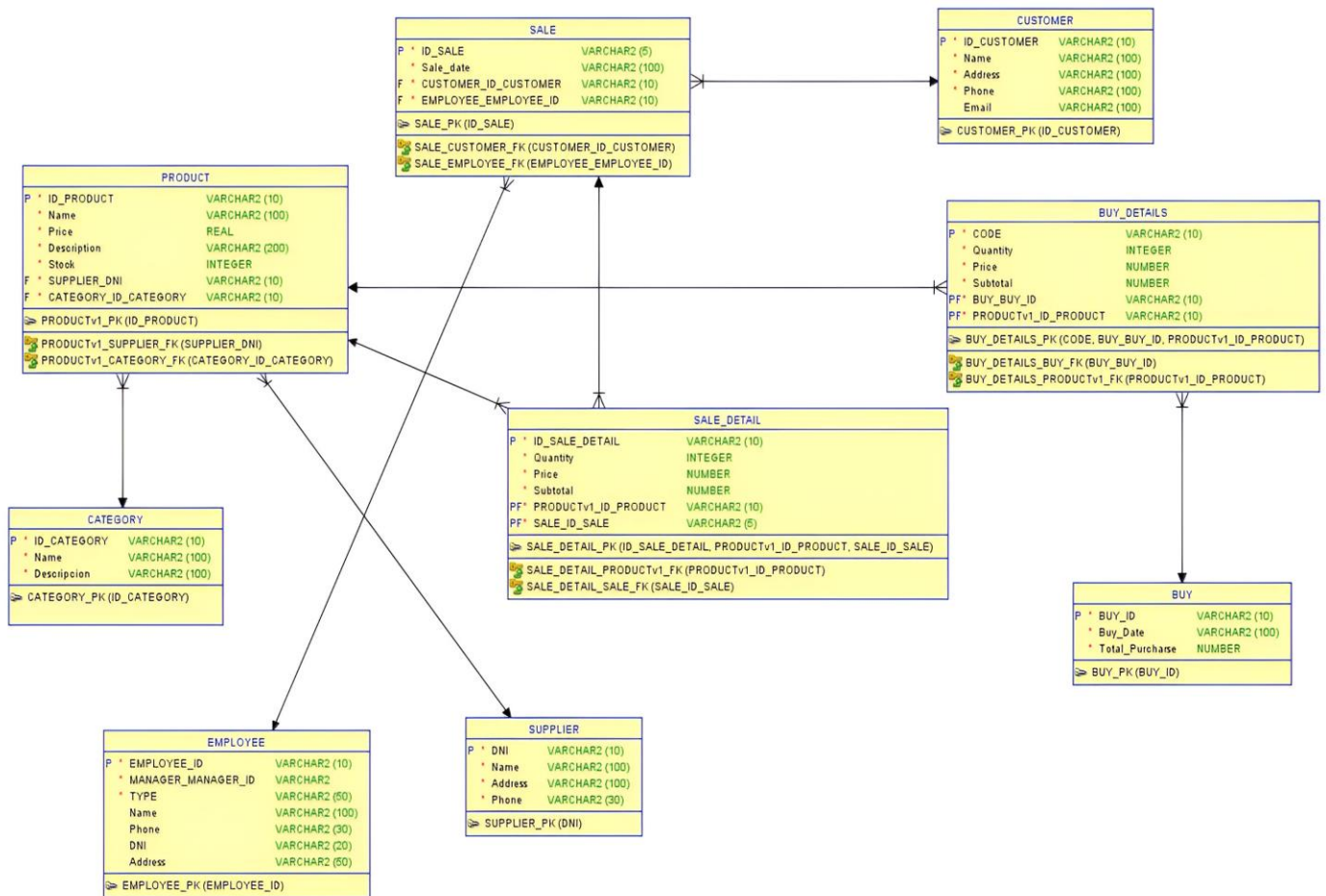
Nombre de la tabla	Nombre abreviado de la tabla			
PRODUCT	PDT			
TIPO DE CLAVE	OPCIONALIDAD	NOMBRE DE LA COLUMNA	TIPO DE DATO	TAMAÑO
PK	*	ID	VARCHAR	15
	*	STOCK	NUMBER	10,2
	*	DESCRIPTION	CLOB	
	*	NAME	VARCHAR	20
	*	PRICE	NUMBER	10,2
FK	*	ID_SUPLIER	VARCHAR	15
FK	*	ID_CATEGORY	VARCHAR	15

Nombre de la tabla	Nombre abreviado de la tabla			
CATEGORY	CTR			
TIPO DE CLAVE	OPCIONALIDAD	NOMBRE DE LA COLUMNA	TIPO DE DATO	TAMAÑO
PK	*	ID	VARCHAR	15
	*	DESCRIPTION	CLOB	
	*	NAME	VARCHAR	50

<Project Name>	Versión: <1.0>
Documento de especificación del diseño e implementación de la BD	Fecha: <dd/mmm/yy>
Bases de datos relacionales	Docente: Byron Cuesta Quintero

Tabla 1. Clientes

MODELO RELACIONAL



<Project Name>	Versión: <1.0>
Documento de especificación del diseño e implementación de la BD	Fecha: <dd/mmm/yy>
Bases de datos relacionales	Docente: Byron Cuesta Quintero

Codigo del script DDL:

-- Generado por Oracle SQL Developer Data Modeler 24.3.1.351.0831 -- en: 2025-04-21 17:45:33 COT -- sitio: Oracle Database 11g -- tipo: Oracle Database 11g

-- predefined type, no DDL - MDSYS.SDO_GEOMETRY

-- predefined type, no DDL - XMLTYPE

CREATE TABLE BUY (ID VARCHAR2 (15) NOT NULL , Buy_date DATE NOT NULL , Total_purchase NUMBER (10,2) NOT NULL , SUPPLIER_ID VARCHAR2 (15 CHAR) NOT NULL) ;

ALTER TABLE BUY ADD CONSTRAINT BUY_PK PRIMARY KEY (ID) ;

CREATE TABLE CATEGORY (ID VARCHAR2 (15 CHAR) NOT NULL , Name VARCHAR2 (50 CHAR) NOT NULL , Description CLOB NOT NULL) ;

ALTER TABLE CATEGORY ADD CONSTRAINT CATEGORY_PK PRIMARY KEY (ID) ;

CREATE TABLE CUSTOMER (ID VARCHAR2 (15 CHAR) NOT NULL , Name VARCHAR2 (30 CHAR) NOT NULL , Address VARCHAR2 (50 CHAR) NOT NULL , Phone VARCHAR2 (20) NOT NULL , Email VARCHAR2 (50 CHAR) NOT NULL) ;

ALTER TABLE CUSTOMER ADD CONSTRAINT CUSTOMER_PK PRIMARY KEY (ID) ;

CREATE TABLE EMPLOYEE (ID VARCHAR2 (15 CHAR) NOT NULL , MANAGER_ID1 VARCHAR2 (15 CHAR) NOT NULL , TYPE VARCHAR2 (50 CHAR) NOT NULL , Name VARCHAR2 (20) NOT NULL , Phone VARCHAR2 (20) NOT NULL , DNI VARCHAR2 (20) NOT NULL , Address VARCHAR2 (50 CHAR) NOT NULL) ;

ALTER TABLE EMPLOYEE ADD CONSTRAINT EMPLOYEE_PK PRIMARY KEY (ID) ;

CREATE TABLE PRODUCT (ID VARCHAR2 (15 CHAR) NOT NULL , Name VARCHAR2 (50 CHAR) NOT NULL , Price NUMBER (10,2) NOT NULL , Description CLOB NOT NULL , Stock NUMBER (10,2) NOT NULL , CATEGORY_ID VARCHAR2 (15 CHAR) NOT NULL , SUPPLIER_ID VARCHAR2 (15 CHAR) NOT NULL) ;

ALTER TABLE PRODUCT ADD CONSTRAINT PRODUCT_PK PRIMARY KEY (ID) ;

CREATE TABLE SALE (ID VARCHAR2 (15) NOT NULL , Sale_date DATE NOT NULL , CUSTOMER_ID VARCHAR2 (15 CHAR) NOT NULL , EMPLOYEE_ID VARCHAR2 (15 CHAR) NOT NULL) ;

ALTER TABLE SALE ADD CONSTRAINT SALE_PK PRIMARY KEY (ID) ;

CREATE TABLE SALE_BUY (CODE VARCHAR2 (15 CHAR) NOT NULL , Quantity NUMBER (10,2) NOT NULL , Price NUMBER (10,2) NOT NULL , Subtotal NUMBER (10,2) NOT NULL , PRODUCT_ID VARCHAR2 (15 CHAR) NOT NULL , BUY_ID VARCHAR2 (15) NOT NULL) ;

<Project Name>	Versión: <1.0>
Documento de especificación del diseño e implementación de la BD	Fecha: <dd/mm/yy>
Bases de datos relacionales	Docente: Byron Cuesta Quintero

ALTER TABLE SALE_BUY ADD CONSTRAINT SALE_BUY_PK PRIMARY KEY (PRODUCT_ID, BUY_ID, CODE) ;

CREATE TABLE SALES_DETAILS (ID VARCHAR2 (15 CHAR) NOT NULL , Quantity NUMBER (10,2) NOT NULL , Price NUMBER (10,2) NOT NULL , Subtotal NUMBER (10,2) NOT NULL , Total_sale NUMBER (10,2) , PRODUCT_ID VARCHAR2 (15 CHAR) NOT NULL , SALE_ID VARCHAR2 (15) NOT NULL) ;

ALTER TABLE SALES_DETAILS ADD CONSTRAINT SALES_DETAILS_PK PRIMARY KEY (PRODUCT_ID, SALE_ID, ID) ;

CREATE TABLE SUPPLIER (ID VARCHAR2 (15 CHAR) NOT NULL , Name VARCHAR2 (30 CHAR) NOT NULL , Address VARCHAR2 (20 CHAR) NOT NULL , Phone VARCHAR2 (20 CHAR) NOT NULL , Dni VARCHAR2 (20 CHAR) NOT NULL) ;

ALTER TABLE SUPPLIER ADD CONSTRAINT SUPPLIER_PK PRIMARY KEY (ID) ;

ALTER TABLE BUY ADD CONSTRAINT BUY_SUPPLIER_FK FOREIGN KEY (SUPPLIER_ID) REFERENCES SUPPLIER (ID) ;

ALTER TABLE PRODUCT ADD CONSTRAINT PRODUCT_CATEGORY_FK FOREIGN KEY (CATEGORY_ID) REFERENCES CATEGORY (ID) ;

ALTER TABLE PRODUCT ADD CONSTRAINT PRODUCT_SUPPLIER_FK FOREIGN KEY (SUPPLIER_ID) REFERENCES SUPPLIER (ID) ;

ALTER TABLE SALE_BUY ADD CONSTRAINT SALE_BUY_BUY_FK FOREIGN KEY (BUY_ID) REFERENCES BUY (ID) ;

ALTER TABLE SALE_BUY ADD CONSTRAINT SALE_BUY_PRODUCT_FK FOREIGN KEY (PRODUCT_ID) REFERENCES PRODUCT (ID) ;

ALTER TABLE SALE ADD CONSTRAINT SALE_CUSTOMER_FK FOREIGN KEY (CUSTOMER_ID) REFERENCES CUSTOMER (ID) ;

ALTER TABLE SALE ADD CONSTRAINT SALE_EMPLOYEE_FK FOREIGN KEY (EMPLOYEE_ID) REFERENCES EMPLOYEE (ID) ;

ALTER TABLE SALES_DETAILS ADD CONSTRAINT SALES_DETAILS_PRODUCT_FK FOREIGN KEY (PRODUCT_ID) REFERENCES PRODUCT (ID) ;

ALTER TABLE SALES_DETAILS ADD CONSTRAINT SALES_DETAILS_SALE_FK FOREIGN KEY (SALE_ID) REFERENCES SALE (ID) ;

-- Informe de Resumen de Oracle SQL Developer Data Modeler:

-- CREATE TABLE 9 -- CREATE INDEX 0 -- ALTER TABLE 18 -- CREATE VIEW 0 -- ALTER VIEW 0 -- CREATE PACKAGE 0 -- CREATE PACKAGE BODY 0 -- CREATE PROCEDURE 0 -- CREATE FUNCTION 0 -- CREATE TRIGGER 0 -- ALTER

<Project Name>	Versión: <1.0>
Documento de especificación del diseño e implementación de la BD	Fecha: <dd/mmm/yy>
Bases de datos relacionales	Docente: Byron Cuesta Quintero

TRIGGER 0 -- CREATE COLLECTION TYPE 0 -- CREATE STRUCTURED TYPE 0 -- CREATE STRUCTURED TYPE BODY 0
 -- CREATE CLUSTER 0 -- CREATE CONTEXT 0 -- CREATE DATABASE 0 -- CREATE DIMENSION 0 -- CREATE
 DIRECTORY 0 -- CREATE DISK GROUP 0 -- CREATE ROLE 0 -- CREATE ROLLBACK SEGMENT 0 -- CREATE
 SEQUENCE 0 -- CREATE MATERIALIZED VIEW 0 -- CREATE MATERIALIZED VIEW LOG 0 -- CREATE SYNONYM 0 --
 CREATE TABLESPACE 0 -- CREATE USER 0 -- -- DROP TABLESPACE 0 -- DROP DATABASE 0 ----- REDACTION
 POLICY 0 -- -- ORDS DROP SCHEMA 0 -- ORDS ENABLE SCHEMA 0 -- ORDS ENABLE OBJECT 0 -- -- ERRORS 0 --
 WARNINGS 0

Codigo de las tablas del proyecto:

```

CREATE TABLE IF NOT EXISTS ferremax.customer (
  id_customer VARCHAR(5) NOT NULL,
  first_name VARCHAR(20),
  last_name VARCHAR(20),
  address VARCHAR(100),
  phone VARCHAR(20),
  email VARCHAR(30),
  CONSTRAINT customer_pk PRIMARY KEY (id_customer),
  CONSTRAINT customer_first_name_nn CHECK (first_name IS NOT NULL),
  CONSTRAINT customer_last_name_nn CHECK (last_name IS NOT NULL),
  CONSTRAINT customer_address_nn CHECK (address IS NOT NULL),
  CONSTRAINT customer_phone_nn CHECK (phone IS NOT NULL) );
  
```

<Project Name>	Versión: <1.0>
Documento de especificación del diseño e implementación de la BD	Fecha: <dd/mm/yy>
Bases de datos relacionales	Docente: Byron Cuesta Quintero

```
CREATE TABLE IF NOT EXISTS ferremax.buy (
buy_id VARCHAR(5) NOT NULL,
buy_date DATE,
total_purchase NUMERIC(10,2),
CONSTRAINT buy_pk PRIMARY KEY (buy_id),
CONSTRAINT buy_date_nn CHECK (buy_date IS NOT NULL),
CONSTRAINT buy_total_nn CHECK (total_purchase IS NOT NULL) );
```

```
CREATE TABLE IF NOT EXISTS ferremax.buy_details ( code character
varying(5) COLLATE pg_catalog."default" NOT NULL, quantity integer,price
numeric(10,2), subtotal numeric(10,2), buy_buy_id character varying(5)
COLLATE pg_catalog."default" NOT NULL, productv1_id_product character
varying(10) COLLATE pg_catalog."default" NOT NULL, CONSTRAINT
buy_details_pk PRIMARY KEY (code, buy_buy_id, productv1_id_product),
CONSTRAINT buy_details_buy_fk FOREIGN KEY (buy_buy_id) REFERENCES
ferremax.buy (buy_id) MATCH SIMPLE ON UPDATE NO ACTION ON DELETE NO
ACTION, CONSTRAINT buy_details_product_fk FOREIGN KEY
(productv1_id_product) REFERENCES ferremax.product (id_product) MATCH
SIMPLE ON UPDATE NO ACTION ON DELETE NO ACTION, CONSTRAINT
buy_details_quantity_nn CHECK (quantity IS NOT NULL), CONSTRAINT
buy_details_price_nn CHECK (price IS NOT NULL), CONSTRAINT
buy_details_subtotal_nn CHECK (subtotal IS NOT NULL), CONSTRAINT
buy_details_buy_nn CHECK (buy_buy_id IS NOT NULL), CONSTRAINT
buy_details_product_nn CHECK (productv1_id_product IS NOT NULL) );
```

<Project Name>	Versión: <1.0>
Documento de especificación del diseño e implementación de la BD	Fecha: <dd/mmm/yy>
Bases de datos relacionales	Docente: Byron Cuesta Quintero

```
CREATE TABLE IF NOT EXISTS ferremax.category ( id_category VARCHAR(5)
NOT NULL, name VARCHAR(50), description VARCHAR(100), CONSTRAINT
category_pk PRIMARY KEY (id_category), CONSTRAINT category_name_nn CHECK
(name IS NOT NULL) );
```

```
CREATE TABLE IF NOT EXISTS ferremax.employee ( employee_id VARCHAR(5)
NOT NULL, type VARCHAR(20), first_name VARCHAR(20), last_name
VARCHAR(20), phone VARCHAR(20), dni VARCHAR(20), address VARCHAR(100),
CONSTRAINT employee_pk PRIMARY KEY (employee_id), CONSTRAINT
employee_type_nn CHECK (type IS NOT NULL), CONSTRAINT
employee_first_name_nn CHECK (first_name IS NOT NULL), CONSTRAINT
employee_last_name_nn CHECK (last_name IS NOT NULL), CONSTRAINT
employee_dni_nn CHECK (dni IS NOT NULL) );
```

```
CREATE TABLE IF NOT EXISTS ferremax.product ( id_product VARCHAR(10) NOT
NULL, name VARCHAR(50), price NUMERIC(10,2), description VARCHAR(100),
stock INTEGER, supplier_dni VARCHAR(10), category_id_category
VARCHAR(5), CONSTRAINT product_pk PRIMARY KEY (id_product), CONSTRAINT
product_category_fk FOREIGN KEY (category_id_category) REFERENCES
ferremax.category (id_category) ON UPDATE NO ACTION ON DELETE NO ACTION,
CONSTRAINT product_supplier_fk FOREIGN KEY (supplier_dni) REFERENCES
ferremax.supplier (id_supplier) ON UPDATE NO ACTION ON DELETE NO ACTION,
CONSTRAINT product_name_nn CHECK (name IS NOT NULL), CONSTRAINT
product_price_nn CHECK (price IS NOT NULL), CONSTRAINT product_stock_nn
CHECK (stock IS NOT NULL), CONSTRAINT product_supplier_nn CHECK
(supplier_dni IS NOT NULL), CONSTRAINT product_category_nn CHECK
(category_id_category IS NOT NULL) );
```

```
CREATE TABLE IF NOT EXISTS ferremax.sale ( id_sale VARCHAR(5) NOT NULL,
sale_date DATE, customer_id_customer VARCHAR(5), employee_employee_id
VARCHAR(5), CONSTRAINT sale_pk PRIMARY KEY (id_sale), CONSTRAINT
```

<Project Name>	Versión: <1.0>
Documento de especificación del diseño e implementación de la BD	Fecha: <dd/mm/yy>
Bases de datos relacionales	Docente: Byron Cuesta Quintero

```

sale_customer_fk FOREIGN KEY (customer_id_customer) REFERENCES
ferremax.customer (id_customer) ON UPDATE NO ACTION ON DELETE NO ACTION,
CONSTRAINT sale_employee_fk FOREIGN KEY (employee_employee_id)
REFERENCES ferremax.employee (employee_id) ON UPDATE NO ACTION ON DELETE
NO ACTION, CONSTRAINT sale_date_nn CHECK (sale_date IS NOT NULL),
CONSTRAINT sale_customer_nn CHECK (customer_id_customer IS NOT NULL),
CONSTRAINT sale_employee_nn CHECK (employee_employee_id IS NOT NULL) );

```

```

CREATE TABLE IF NOT EXISTS ferremax.sale_detail ( id_sale_detail
VARCHAR(5) NOT NULL, quantity INTEGER, price NUMERIC(10,2), subtotal
NUMERIC(10,2), productv1_id_product VARCHAR(10) NOT NULL, sale_id_sale
VARCHAR(5) NOT NULL, CONSTRAINT sale_detail_pk PRIMARY KEY
(id_sale_detail, productv1_id_product, sale_id_sale), CONSTRAINT
sale_detail_product_fk FOREIGN KEY (productv1_id_product) REFERENCES
ferremax.product (id_product) ON UPDATE NO ACTION ON DELETE NO ACTION,
CONSTRAINT sale_detail_sale_fk FOREIGN KEY (sale_id_sale) REFERENCES
ferremax.sale (id_sale) ON UPDATE NO ACTION ON DELETE NO ACTION,
CONSTRAINT sale_detail_quantity_nn CHECK (quantity IS NOT NULL),
CONSTRAINT sale_detail_price_nn CHECK (price IS NOT NULL), CONSTRAINT
sale_detail_subtotal_nn CHECK (subtotal IS NOT NULL), CONSTRAINT
sale_detail_product_nn CHECK (productv1_id_product IS NOT NULL),
CONSTRAINT sale_detail_sale_nn CHECK (sale_id_sale IS NOT NULL) );

```

```

CREATE TABLE IF NOT EXISTS ferremax.supplier ( id_supplier VARCHAR(10)
NOT NULL, name VARCHAR(50), address VARCHAR(100), phone VARCHAR(20),
CONSTRAINT supplier_pk PRIMARY KEY (id_supplier), CONSTRAINT
supplier_name_nn CHECK (name IS NOT NULL) );

```

<Project Name>	Versión: <1.0>
Documento de especificación del diseño e implementación de la BD	Fecha: <dd/mmm/yy>
Bases de datos relacionales	Docente: Byron Cuesta Quintero

```

CREATE OR REPLACE FUNCTION ferremax.fn_auditar_product() RETURNS TRIGGER
AS $$ BEGIN IF TG_OP = 'INSERT' THEN INSERT INTO ferremax.audi_product (
id_product, name, price, description, stock, supplier_dni,
category_id_category, fecha_registro, usuario, accion ) VALUES (
NEW.id_product, NEW.name, NEW.price, NEW.description, NEW.stock,
NEW.supplier_dni, NEW.category_id_category, now(), current_user, 'I' );
RETURN NEW;

ELSIF TG_OP = 'UPDATE' THEN
    INSERT INTO ferremax.audi_product (
        id_product, name, price, description, stock,
        supplier_dni, category_id_category,
        fecha_registro, usuario, accion
    ) VALUES (
        NEW.id_product, NEW.name, NEW.price, NEW.description, NEW.stock,
        NEW.supplier_dni, NEW.category_id_category,
        now(), current_user, 'U'
    );
    RETURN NEW;

ELSIF TG_OP = 'DELETE' THEN
    INSERT INTO ferremax.audi_product (
        id_product, name, price, description, stock,
        supplier_dni, category_id_category,
        fecha_registro, usuario, accion
    ) VALUES (
        OLD.id_product, OLD.name, OLD.price, OLD.description, OLD.stock,
        OLD.supplier_dni, OLD.category_id_category,
        now(), current_user, 'D'
    );
    RETURN OLD;
END IF;

END; $$ LANGUAGE plpgsql;

```

<Project Name>	Versión: <1.0>
Documento de especificación del diseño e implementación de la BD	Fecha: <dd/mm/yy>
Bases de datos relacionales	Docente: Byron Cuesta Quintero

4.3 Repositorio Github

<https://github.com/fraidertara10/BD>

6. conclusion :

La implementación de un sistema automatizado de gestión de inventarios para La Ferretería de la Rutina representa un paso decisivo hacia la modernización y eficiencia operativa del negocio. A través del diseño y desarrollo de una base de datos bien estructurada, se busca solucionar los problemas derivados del manejo manual del inventario, tales como errores en el registro, demoras en las actualizaciones y pérdidas de productos.

La nueva solución permitirá mantener información actualizada en tiempo real sobre productos, precios, proveedores y ubicaciones, así como llevar un control preciso de las ventas, compras y niveles de stock. Además, al establecer reglas de negocio claras y automatizar procesos clave como la actualización del inventario tras cada transacción o la generación de alertas por niveles bajos de productos, se asegura una gestión más ágil y confiable.

Este proyecto no solo responde a las necesidades actuales del negocio, sino que también sienta las bases para un crecimiento sostenido, alineado con la misión de la ferretería de ofrecer productos de calidad con un excelente servicio al cliente. En definitiva, la base de datos desarrollada fortalecerá la capacidad operativa de la ferretería, reducirá errores humanos y mejorará la toma de decisiones, consolidando a La Ferretería de la Rutina como una empresa moderna y competitiva en su sector.

7. Recomendaciones:

- Implementar copias de seguridad automáticas: Configurar respaldos diarios o semanales para garantizar la disponibilidad de los datos en caso de fallos o errores humanos.
- Actualizar la documentación técnica periódicamente: Mantener actualizados el modelo lógico, físico y manual de usuario para facilitar futuras modificaciones o integraciones

<Project Name>	Versión: <1.0>
Documento de especificación del diseño e implementación de la BD	Fecha: <dd/mm/yy>
Bases de datos relacionales	Docente: Byron Cuesta Quintero

· Actualizar la documentación técnica periódicamente: Mantener actualizados el modelo lógico, físico y manual de usuario para facilitar futuras modificaciones o integraciones

Base de Datos Relacional Modelo de almacenamiento que organiza datos en tablas relacionadas por claves.

Entidad Representación de un objeto real (como cliente, producto, proveedor).

Clave Primaria

(PK) Identificador único de cada registro dentro de una tabla.

Clave Foránea (FK) Campo que vincula una tabla con otra para mantener integridad referencial.

Consulta SQL Instrucción utilizada para recuperar o manipular datos en la base.

8. Glosario

Índice Estructura que mejora la velocidad de acceso a datos específicos.

Transacción Conjunto de operaciones que se ejecutan como una unidad atómica.

Backup Copia de seguridad de la base de datos para evitar pérdida de información.

Integridad de Datos Garantía de que los datos son correctos, válidos y coherentes.

Supabase Plataforma de backend como servicio basada en PostgreSQL, utilizada para alojar esta base de datos en la nube.

9. Bibliografía

Silva, D. d. (15 de Julio de 2022). ZENDESK. Obtenido de Base de datos de un sistema de ventas: ¿qué tipo de información contiene?:

<https://www.zendesk.com.mx/blog/base-de-datos-de-un-sistema-de-ventas/>

Silberschatz, A., Korth, H. F., & Sudarshan, S. (2020). Conceptos de sistemas de bases de datos (7.^a ed.). McGraw-Hill. Obtenido de:

<https://www.mheducation.com/highered/product/database-system-concepts-silberschatz-korth/M9780078022159.html>

<Project Name>	Versión: <1.0>
Documento de especificación del diseño e implementación de la BD	Fecha: <dd/mm/yy>
Bases de datos relacionales	Docente: Byron Cuesta Quintero

Supabase. (2023). PostgreSQL en la nube: la base de datos detrás de Supabase. Obtenido de: <https://supabase.com/docs/guides/database>

Elmasri, R., & Navathe, S. B. (2017). Fundamentos de sistemas de bases de datos (7.^a ed.). Pearson Educación. Obtenido de: <https://www.pearson.com/store/p/fundamentals-of-database-systems/P100000648870>

PostgreSQL Global Development Group. (2024). Documentación oficial de PostgreSQL 16. Obtenido de: <https://www.postgresql.org/docs/>

Silva, D. d. (15 de julio de 2022). ZENDESK. Obtenido de: Base de datos de un sistema de ventas: ¿qué tipo de información contiene? <https://www.zendesk.com.mx/blog/base-de-datos-de-un-sistema-de-ventas/>

IBM. (2023). ¿Qué es una base de datos relacional? Obtenido de: <https://www.ibm.com/es-es/topics/relational-databases>

Oracle. (2022). Introducción a las bases de datos relacionales. Obtenido de: https://docs.oracle.com/cd/B19306_01/server.102/b14220/intro.htm

Microsoft. (2023). Fundamentos de bases de datos en SQL Server. Obtenido de: <https://learn.microsoft.com/es-es/sql/relational-databases/databases/databases>

MySQL. (2023). MySQL 8.0 Reference Manual. Obtenido de: <https://dev.mysql.com/doc/refman/8.0/en/>

Fundación UNAM. (2021). ¿Qué es una base de datos y para qué sirve?. Obtenido de: <https://www.fundacionunam.org.mx/unam-al-dia/que-es-una-base-de-datos-y-para-que-sirve/>

Microsoft. (2023). Introducción a bases de datos en SQL Server. Obtenido de: <https://learn.microsoft.com/es-es/sql/relational-databases/databases/databases-sql-server>

<Project Name>	Versión: <1.0>
Documento de especificación del diseño e implementación de la BD	Fecha: <dd/mm/yy>
Bases de datos relacionales	Docente: Byron Cuesta Quintero

<Project Name>	Versión: <1.0>
Documento de especificación del diseño e implementación de la BD	Fecha: <dd/mm/yy>
Bases de datos relacionales	Docente: Byron Cuesta Quintero

<Project Name>	Versión: <1.0>
Documento de especificación del diseño e implementación de la BD	Fecha: <dd/mm/yy>
Bases de datos relacionales	Docente: Byron Cuesta Quintero

<Project Name>	Versión: <1.0>
Documento de especificación del diseño e implementación de la BD	Fecha: <dd/mm/yy>
Bases de datos relacionales	Docente: Byron Cuesta Quintero

<Project Name>	Versión: <1.0>
Documento de especificación del diseño e implementación de la BD	Fecha: <dd/mm/yy>
Bases de datos relacionales	Docente: Byron Cuesta Quintero

Pruebas de Rendimiento

<Project Name>	Versión: <1.0>
Documento de especificación del diseño e implementación de la BD	Fecha: <dd/mm/yy>
Bases de datos relacionales	Docente: Byron Cuesta Quintero