



**Universidad Nacional Autónoma de México**  
**Facultad de Ingeniería**  
**Bases de datos teoría**

# **Proyecto final**

## **de la asignatura de bases de datos**

**Alumnos integrantes:**

Alejandro santos Jiménez  
Jurado Rosas Josue David  
Martínez García José Eduardo  
Montiel Juarez Oscar Ivan  
Ramos Cabrera Paul Manuel

**Profesor@: Ing. Fernando Arreola Franco**

Grupo teoría: 01

Semestre 2025-2

Fecha de entrega: 25 de Junio de 2025

## Introducción

El proyecto que mostraremos se limita a la materia de Bases de Datos con manejo de análisis, diseño, implementación y aplicación en casos de negocio y requerimientos basados en un contexto de mercado laboral real, y tiene como objetivo el diseño e implementación de una solución para la gestión de la información de una mueblería, hablando de manera puntual. En un entorno comercial, la capacidad de manejar, organizar y acceder a los datos es crucial y de suma importancia para la toma de decisiones y la optimización de las operaciones.

Nuestro propósito es montar una base de datos que cumpla con los requerimientos de un cliente: el dueño de la mueblería. Esto implica la estructuración de la información y la creación de un sistema que responda a las necesidades del negocio, abarcando desde la gestión de stock y ventas hasta el registro de empleados y proveedores.

Como propuesta, hemos considerado la implementación de técnicas y estructuras para asegurar la eficiencia y la integridad de los datos. Entre ellas, destacamos la integración de índices, como la propuesta de un índice para facilitar la búsqueda por tipos de empleado, lo que permitirá agilizar consultas y mejorar el rendimiento del sistema en la gestión del personal. Este enfoque busca proporcionar al cliente una herramienta que organice su información e impulse la eficiencia y el crecimiento de su negocio. También se optó por implementar disparadores a movimientos específicos que realiza un cliente, tiene como objetivo desarrollar un método de consultas basado en conocimientos básicos de manejo como el manejo de consultas select por parte del cliente, y realiza procesos para mostrar la salida a requerimientos específicos.

# Plan de Trabajo

El desarrollo del proyecto se llevó a cabo en diversas etapas cuidadosamente planificadas con el objetivo de satisfacer los requerimientos del cliente y garantizar un sistema de base de datos funcional, optimizado, adaptable y digital para migrar la forma de operar de la empresa con la finalidad de tener un panorama y control más amplio de sus operaciones. Dichas etapas se dividieron principalmente en el análisis, diseño, implementación, mismas que fueron parte del trabajo del acondicionamiento de acuerdo con los requerimientos, y la aplicación de un caso de estudio referente a una mueblería el cual se trató de un proceso de vista de los resultados obtenidos tiempo después de que se comenzó a utilizar dicha base de datos.

## Gestión de actividades:

### 1. Análisis de requerimientos

Identificación de entidades, atributos y relaciones a partir del texto base.

Responsables: Martínez García José Eduardo, Montiel Juárez Óscar Iván

### 2. Modelo Entidad-Relación (MER)

Diseño conceptual con cardinalidades, atributos y relaciones entre entidades.

Responsables: Martínez García José Eduardo, Ramos Cabrera Paul Manuel

### 3. Modelo Relacional (MR)

Conversión del MER a tablas con claves primarias y foráneas, tipos de datos y opcionales.

Responsables: Martínez García José Eduardo, Ramos Cabrera Paul Manuel, Alejandro Santos Jiménez

### 4. DDL (Definición de estructura)

Creación de tablas en SQL con restricciones y estructura formal del esquema.

Responsable: Montiel Juárez Óscar Iván

### 5. DML, funciones y triggers

Implementación de reglas de negocio y automatizaciones dentro de la base de datos.

Responsables: Martínez García José Eduardo, Ramos Cabrera Paul Manuel, Montiel Juárez Óscar Iván, Jurado Rosas Josué David, Alejandro Santos Jiménez

## **6. Pruebas e inserciones**

Verificación del funcionamiento con inserciones de prueba y ajustes al diseño.

Responsables: Montiel Juárez Óscar Iván, Jurado Rosas Josué David, Alejandro Santos Jiménez

## **7. Consultas y dashboards**

Extracción y visualización de información útil para el cliente.

Responsable: Ramos Cabrera Paul Manuel

## **8. Evaluación del proceso**

Revisión del trabajo en equipo y análisis del proceso seguido.

Responsable: Martínez García José Eduardo

## **9. Presentación final**

Exposición conjunta del producto al cliente, mostrando su utilidad y beneficios.

Responsables: Martínez García José Eduardo, Ramos Cabrera Paul Manuel, Montiel Juárez Óscar Iván, Jurado Rosas Josué David, Alejandro Santos Jiménez

# Diseño

## 1. Análisis de Requerimientos y Modelado Conceptual

El proyecto se inició con un análisis detallado de los requerimientos y el funcionamiento de la mueblería, utilizando el texto descriptivo proporcionado. Este proceso permitió identificar las entidades, sus atributos y las relaciones fundamentales entre ellas, estableciendo una base sólida para el modelado posterior.

Cuadro 1: Entidades y sus Atributos

Entidad	Atributos
<b>CATEGORÍA</b>	categoria_id (PK), nombre, descripcion
<b>ARTÍCULO</b>	código_barras (PK), nombre, precio_venta, precio_compra, stock, fotografía
<b>PROVEEDOR</b>	rfc_proveedor (PK), razón_social_proveedor, cuenta_pago, teléfonos_proveedor, <i>direccion_proveedor</i> (compuesto: cp_prov, estado_prov, calle_prov, colonia_prov, num_dr_prov)
<b>SUCURSAL</b>	sucursal_id (PK), nombre, año_fundación, teléfono_sucursal, <i>direccion_sucursal</i> (compuesto: calle_sucursal, colonia_sucursal, estado_sucursal, cp_sucursal, num_dr_sucursal)
<b>EMPLEADO</b>	rfc_empleado (PK), nom_pila_empleado, ap_p_empleado, ap_m_empleado, email_empleado, tipo_empleado, curp_empleado, fecha_ingreso, <i>direccion_empleado</i> (compuesto: estado_empleado, calle_empleado, colonia_empleado, cp_empleado, num_dr_empleado)
<b>VENTA</b>	folio_venta (PK), fecha_venta, monto_total, cantidad_total_articulos
<b>FACTURACIÓN</b>	id_factura (PK)
<b>CLIENTE</b>	rfc_cliente (PK), nom_pila, ap_p_cliente, ap_m_cliente, razon_social, email_cliente, telefono_cliente, <i>direccion_cliente</i> (compuesto: calle_cliente, colonia_cliente, estado_cliente, cp_cliente, num_dr_cliente)

Cuadro 2: Relaciones y sus Atributos

Relación	Atributos
ARTÍCULO - VENTA ( <i>presenta</i> )	cantidad_artículo, monto_artículo (implícitos)
PROVEEDOR - ARTÍCULO ( <i>surte</i> )	precio_compra, fecha_inicio_surtido

## 2. Diseño del Modelo Entidad-Relación (MER)

A partir del análisis previo, se diseñó un Modelo Entidad-Relación (MER) exhaustivo. En esta fase, se definieron con precisión las cardinalidades, los tipos de atributos y la naturaleza de las relaciones entre las entidades. Se priorizó una implementación óptima, proyectando la solución más apropiada con miras a futuras fases de diseño y manteniendo un orden y distribución coherentes del modelado conceptual.

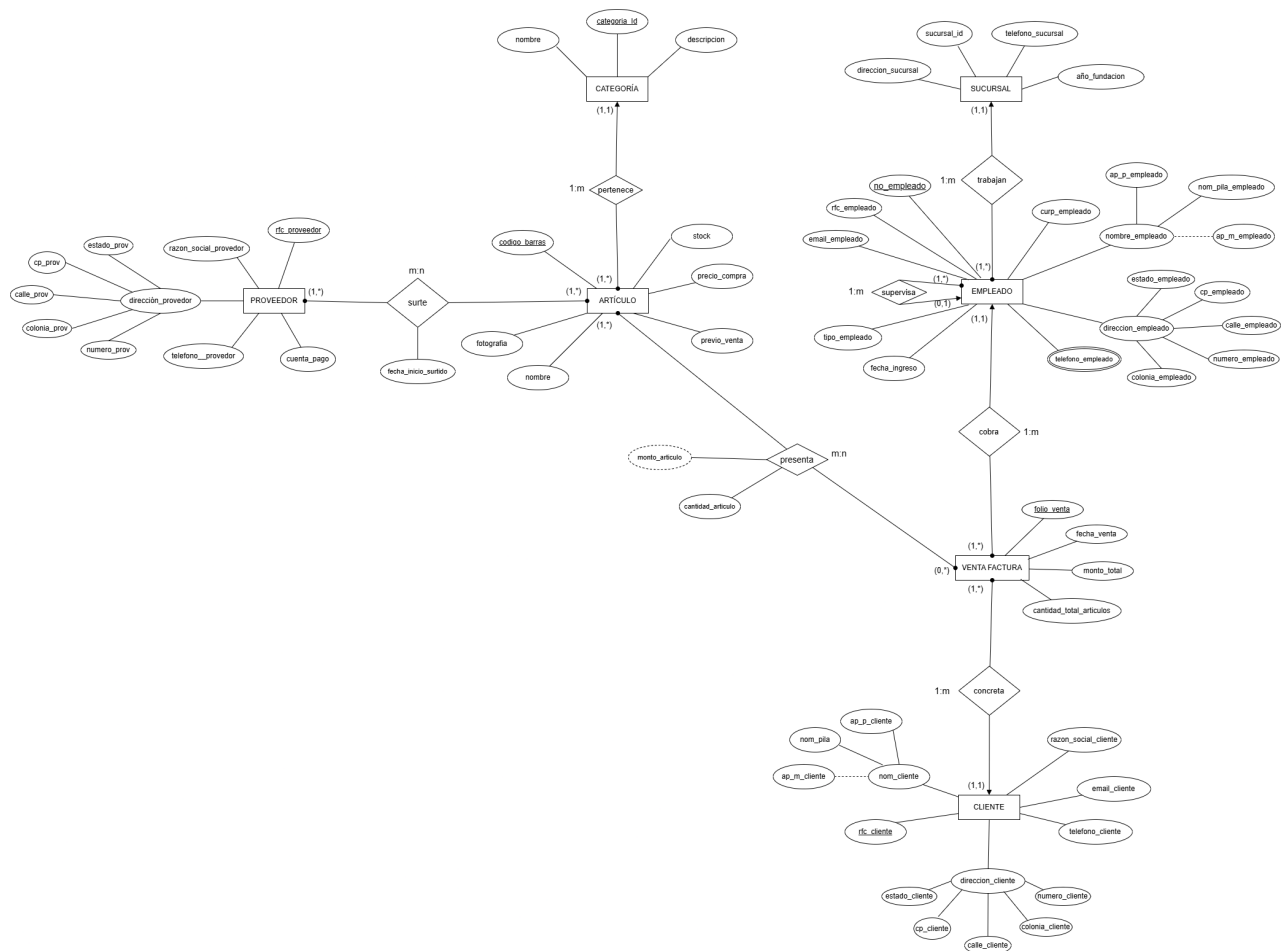


Figura 1: Modelo Entidad-Relación diseñado con la herramienta draw.io

## 3. Transformación al Modelo Relacional (MR)

EL MER fue transformado al Modelo Relacional (MR). En esta etapa, se establecieron las llaves primarias (PKs) y foráneas (FKs) para representar de manera explícita las relaciones entre tablas que en la fase anterior se habían concebido conceptualmente. Las relaciones se visualizaron gráficamente utilizando la notación "pata de cuervo" (crows Foot). Asimismo, se definieron los tipos de datos, los valores opcionales y las restricciones generales necesarias, sin adentrarse aún en los requerimientos particulares.

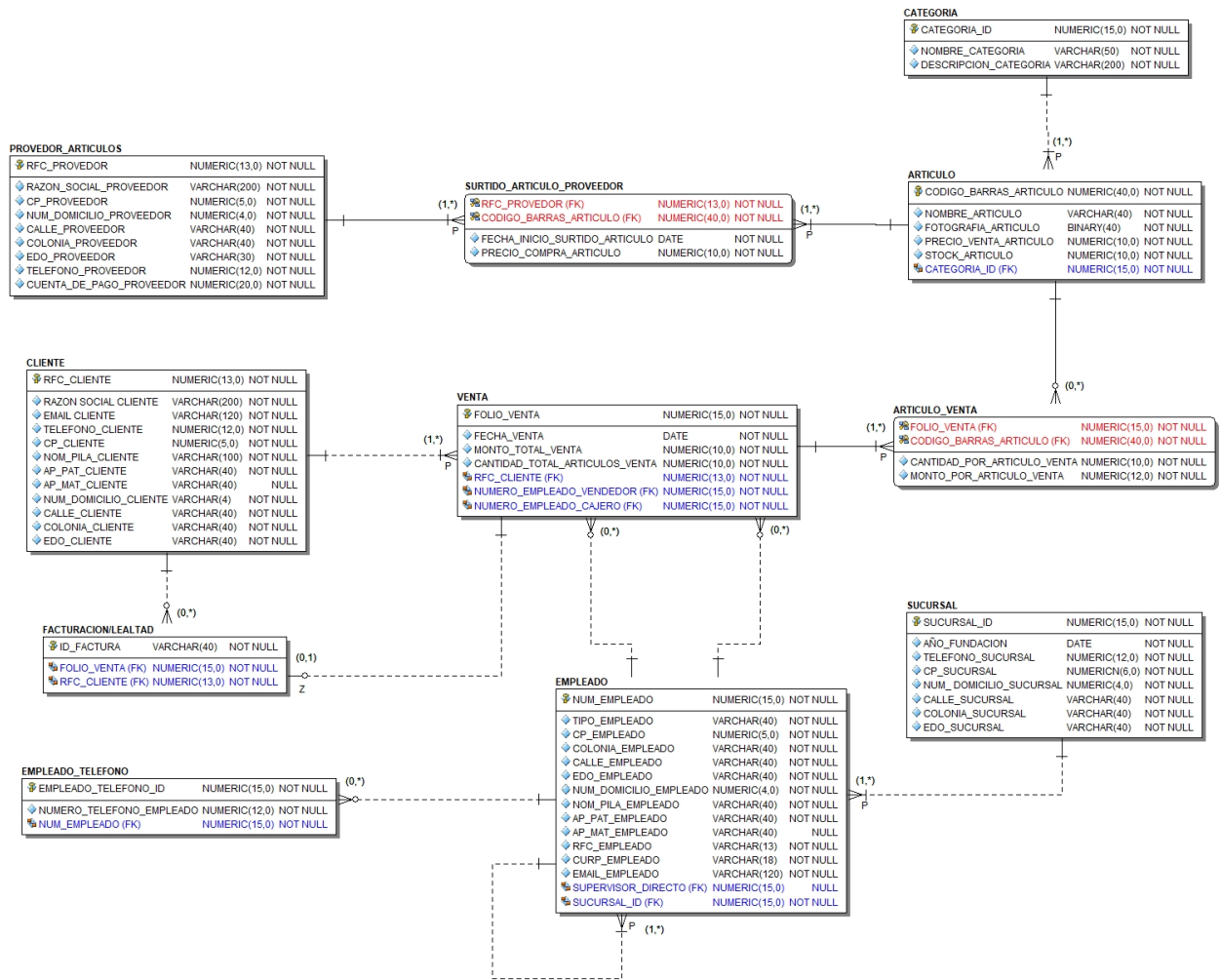


Figura 2: Modelo Relacional diseñado con la herramienta ERStudio

#### 4. Definición de la Estructura con DDL (Data Definition Language)

Se procedió a implementar las tablas-entidades del modelo relacional mediante sentencias DDL (Data Definition Language) en el manejador de base de datos PostgreSQL. Este paso incluyó la creación de las tablas en SQL y la incorporación de restricciones esenciales como claves primarias, foráneas, valores por defecto, índices, la definición de secuencias para ciertos atributos y otras restricciones de integridad.

#### 5. Implementación de Lógica con DML, Funciones y Triggers

Para atender los requerimientos específicos del negocio, se implementó lógica de negocio a través de sentencias DML (Data Manipulation Language), así como funciones y disparadores (triggers). Estas herramientas permitieron automatizar ciertos comportamientos, validar reglas de negocio y garantizar la consistencia de los datos dentro de la base de datos.

## **6. Pruebas e Inserción de Datos**

Se realizaron múltiples inserciones de datos para verificar el funcionamiento correcto de la base de datos y confirmar que las restricciones establecidas se respetaban. Durante esta fase, se lograron identificar y corregir diversos errores que se habían arrastrado desde etapas anteriores, lo que permitió optimizar el diseño y la estructura.

## **7. Desarrollo de Consultas y Dashboards (Aplicación)**

Posteriormente, se diseñaron consultas SQL para extraer información relevante y se construyeron dashboards. El objetivo fue proporcionar al cliente una visualización clara del comportamiento de la mueblería, facilitando así la toma de decisiones estratégicas.

## **8. Evaluación del Proceso y Trabajo en Equipo**

Se llevó a cabo un recuento exhaustivo de las actividades realizadas y un análisis logístico del trabajo colaborativo. Esta evaluación permitió determinar la eficiencia del proceso de desarrollo y el grado de cumplimiento de los objetivos planteados inicialmente.

## **9. Presentación Final al Cliente**

Finalmente, se realizó una presentación conjunta con todos los integrantes del equipo. El propósito fue "vender" la solución al cliente, destacando sus funcionalidades, beneficios y el valor agregado que aporta al negocio, así como rendir cuentas sobre el cumplimiento de todos los requerimientos solicitados por el responsable de la empresa.



## Implementación

Con base en el análisis de requerimientos y el diseño de la base de datos, la siguiente sección detalla la implementación de las funcionalidades clave desarrolladas para el sistema de la mueblería. Aquí se describe cómo cada necesidad operativa y lógica del negocio fue convertida en soluciones prácticas mediante la aplicación de sentencias SQL, funciones y vistas, asegurando la automatización de procesos, la validación de datos y la provisión de información a conveniencia para la posterior aplicación de los datos.

### Requerimientos Específicos

#### Folio de Venta con Formato MBL-001

Para cumplir con este requerimiento, se creó una secuencia llamada **secuencia folio venta**, configurada para iniciar en 1 e incrementarse en 1 con cada uso. Esta secuencia se utiliza dentro de una función llamada **generar folio venta()**, cuyo propósito es obtener el siguiente valor de la secuencia y concatenarlo con el prefijo "MBL-", rellenando con ceros a la izquierda para mantener el formato deseado (por ejemplo, "MBL-001"). Esta función se ejecuta automáticamente antes de insertar un nuevo registro en la tabla venta, de esta manera se logró cumplir con el requerimiento.

#### Artículos No Disponibles o con Stock Menor a 3

Para cumplir con este requerimiento, se creó una vista que permite visualizar la lista de artículos con poco stock o sin stock. Aunque se trata de una consulta, se optó por crear una vista para hacerla más dinámica y evitar escribir toda la consulta cada vez. La vista accede a la tabla original, pero solo muestra los atributos necesarios. Para que en la lista se muestre el mensaje "No disponible", se utilizó la sentencia **CASE**, que funciona de forma similar a un switch case. Esta evalúa el stock del artículo: si el stock es 0, se agrega una nueva columna con el texto "No disponible"; si el stock es menor a 3, se muestra el mensaje "Muy poco stock". La consulta se hace con: **SELECT codigo barras articulo, nombre articulo, stock, estado articulo FROM public.articulos poco stock;**

#### Razón Social del Cliente como Valor por Defecto

Para lograr esto, se creó una función llamada **actualizar razon social**, la cual genera un valor para la columna **razon social cliente** asignando una concatenación del nombre de pila, el apellido paterno y el apellido materno. Se usa la función **COALESCE** para que, en caso de tener un valor NULL, se inserte una cadena vacía, para luego borrar todos los espacios adicionales con la función **TRIM** en caso de que los haya. Esta función se activa antes de insertar o actualizar los datos del cliente.

#### Ventas sin Cliente Registrado

Para este requerimiento, debido a la forma en que teníamos las tablas (el buen análisis de requerimientos), fue suficiente con poner el **rfc cliente** de la tabla venta como NULL.

#### Validación de Sucursal para Vendedor y Cajero en Venta

Se creó una función llamada **validar sucursal venta** en la que, por medio de un **SELECT**, se obtiene el ID del vendedor y el cajero para posteriormente compararlo con un **IF**. Si resultan de diferente sucursal, se cancela la operación con ayuda de un **RAISE EXCEPTION**.

## Generación Automática de Ticket de Venta y Folio de Facturación

Para automatizar la generación de tickets de venta con factura, se implementó la función **generar factura auto()**, la cual se ejecuta automáticamente después de insertar una nueva venta. Esta función verifica si la venta contiene un RFC de cliente registrado. En caso afirmativo, se genera un identificador de factura con el formato FACT-XXXXXX, donde los caracteres aleatorios son obtenidos a partir de una combinación de funciones como **RANDOM()**, **MD5()**, **SUBSTRING()** y **UPPER()** para formar un código único en cada ejecución. Posteriormente, se inserta ese registro en la tabla **facturacion realizada**, vinculando el **id factura**, el **folio venta** correspondiente y el **rfc cliente**. Para que esta acción se dispare de forma automática, se creó el trigger **trg factura automatica**, el cual se activa inmediatamente después de registrar una venta en la tabla correspondiente. Además, se desarrolló una vista llamada **vista ticket facturado**, que integra información de diversas tablas del modelo, incluyendo **facturacion realizada**, **venta**, **cliente**, **empleado**, **articulo venta** y **articulo**. Esta vista permite visualizar un ticket completo que incluye el folio de factura y venta, fecha, información del cliente, nombres completos del vendedor y cajero, así como el detalle de los artículos vendidos con sus respectivas cantidades, precios y subtotales, junto con el monto total de la venta. La consulta de la vista se hace con: **SELECT \* FROM vista ticket facturado WHERE folio venta = 'MBL-XXX'**;

## Creación de Índice

Se creó un índice de tipo **no clustered**, es decir, que no afecta el orden de los datos indexados en la tabla principal. Este índice se implementó en la tabla **empleado**, específicamente en el atributo **tipo empleado**, con el objetivo de que, al realizar una consulta, los datos estén organizados en secciones correspondientes a su tipo. De esta manera, en lugar de comparar registro por registro, el sistema realiza una especie de escaneo y encuentra el registro solicitado de forma inmediata. La consulta para llegar al registro, tomando como ejemplo al usuario “Cantinflas Rodríguez”, se realiza de la siguiente manera: **SELECT \* FROM empleado WHERE tipo empleado = 'Cantinflas Rodríguez'**; Esta sentencia arrojará el registro que coincida con la condición de búsqueda establecida en la cláusula WHERE.

## Obtener Jerarquía Organizacional de un Empleado

La función **jerarquia organizacional** es una herramienta para visualizar las relaciones jerárquicas de empleados en una mueblería. Recibe el nombre y apellido paterno de un empleado para identificar y mostrar su cadena de supervisión ascendente y sus reportes directos. La función primero localiza el **num empleado** del empleado. Si no se encuentra, lanza una excepción. Luego, la función construye la jerarquía: por un lado, asciende en la cadena de mando, asignando niveles negativos a los supervisores (por ejemplo, -1 para el inmediato); por otro, desciende, identificando a los reportes directos con niveles positivos (por ejemplo, +1). Finalmente, la función combina estos dos conjuntos de resultados, consolidando la información de todos los empleados relevantes en una única tabla. Se aplican **CASTs** explícitos a **VARCHAR** en las columnas de texto para asegurar la compatibilidad con el tipo de retorno definido. La tabla resultante se ordena por **nivel jerarquia**, ofreciendo una visión clara de la posición del empleado y su ambiente organizacional a nivel jerárquico. Forma de consulta de la función: **SELECT \* FROM jerarquia organizacional('NombrePila', 'ApellidoPaterno');**

### Actualización Automática de Totales y Validación de Stock al Agregar Artículo a Venta

Para actualizar los totales, así como el stock de cierto artículo, fue necesario modificar los atributos **cantidad por articulo venta** (que corresponde a la cantidad de cierto artículo presente en la venta), **stock articulo** (cantidad de artículos existentes en el establecimiento), **cantidad total articulos venta** (que corresponde a la cantidad de artículos totales en la venta) y, por último, el **monto total venta** (dinero total a pagar por todos los artículos de la venta). La función fue definida con el nombre **actualizar venta y stock**, retorna un trigger que se dispara cada vez que se intenta insertar un registro en la tabla **articulo venta**. En términos simples, hace lo siguiente:

- Comprueba si hay stock suficiente para el artículo que se quiere vender.
- Si no hay stock, lanza un error y detiene la operación.
- Si sí hay stock, revisa si ese mismo artículo ya está en la venta actual, a través de una comparación entre los atributos presentes y nuevos en **folio venta y codigo barras articulo**:
  - Si ya existe, simplemente se actualiza la cantidad del artículo ya existente, sumando la nueva cantidad al registro presente.
  - Si no existe, deja que el registro nuevo se inserte normalmente.
- En ambos casos, resta del stock la cantidad de artículos almacenados en la venta y actualiza los totales de la venta:
  - La cantidad total de artículos (dada la suma de los artículos totales en “venta” más la cantidad de artículos agregados a la venta).
  - El monto total de la venta (obtenido de la suma del monto actual de la venta más la cantidad de artículos nuevos agregados a la venta por el precio del artículo agregado).
- Finalmente, devuelve NEW (o NULL) para que la inserción en **articulo venta** continúe.

# Aplicación

## Dashboard 1

Cantidad de ingresos totales mensuales por sucursal en un año determinado.

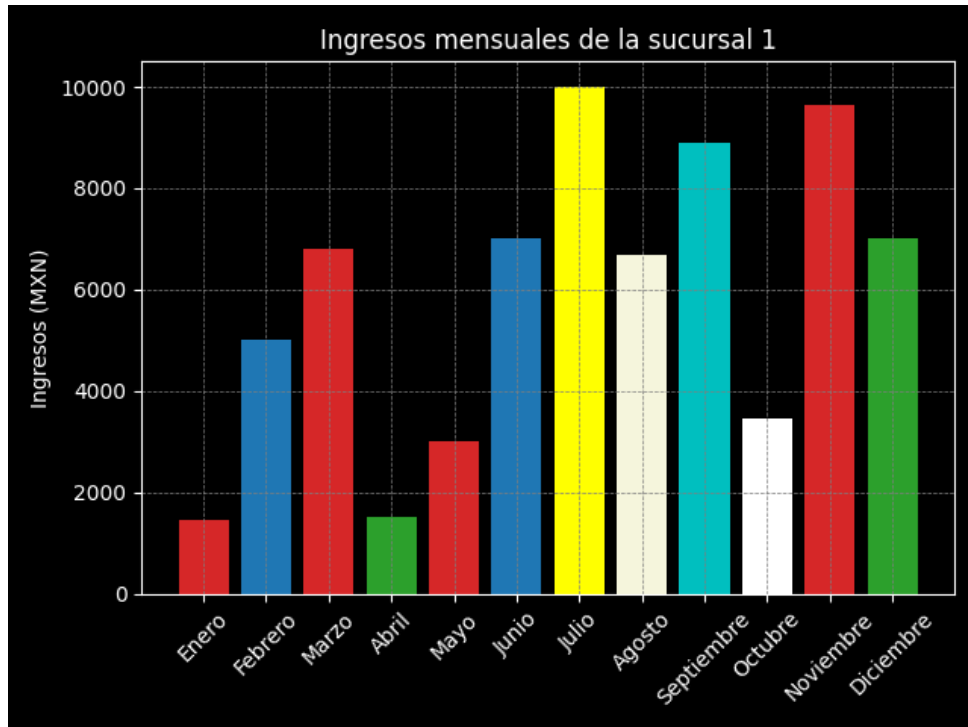


Figura 3: Ingresos mensuales de la sucursal 1

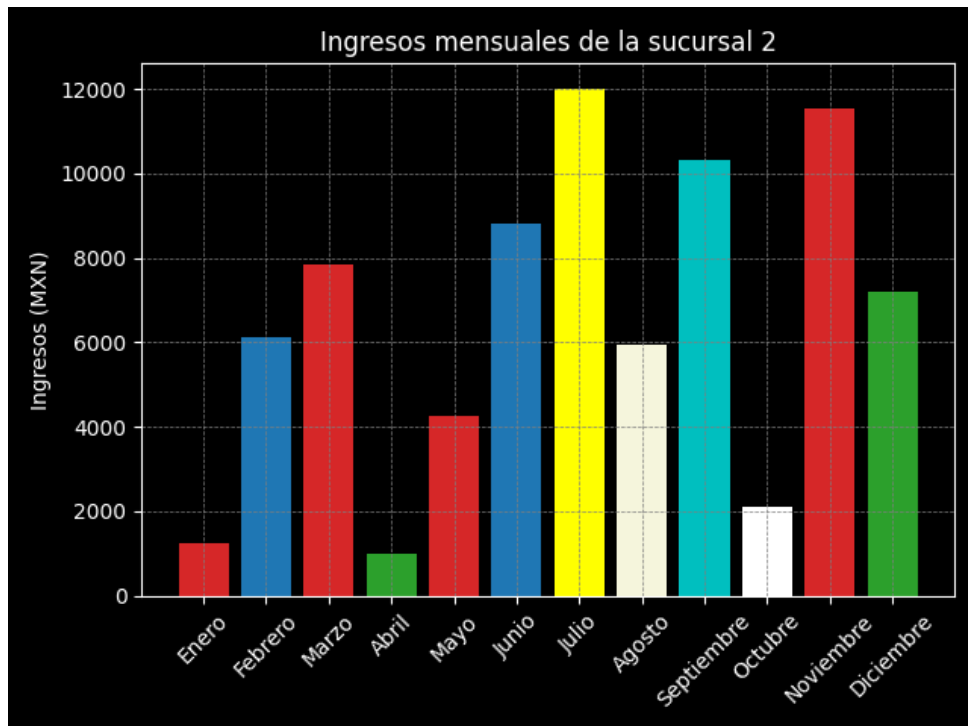


Figura 4: Ingresos mensuales de la sucursal 2

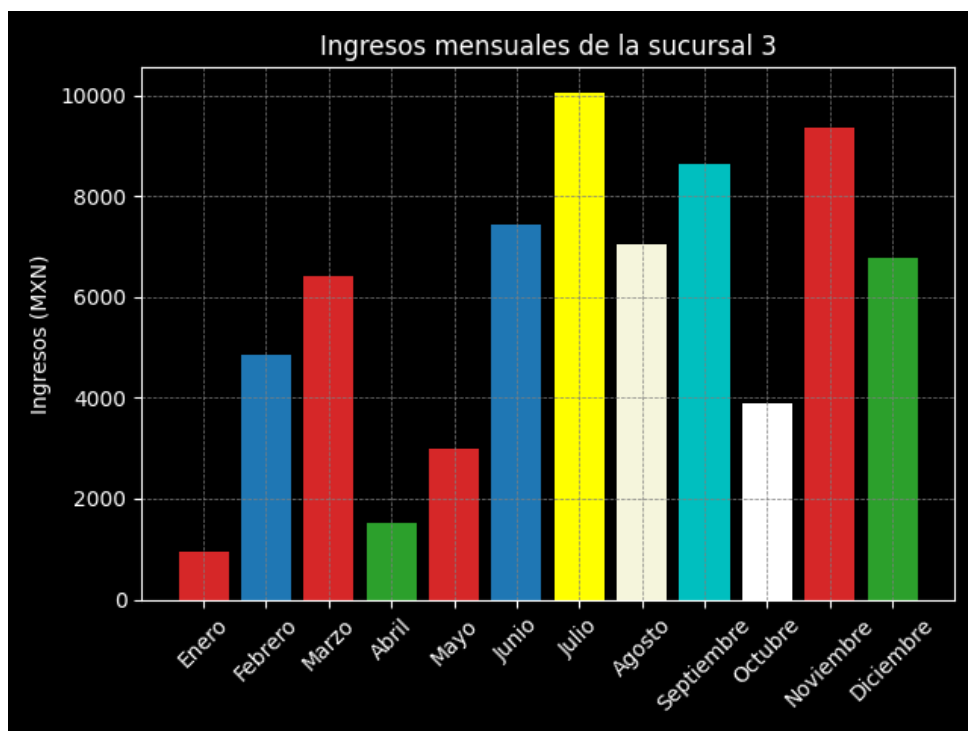


Figura 5: Ingresos mensuales de la sucursal 1

## Dashboard 2

Obtiene los 5 artículos más vendidos de cierto periodo de tiempo.

Top 5 artículos vs. ventas totales

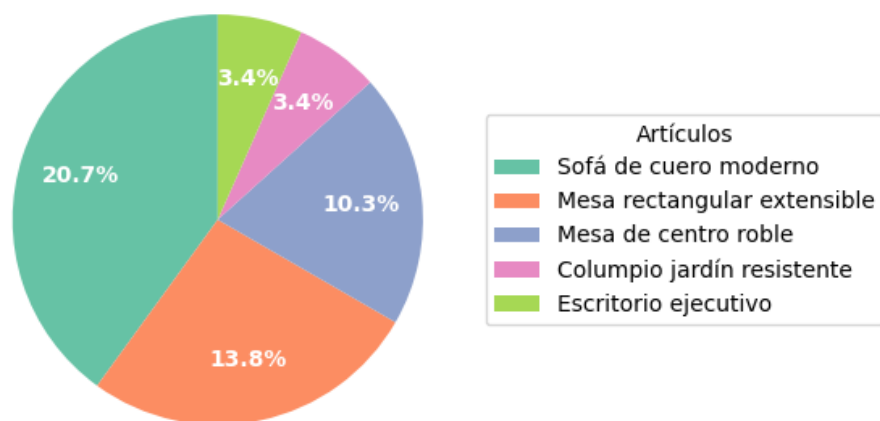


Figura 6: Top 5 artículos mas vendidos respecto al total de ventas

## Conclusiones

### Montiel Juárez Oscar Iván

Durante este proyecto, consolidamos nuestros conocimientos sobre el lenguaje PostgreSQL, así como sobre la administración de bases de datos y la gestión de usuarios. Al desarrollar el proyecto desde cero, tuvimos la oportunidad de realizar un análisis de requerimientos completo, lo cual fue fundamental, ya que nos permitió entender el contexto de la base de datos desde el inicio. Este fue, sin duda, uno de los mayores aciertos, pues comprobamos que, al contar con un buen análisis de requerimientos, el resto del desarrollo se vuelve mucho más sencillo.

Aunque se realizaron algunas modificaciones a nuestro modelo relacional inicial, este sirvió como una excelente guía durante todo el proceso. Me queda claro que hacer un buen análisis equivale a tener resuelta gran parte del trabajo. Si bien aún no somos expertos en SQL, su sintaxis y lógica son similares a otros lenguajes de programación, por lo que, con un poco de documentación adicional, pudimos implementar las funciones necesarias.

Entre las áreas de oportunidad que identificamos, una de las más relevantes fue la necesidad de aprender a depurar correctamente en PostgreSQL, ya que esto nos habría sido de gran ayuda al momento de corregir funciones que no funcionaban como esperábamos. Además, al no utilizar transacciones en una parte del desarrollo, accidentalmente eliminamos una tabla, lo que nos hizo entender la importancia de utilizar transacciones para proteger la integridad de la base de datos. Estos errores nos permitieron aprender lecciones valiosas que, sin duda, aplicaremos en futuros proyectos.

### Martínez García Jose Eduardo

Este proyecto representó un recorrido integral por los temas clave del curso, permitiéndonos aplicar tanto los fundamentos teóricos como las herramientas prácticas en un contexto realista. A lo largo del desarrollo, trabajamos con un caso de estudio centrado en la digitalización de los sistemas de una mueblería, abarcando procesos de ventas, operaciones y administración. Cada fase del proyecto nos exigió utilizar diferentes metodologías y técnicas, desde el análisis inicial hasta la implementación final.

En la etapa de diseño, el método tabular nos ayudó a organizar la información dispersa y traducir los requerimientos del cliente en una estructura clara. Complementamos esto con el modelo Entidad Relación, que nos permitió visualizar gráficamente las relaciones entre los datos, y el modelo relacional, que nos acercó a la implementación concreta en PostgreSQL. Durante la codificación, el uso de DDL nos facilitó la creación de la estructura de la base de datos, mientras que el DML nos permitió manipular los datos y detectar errores durante la inserción, lo que enriqueció nuestro aprendizaje.

Uno de los aspectos más desafiantes fue la implementación de funciones y triggers con PL/pgSQL, que nos obligó a pensar en soluciones específicas para problemas puntuales, como validaciones y automatizaciones. Aunque el tema de normalización no se incluyó, el proyecto sí cubrió una amplia gama de conceptos esenciales, consolidando nuestra comprensión de las bases de datos relacionales.

En mi opinión, el resultado final refleja un esfuerzo bien orientado, donde los conocimientos adquiridos se aplicaron de manera coherente y práctica. Si bien siempre hay margen para mejorar, el proyecto cumplió con su objetivo de integrar teoría y práctica, preparándonos para enfrentar desafíos similares en entornos profesionales. La experiencia no solo reforzó nuestras habilidades técnicas, sino que también nos enseñó la importancia de un enfoque estructurado y metódico en el desarrollo de sistemas basados en datos.

## Ramos Cabrera Paul Manuel

Durante el desarrollo de este proyecto se construyó una base de datos funcional, con el propósito de mejorar la gestión de una mueblería y sus sucursales. Para ello, fue necesario pasar por diferentes etapas: primero se analizaron los requerimientos, luego se generó el modelo entidad relación y su transformación al modelo relacional, y finalmente se implementaron las tablas, restricciones, funciones y triggers.

El trabajo en equipo implicó cierto grado de coordinación, especialmente al integrar las aportaciones de cada integrante; en ocasiones resultó difícil organizar y unificar lo desarrollado, pero esto permitió fortalecer la comunicación y los procesos de revisión. Al inicio, adaptarse a pgAdmin 4 presentó un reto, ya que hasta ese momento había trabajado exclusivamente en terminal con Oracle; sin embargo, esta experiencia facilitó la aplicación práctica de los conceptos vistos en clase, en especial las funciones, consultas y triggers. Además, se pudieron consolidar conocimientos un poco externos a lo visto; por ejemplo, aprendí a cómo usar Python conectado a una base de datos Postgres, con una librería llamada "psycopg2", lo cual fue interesante porque es posible acceder a las tablas de la base, y además trabajar con los datos presentes en las tablas. También se implementó una forma de obtener los dashboards a través de graficación con la librería matplotlib, algo con lo que ya tenía un poco de experiencia. De hecho, me recordó a mis clases de EDA II.

Aunque la corrección de errores fue a veces frustrante, pues requería depurar scripts y ajustar parámetros, también resultó enriquecedora y divertida al ver cómo cada ajuste mejoraba el comportamiento de la base de datos. En conjunto, este proyecto no solo consolidó conocimientos teóricos, sino que brindó habilidades prácticas en diseño, implementación y trabajo colaborativo.

## Jurado Rosas Josue David

Durante el desarrollo de este proyecto de base de datos se presentaron varios retos, aprendizajes y también aciertos importantes. Uno de los principales desafíos fue la coordinación con los compañeros del equipo, especialmente porque cada quien fue trabajando ciertas partes de manera individual, y al momento de integrar todo, resultaba complicado que no se notara un estilo "parchado" o inconexo. Esto nos llevó a invertir tiempo extra en revisar el código, homologar nombres, formatos y estilos, para que la solución final fuera coherente y funcional.

Otra dificultad importante fue cuando, por accidente o error, se eliminó una de las tablas del modelo ya avanzado. Esto implicó tener que reconstruirla desde cero, volver a definir sus claves primarias y foráneas, validar las dependencias con otras tablas y asegurarnos de que todo siguiera funcionando con los triggers y las vistas que ya se habían hecho. Además, tuvimos que adaptar todos los datos relacionados que ya se habían insertado y ajustar el resto del código que dependía de esa tabla. Fue una situación que nos retrasó, pero también nos ayudó a tener más cuidado con los cambios y a entender mejor la lógica interna del sistema.

También tomamos la decisión de cambiar el tipo de dato del campo RFC de BIGINT a VARCHAR, lo cual, aunque parecía sencillo, implicó revisar y modificar varias tablas relacionadas, funciones y vistas para que todo siguiera funcionando correctamente. Este tipo de ajustes nos hizo ver la importancia de planear bien los tipos de datos desde el inicio.

Como reto técnico adicional, uno de los más interesantes fue implementar correctamente los triggers y funciones para que ciertas operaciones se hicieran de forma automática, como la actualización del stock, los totales de una venta y la generación del folio de factura. Entender cómo y cuándo se deben usar los triggers, y cómo evitar errores como la



duplicación de montos, fue algo que requirió bastante atención.

Entre los aciertos, puedo destacar el uso de vistas bien estructuradas para mostrar la información de los tickets, así como el diseño de funciones que realmente automatizaron procesos importantes y evitaron errores humanos. También fue un acierto dividir el trabajo desde el inicio, y aunque eso nos trajo problemas al momento de unirlos, al final logramos consolidar un sistema funcional que cumple con los requerimientos del proyecto.

### **Santos Jiménez Alejandro**

A lo largo del desarrollo de este proyecto, se logró aplicar de manera práctica los conocimientos adquiridos en la asignatura de Bases de Datos. Comenzamos con la elaboración del modelo Entidad Relación, posteriormente se tradujo al modelo relacional, y finalmente se implementó en una base de datos real. En nuestro caso, utilizamos Google Cloud, lo cual nos permitió conectarnos de forma remota; en mi caso particular, trabajé desde PgAdmin 4.

El proyecto abordó un problema real con requerimientos complejos y específicos. Se diseñó e implementó una base de datos robusta y funcional que cumplió con dichos requerimientos.

Entre los principales logros se encuentran la creación de vistas especializadas para facilitar el acceso a datos clave, el uso de procedimientos almacenados y triggers para automatizar procesos críticos, así como la integración de mecanismos de validación para garantizar la consistencia de la información. Además, el desarrollo del dashboard permitió transformar los datos almacenados en representaciones visuales útiles para la toma de decisiones por parte del negocio.

Durante el proceso se enfrentaron varios retos técnicos relacionados con el modelado de datos y la implementación de restricciones lógicas. Estos desafíos fueron superados gracias al trabajo en equipo, la capacidad de análisis y el uso adecuado de las herramientas disponibles.

Personalmente, este proyecto me gustó mucho, ya que permitió poner en práctica todo lo aprendido a lo largo del curso, a pesar de los inconvenientes surgidos durante el semestre.