



**Universidad Nacional Autónoma de
México
Facultad de Ingeniería**

Bases de Datos

Profesor: Ing. Fernando Arreola Franco

Semestre: 2025-2

**Proyecto Final
Sistema de Gestión de Mueblería**

Grupo: 1

Alumnos:

Flores Herrera Oswaldo
Valencia Lerdo Dalia Jimena

Cd. Universitaria a 25 de Mayo de 2025

Proyecto Final: Sistema de Gestión para Mueblería: Casa Tzalam S.A.

Integrantes:
Flores Herrera Oswaldo
Valencia Lerdo Dalia Jimena

Mayo 2025

Resumen

Este documento presenta el diseño e implementación de un sistema de gestión de bases de datos para la mueblería “Casa Tzalam S.A.”, que busca digitalizar sus operaciones y centralizar la información de sus múltiples sucursales. El sistema permite gestionar artículos, proveedores, ventas, clientes y empleados, cumpliendo con los requerimientos funcionales establecidos. Se implementaron triggers, vistas y un dashboard de visualización para apoyar la toma de decisiones. El proyecto sigue los principios de normalización y garantiza la integridad de los datos mediante restricciones y validaciones.

Índice

1. Introducción	3
1.1. Contexto y Problema	3
1.2. Objetivos	3
1.2.1. Objetivo General	3
1.2.2. Objetivos Específicos	3
1.3. Alcance	4
2. Plan de Trabajo	4
2.1. Metodología	4
2.2. Cronograma	5
2.3. Distribución de Trabajo	5
3. Diseño de la Base de Datos	6
3.1. Modelo Entidad-Relación	6
3.2. Modelo Relacional	7
3.3. Diccionario de Datos	7
4. Implementación	8
4.1. Script de Creación	8
4.2. Restricciones e Integridad	8
4.3. Índices	9
4.4. Triggers y Funciones	9
4.5. Vistas	10
5. Aplicación	11
5.1. Dashboard	11
5.2. Manual de Usuario Básico	11
5.2.1. Registrar una Venta	11
5.2.2. Consultar Stock	12
6. Resultados	13
6.1. Resultados Obtenidos	13
6.2. Dificultades y Soluciones	14
6.3. Trabajo Futuro	14
7. Conclusiones	15
7.1. Conclusiones Personales	15
A. Código Completo	16
A.1. Scripts SQL	16

1. Introducción

1.1. Contexto y Problema

Casa Tzalam S.A. es una reconocida cadena de mueblerías con presencia en 10 estados de la república mexicana. A lo largo de los años, ha crecido gracias a su enfoque en la calidad y el diseño de muebles, pero actualmente enfrenta un reto importante: todos sus registros se llevan de forma física y cada sucursal gestiona su información por separado. Esto ha provocado

varios problemas, como pérdida de datos, inconsistencias en los inventarios y dificultades para obtener reportes claros y unificados sobre lo que ocurre en toda la empresa. Por eso, el proyecto se enfoca en diseñar una base de datos

que le permita a Casa Tzalam dar el paso hacia la digitalización. La idea es contar con un sistema centralizado que unifique la información de todas las sucursales y haga más eficiente la operación del negocio.

Con esta propuesta de solución se busca:

- Gestionar el inventario en todas las sucursales
- Registrar ventas de manera electrónica
- Generar reportes financieros, de forma automática y confiable
- Mantener información centralizada de clientes y empleados para un mejor manejo

1.2. Objetivos

1.2.1. Objetivo General

Diseñar e implementar un sistema de gestión de bases de datos que centralice las operaciones de “Casa Tzalam S.A.” y que permita manejar eficientemente el registro, consulta, actualización y reporte de la información relacionada con inventarios, ventas, compras, clientes y proveedores, asegurando la integridad, disponibilidad y seguridad de los datos para apoyar la toma de decisiones operativas.

1.2.2. Objetivos Específicos

- Modelar la estructura de datos que represente todas las entidades del negocio

- Implementar mecanismos para garantizar la integridad de la información
- Desarrollar procedimientos automatizados para las operaciones recurrentes
- Crear un dashboard para visualización de indicadores clave

1.3. Alcance

El sistema cubrirá:

- Gestión de artículos y categorías
- Registro de proveedores
- Proceso completo de ventas
- Administración de clientes
- Control de empleados y sucursales
- Generación de reportes básicos

2. Plan de Trabajo

2.1. Metodología

Se utilizó una metodología ágil adaptada para proyectos de bases de datos, con las siguientes fases:

1. Análisis de requerimientos
2. Diseño conceptual (MER)
3. Diseño lógico (MR)
4. Implementación física
5. Pruebas y ajustes
6. Documentación y presentación

2.2. Cronograma

Fase	Responsable	Fecha Inicio	Fecha Fin
Análisis	Todos	14-Abr	20-Abr
Diseño MER	Todos	21-Abr	27-Abr
Diseño MR	Todos	28-Abr	04-May
Implementación	Todos	05-May	17-May
Pruebas	Todos	18-May	21-May
Documentación	Todos	22-May	24-May

Cuadro 1: Cronograma del proyecto

2.3. Distribución de Trabajo

Como un equipo conformado por dos integrantes, adoptamos un enfoque colaborativo en el que ambos participamos activamente en todas las fases del proyecto. Desde el análisis de requerimientos hasta la implementación final, nos mativukos involucrados de manera equitativa, asegurando que ambos estuvieramos involucrados en la toma de decisiones y comprendieramos cada una de estas. Dividimos las responsabilidades principales de forma estraté-

ca: uno de nosotros se centró más en el diseño e implementación de la base de datos, mientras que el otro puso mayor atención en el análisis de datos y la creación del dashboard. Aun así, ambos nos involucramos en cada parte del proyecto. Revisamos el trabajo del otro, hicimos sugerencias y colaboramos en tareas como la creación de tablas, el desarrollo de vistas y la programación de triggers. Trabajar en un equipo pequeño fue una ventaja, pudimos comu-

nicarnos constantemente, adaptarnos rápido a los cambios y mantener una buena coordinación. Al estra involucrados en todas las etapas del proyecto, no solo ayudó a lograr un buen resultado, también nos permitió aprender más durante el proceso.

3. Diseño de la Base de Datos

3.1. Modelo Entidad-Relación

El modelo entidad-relación captura las principales entidades del negocio y sus relaciones.

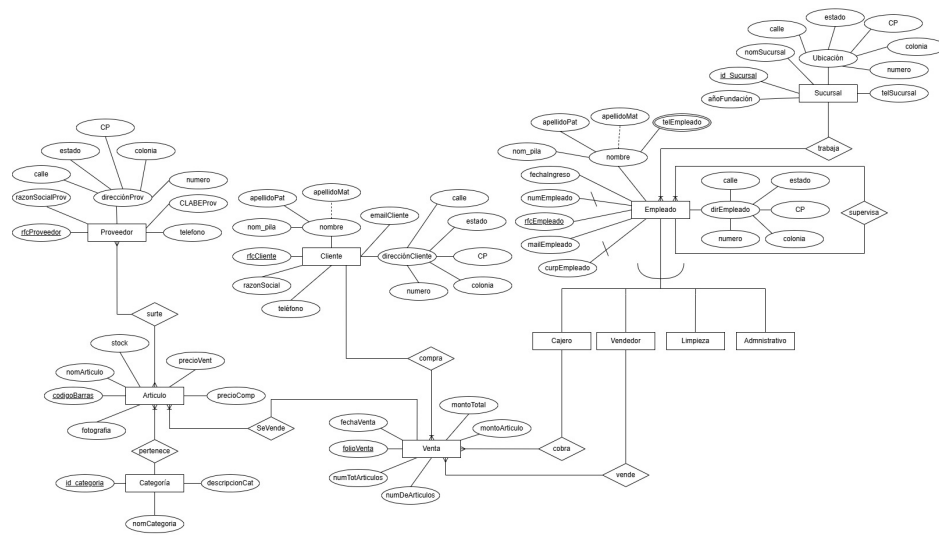


Figura 1: Modelo Entidad-Relación del sistema.

Principales entidades:

- **Artículo:** Productos que comercializa la mueblería
- **Categoría:** Clasificación de artículos
- **Proveedor:** Empresas que surten los artículos
- **Cliente:** Personas o empresas que compran en la mueblería
- **Empleado:** Personal de la empresa
- **Sucursal:** Tiendas físicas
- **Venta:** Transacciones comerciales

3.2. Modelo Relacional

El modelo relacional resultante consta de 8 tablas principales:

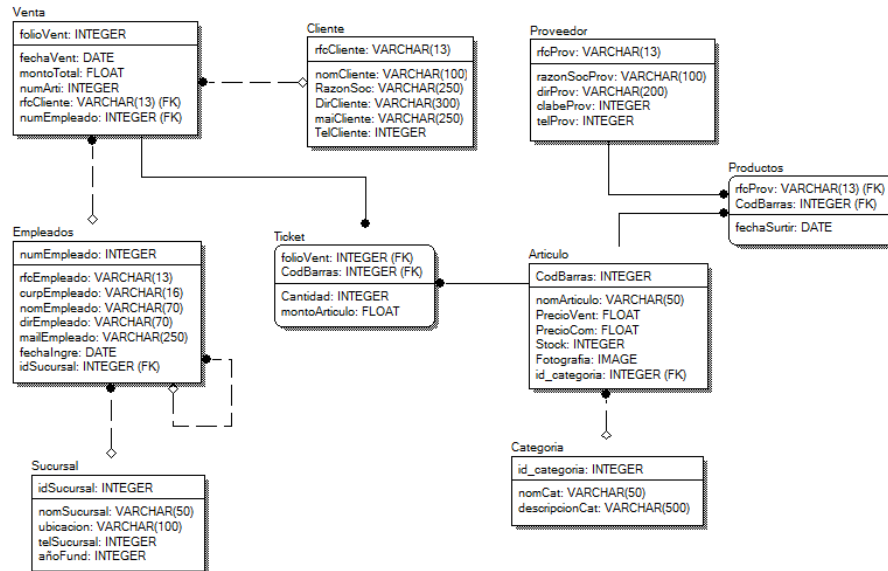


Figura 2: Modelo Entidad-Relación del sistema

3.3. Diccionario de Datos

A continuación se presenta un fragmento del diccionario de datos completo:

Tabla	Atributo	Tipo	Descripción
articulo	codigo_barras	bigint	Identificador único del artículo
articulo	nom_articulo	varchar(100)	Nombre comercial del artículo
articulo	precio_venta	numeric(10,2)	Precio al público con 2 decimales
articulo	stock	integer	Cantidad disponible en inventario
categoria	id_categoria	serial	Llave primaria autoincremental
categoria	nom_categoria	varchar(100)	Nombre de la categoría
venta	folio_venta	varchar(10)	Folio con formato MBL-001
venta	fecha_venta	timestamp	Fecha y hora de la venta
venta	monto_total	numeric(10,2)	Suma total de la venta

Cuadro 2: Fragmento del diccionario de datos

4. Implementación

4.1. Script de Creación

Se implementó el siguiente esquema de base de datos en PostgreSQL:

El sistema de gestión para Casa Tzalam S.A. se implementó mediante un conjunto de scripts SQL diseñados para crear y configurar la base de datos en PostgreSQL. Estos scripts se organizaron en módulos lógicos que abarcan desde la creación de la estructura básica hasta la programación de funcionalidades. El script inicial establece el entorno de la base de datos, configurando

la codificación UTF-8 para soportar caracteres especiales y el esquema de búsqueda predeterminado. La base de datos se creó desde cero con el nombre DBMB para garantizar un entorno limpio. Las tablas principales incluyen:

- **Artículo:** Contiene información de productos como código de barras, nombre, precios de venta y compra, stock y categoría.
- **Categoría:** Almacena la clasificación de los artículos con campos para nombre y descripción.
- **Cliente:** Registra datos personales, dirección y contacto de los clientes, incluyendo un campo generado para la razón social.
- **Empleado:** Gestiona información del personal, con validación de tipos de empleado mediante una restricción CHECK.
- **Venta y Ticket:** Administran el proceso de ventas, con triggers para actualizar automáticamente los totales y validar el stock.

4.2. Restricciones e Integridad

Se definieron restricciones para garantizar la integridad referencial y garantizar la integridad de los datos, como llaves primarias en todas las tablas y llaves foráneas que vinculan artículos con categorías, ventas con empleados y tickets con artículos:

- **LLaves primarias y foráneas:** Para mantener relaciones válidas
- **CHECK constraints:** Para validar dominios (ej. tipo_empleado)
- **UNIQUE:** Para campos como RFC y CURP de empleados
- **NOT NULL:** Para campos obligatorios

- **Acciones en cascada:** Gestionan automáticamente la propagación o anulación de cambios en relaciones

4.3. Índices

La creación de índices en las tablas clave de la base de datos, una estrategia fundamental para optimizar el rendimiento del sistema. Estos índices funcionan como atajos que permiten acelerar significativamente las operaciones de búsqueda en las columnas más frecuentemente consultadas. En concreto, se crea un índice sobre el nombre de los artículos (*idx_articulo_nombre*) para agilizar la búsqueda de productos por su denominación, otro sobre el RFC de los clientes (*idx_cliente_rfc*) que facilita la localización rápida de clientes específicos, un índice en el número de empleado (*idx_empleado_num*) que optimiza el acceso a los registros del personal, y finalmente un índice sobre la fecha de venta (*idx_venta_fecha*) que mejora el rendimiento de las consultas que filtran información por períodos temporales. Estos índices actúan como estructuras organizadas que permiten al motor de la base de datos encontrar rápidamente los registros sin necesidad de escanear todas las filas de las tablas, reduciendo así el tiempo de respuesta de las consultas y mejorando la eficiencia general del sistema, especialmente importante en operaciones frecuentes como búsquedas de productos, consulta de historiales de clientes o generación de reportes por períodos específicos.

4.4. Triggers y Funciones

Se implementaron varios triggers para automatizar procesos:

Funciones Las funciones en este sistema están diseñadas para automatizar procesos clave y generar reportes:

- **reporteInventario():** Analiza el stock de productos, clasificándolo en estados como “AGOTADO” o “BAJO”, y sugiere acciones de reposición, combinando datos de inventario con ventas recientes.
- **venta():** Gestiona todo el proceso de una transacción: valida empleados, verifica stock, calcula totales, registra la venta y actualiza el inventario, asegurando integridad en cada paso.
- **reporteIngresos():** Genera reportes financieros mensuales por sucursal, mostrando tendencias y comparativas.

Estas funciones encapsulan lógica compleja para simplificar operaciones recurrentes y garantizar consistencia en los datos.

Triggers Los triggers actúan como guardianes automáticos de la integridad de los datos:

- **tr_validar_sucursal_empleados**: Verifica que vendedor y cajero pertenezcan a la misma sucursal antes de registrar una venta, evitando inconsistencias.
- **validar_stock_articulo**: Rechaza ventas si no hay suficiente inventario, mostrando mensajes claros al usuario.
- **actualizar_totales_venta**: Mantiene sincronizados los montos totales en la tabla venta cuando se modifican los tickets asociados, asegurando que los cálculos siempre reflejen los detalles actualizados.

Estos triggers operan de forma transparente, ejecutándose antes o después de ciertas acciones (INSERT/UPDATE/DELETE) para aplicar reglas de negocio críticas sin intervención manual.

4.5. Vistas

Las vistas del sistema son herramientas prácticas que facilitan el trabajo diario: *vw_stock_art* funciona como un semáforo de inventario, destacando productos con stock bajo o agotados; *vw_ticket_venta* actúa como un comprobante digital completo, mostrando artículos vendidos, vendedores, sucursal y datos del cliente en un solo lugar; y *vw_organigrama* dibuja el árbol de personal de la empresa, clarificando quién reporta a quién. Estas vistas

ahorran tiempo al evitar consultas repetitivas, garantizan que todos usen la misma información actualizada y simplifican la generación de reportes, haciendo más eficiente el análisis de ventas, el control de inventario y la gestión del personal.

5. Aplicación

5.1. Dashboard

El dashboard del sistema integra información clave a través de las siguientes consultas:

- **Ingresos mensuales por sucursal:** Agrupa las ventas por mes y sucursal, mostrando los ingresos totales para un año específico (2025 en este caso), ordenados por sucursal y mes.
- **Artículos más vendidos por categoría:** Proporciona un ranking de productos mostrando las cantidades vendidas e ingresos generados, agrupados por categoría y ordenados por volumen de ventas.
- **Comparativo mensual de sucursales:** Muestra el desempeño de cada sucursal en el mes en curso, incluyendo el número total de ventas e ingresos generados, ordenados por volumen de ingresos.
- **Indicadores clave (KPIs):** Calcula métricas esenciales como:
 - Número total de ventas
 - Ingresos totales
 - Ticket promedio
 - Clientes activos
 - Artículos con stock bajo

para los últimos 30 días.

- **Ventas totales por sucursal:** Proporciona una visión consolidada de los ingresos por cada ubicación.

Estas vistas proporcionan los datos fundamentales que alimentan las visualizaciones del panel, garantizando información actualizada para la toma de decisiones.

5.2. Manual de Usuario Básico

5.2.1. Registrar una Venta

Para registrar una nueva venta en el sistema:

1. Ejecutar la función `venta()` con los siguientes parámetros:

- ID del vendedor (ej: 'EMP002')
- ID del cajero (ej: 'EMP001')
- RFC del cliente (ej: 'LORE930606MDF')
- Array de códigos de barras de artículos (ej: ARRAY[140433330034, 140433330006])
- Array de cantidades correspondientes (ej: ARRAY[2, 1])

Ejemplo:

```
SELECT venta(
    'EMP002',
    'EMP001',
    'LORE930606MDF',
    ARRAY[140433330034, 140433330006],
    ARRAY[2, 1]
);
```

2. El sistema procesará automáticamente:

- Validación de stock disponible
- Cálculo de totales
- Actualización de inventario
- Generación del folio de venta

5.2.2. Consultar Stock

Para verificar el estado del inventario:

1. Consultar la vista `vw_stock_art` que muestra:

- Código de barras y nombre del artículo
- Precios de venta y compra
- Estado de disponibilidad (Disponible/Baja disponibilidad/No disponible)
- Cantidad exacta en inventario

Ejemplo básico:

```
SELECT * FROM vw_stock_art;
```

2. Para filtrar resultados:

- Por disponibilidad: `WHERE disponibilidad LIKE 'Baja disponibilidad%'`
- Por categoría: `WHERE id_categoria = X`
- Por rango de precios: `WHERE precio_venta BETWEEN Y AND Z`

6. Resultados

6.1. Resultados Obtenidos

Durante el desarrollo de nuestro proyecto logramos implementar un sistema que gestionara una base de datos integral de una mueblería, logrando transformar procesos manuales en soluciones automatizadas. Dentro de nuestros resultados mas significativos destacaremos Una centralización efi-

ciente donde consolidamos toda la información operacional ejemplo de ello los inventarios, ventas, clientes y proveedores en una plataforma unificada, eliminando la redundancia y facilitando el acceso a los datos en tiempo real, de igual forma la automatización en los procesos optimizo las tareas criticas como el control de stock, facturación y seguimiento de pedidos, reduciendo errores humanos y mejorando la productividad del negocio esto trae consigo un análisis estratégico de generación de reportes personalizados como lo son ventas por periodo y la producción mas demandada, de igual manera el implementar seguridad e integridad en los datos permitió validar mecanismos como backups periódicos y perfiles de usuario para garantizar la protección y consistencias de la información. La implementación de este sistema de ges-

tión de bases de datos no solo resuelve las necesidades operativas inmediatas de la mueblería, sino que establece un marco robusto en el crecimiento sostenible gracias a la s bases de datos. Este caso de estudio nos evidencia como la transición hacia un modelo digitalizado centrado en bases de datos estructura de forma útil y moderniza los negocios tradicionales. De igual forma diferencia estratégicamente el impulso a explorar integraciones de este tipo de herramientas para profundizar en el análisis y automatización de decisiones.

6.2. Dificultades y Soluciones

- **Problema:** Validación compleja de vendedor y cajero de misma sucursal
- **Solución:** Implementación de trigger `validar_sucursal_empleados`
- **Problema:** Generación automática de folios con formato MBL-001
- **Solución:** Uso de secuencia y función de formato en DEFAULT
- **Problema:** Actualización del inventario después de cada venta
- **Solución:** Implementación de trigger `actualizar_stock_venta()` y `actualizar_stock_articulo()` que reduce automáticamente el stock al registrar una venta y verifican disponibilidad
- **Problema:** Inconsistencias en los totales de ventas al modificar tickets
- **Solución:** Implementación del trigger `tr_actualizar_totales` con la función `actualizar_totales_venta()` que:
 - Recalcula automáticamente el monto total sumando todos los sub-totales de tickets
 - Actualiza la cantidad total de artículos vendidos
 - Se ejecuta después de cada inserción, actualización o eliminación en tickets
 - Garantiza coherencia entre ventas y sus tickets asociados

6.3. Trabajo Futuro

El sistema implementado ofrece una base sólida para futuras mejoras, comenzando con la integración a plataformas de e-commerce mediante APIs que conectarán nuestra base de datos con sistemas en línea, permitiendo la sincronización en tiempo real de inventarios, precios y pedidos. Este desarrollo puede complementarse con interfaces gráficas más robustas, ya sea mediante un portal web o una aplicación móvil para vendedores, facilitando el acceso remoto a datos, la generación de cotizaciones y la gestión de entregas. Como proyecto culminante, se plantea la migración completa a la nube, aprovechando sus ventajas de escalabilidad automática, mayor capacidad de almacenamiento, mantenimiento simplificado y reducción de costos en infraestructura física, todo ello respaldado por protocolos de seguridad y disponibilidad garantizada.

7. Conclusiones

7.1. Conclusiones Personales

Flores Herrera Oswaldo

Este proyecto representó un reto significativo pero también una gran oportunidad de aprendizaje. Una de las dificultades más grandes que enfrenté fue el diseño de los triggers para garantizar la integridad de los datos, especialmente en el proceso de ventas, donde era crucial validar el stock, calcular totales y asegurar que vendedor y cajero pertenecieran a la misma sucursal. Al principio, tuvimos problemas con la sintaxis de PL/pgSQL, pero después de investigar y practicar, logramos implementar funciones robustas como `venta()` y `actualizar_totales_venta()`. Uno de los puntos más impor-

tantes del proyecto fue la creación del dashboard de visualización, donde pude aplicar mis conocimientos de SQL para generar consultas complejas que muestran KPIs clave como ingresos mensuales y artículos más vendidos. Aprendí la importancia de planificar con anticipación, ya que algunos cambios en el modelo relacional requirieron ajustes en múltiples componentes del sistema. Lo más relevante desde mi punto de vista fue comprobar cómo conceptos teóricos se traducen en soluciones concretas para la resolución de problemas reales.

Valencia Lerdo Dalia Jimena

El desarrollo de este sistema me permitió aplicar conceptos teóricos a un caso real. Uno de los retos más grandes fue normalizar la base de datos para evitar redundancias, especialmente en tablas como `Artículo` y `Venta`, donde era fácil caer en diseños denormalizados. Una función que como equipo nos costó mucho implementar fue `venta()`, ya que requería coordinar múltiples operaciones críticas en una sola transacción. Un acierto importante fue la

implementación de las vistas, especialmente `vw_ticket_venta`, que simplifica la generación de comprobantes de compra-venta. Me di cuenta de lo útil que es documentar cada paso, ya que cuando revisábamos el código en equipo, tener comentarios claros facilitó la colaboración. Aprendí que la comunicación constante es clave en proyectos colaborativos, y que probar exhaustivamente cada componente evita errores futuros. Este proyecto me demostró el impacto que un buen diseño de bases de datos puede tener en la eficiencia de un negocio.

A. Código Completo

A.1. Scripts SQL

Los scripts completos están disponibles en el repositorio del proyecto:

- `Muebleria.sql`: Backup de la creación de esquema y tablas
- `Carpeta CSV`: Datos de prueba
- `programacion.sql`: Triggers, funciones y vistas
- `Contenido visual`: Diagramas, imágenes y gráficas