UNIVERSIDAD TECNOLÓGICA DE PANAMÁ

FACULTAD DE INGENIERÍA DE SISTEMAS COMPUTACIÓNALES

LIC. Desarrollo y gestión de Software

PROYECTO – Base de Datos 2

Trabajo escrito:

Sistema de Reciclaje

Profesora: Gema Castillo Sánchez

Integrantes:

Alex Hernández ([alex.hernandez3@utp.ac.pa](mailto:alex.hernandez3@utp.ac.pa) 6720-1761)

Francisco Herrera ([francisco.herrera2@utp.ac.pa](mailto:francisco.herrera2@utp.ac.pa) 6775-0422)

José Rodríguez ([Jose.rodriguez100@utp.ac.pa](mailto:Jose.rodriguez100@utp.ac.pa) 6519-3032)

Iván Wu ([ivan.wu@utp.ac.pa](mailto:ivan.wu@utp.ac.pa) 6312-9117)

PRIMER SEMESTRE

2024

**ÍNDICE**

1. Introducción **1**

2. Generalidades del proyecto **2**

- Objetivo del Proyecto

- Justificación

- Alcance del Proyecto

- Beneficios Esperados

3. Metodología **3**

- Análisis de requisitos

- Diseño del sistema

- Implementación

- Pruebas y ajustes

- Despliegue y capacitación

4. Recursos Necesarios **3**

5. Cronograma **3**

6. Diseño del E/R **4**

7. Script de la base de datos **5**

8. Principales Recomendaciones (Cómo desarrolladores) sobre el uso de DB **12**

- Creación y Uso de Vistas

- Funciones y Procedimientos Almacenados

- Resolución de Problemas Comunes

9. Vistas, funciones, restricciones, etc.

sobre nuestro proyecto de base de datos **14**

10. Conclusión **20**

11. Referencias **21**

**Introducción**

En la era moderna, la gestión eficiente de los recursos naturales se ha convertido en una prioridad global. El reciclaje de materiales juega un papel crucial en este esfuerzo, permitiendo la recuperación de recursos valiosos y la reducción del impacto ambiental asociado con la extracción y procesamiento de materias primas vírgenes. Este proyecto de base de datos se centra en la creación de un sistema integral para la gestión de materiales reciclables, diseñado para optimizar la recolección, clasificación, y procesamiento de estos materiales con el fin de transformarlos en productos comerciales viables.

El objetivo principal de este proyecto es desarrollar base de datos robusta que permita a una empresa especializada en reciclaje no solo rastrear eficazmente el flujo de materiales desde su recolección inicial hasta su transformación final en productos comerciales, sino también mejorar la eficiencia operativa y la toma de decisiones estratégicas. Esta base de datos facilitará la captura y análisis de datos relevantes en cada etapa del ciclo de vida de los materiales reciclables, proporcionando información crítica para optimizar procesos, minimizar residuos y maximizar la rentabilidad.

El proyecto se enfoca en integrar diversas funcionalidades dentro de la base de datos, incluyendo la gestión de inventario de materiales entrantes, el seguimiento de la calidad y estado de los materiales durante su procesamiento, la planificación de la producción basada en la demanda del mercado, y la generación de informes y análisis para apoyar la toma de decisiones estratégicas.

**Generalidades del proyecto**

Objetivo del Proyecto:

Desarrollar un sistema de bases de datos eficiente que permita gestionar y optimizar los procesos de recolección, clasificación, procesamiento y comercialización de materiales reciclables, garantizando la trazabilidad y la eficiencia en cada etapa del ciclo de reciclaje.

Justificación:

La problemática de la acumulación de residuos y su impacto negativo en el medio ambiente es un tema de creciente preocupación global. Este proyecto busca ofrecer una solución sostenible mediante el reciclaje de basura, convirtiendo los desechos en productos útiles. La implementación de un sistema de bases de datos permitirá una gestión más eficaz de los recursos y procesos, mejorando la productividad y contribuyendo a la economía circular.

Alcance del Proyecto

El proyecto abarcará las siguientes áreas:

1. Recolección de residuos: Registro y seguimiento de los puntos de recolección y cantidades de residuos recogidos.

2. Clasificación de materiales: Identificación y clasificación de los diferentes tipos de materiales reciclables.

3. Procesamiento: Gestión de los procesos de transformación de residuos en materiales comerciales.

4. Almacenamiento y distribución: Control del inventario de materiales reciclados y su distribución para la venta.

5. Comercialización: Registro de ventas y gestión de clientes.

Beneficios Esperados:

- Económicos: Generación de ingresos a través de la venta de materiales reciclados.

- Ambientales: Reducción de la cantidad de residuos que terminan en vertederos.

- Sociales: Concienciación y educación de la comunidad sobre la importancia del reciclaje y la sostenibilidad.

Metodología

El desarrollo del proyecto seguirá una metodología en etapas, incluyendo:

1. Análisis de requisitos: Identificación de las necesidades específicas del sistema de gestión.

2. Diseño del sistema: Creación de modelos de datos y diseño de la arquitectura del sistema.

3. Implementación: Desarrollo del sistema de bases de datos y las interfaces de usuario.

4. Pruebas y ajustes: Verificación del funcionamiento del sistema y ajustes necesarios.

5. Despliegue y capacitación: Implementación del sistema en la empresa y capacitación al personal.

Recursos Necesarios

- Humanos: Personal de TI, ingenieros de datos, operarios de reciclaje, equipo administrativo.

- Tecnológicos: Servidores, software de bases de datos, computadoras, dispositivos de entrada de datos.

- Económicos: Inversión inicial para el desarrollo e implementación del sistema.

Cronograma

El proyecto se desarrollará en un período de 12 meses, dividido en las siguientes fases:

1. Mes 1-2: Análisis de requisitos

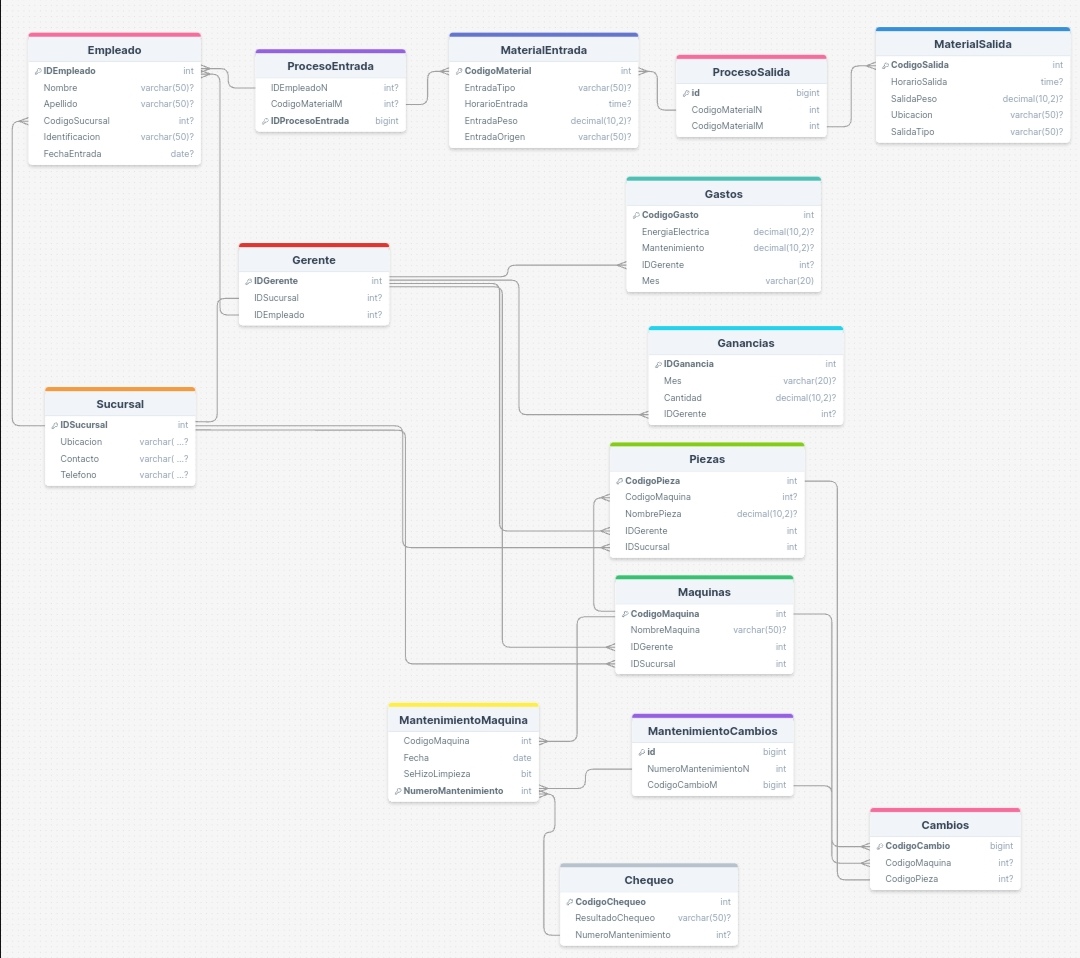
2. Mes 3-4: Diseño del sistema

3. Mes 5-8: Implementación del sistema

4. Mes 9-10: Pruebas y ajustes

5. Mes 11-12: Despliegue y capacitación

**Diseño del E/R**



**Script de la base de datos**

create database my\_database3

go

use my\_database3

go

CREATE TABLE "ProcesoSalida"(

"id" BIGINT NOT NULL,

"CodigoMaterialN" INT NOT NULL,

"CodigoMaterialM" INT NOT NULL

);

ALTER TABLE

"ProcesoSalida" ADD CONSTRAINT "procesosalida\_id\_primary" PRIMARY KEY("id");

CREATE TABLE "Piezas"(

"CodigoPieza" INT NOT NULL,

"CodigoMaquina" INT NULL,

"NombrePieza" DECIMAL(10, 2) NULL,

"IDGerente" INT NOT NULL,

"IDSucursal" INT NOT NULL

);

ALTER TABLE

"Piezas" ADD CONSTRAINT "piezas\_codigopieza\_primary" PRIMARY KEY("CodigoPieza");

CREATE TABLE "Maquinas"(

"CodigoMaquina" INT NOT NULL,

"NombreMaquina" VARCHAR(50) NULL,

"IDGerente" INT NOT NULL,

"IDSucursal" INT NOT NULL

);

ALTER TABLE

"Maquinas" ADD CONSTRAINT "maquinas\_codigomaquina\_primary" PRIMARY KEY("CodigoMaquina");

CREATE TABLE "MaterialSalida"(

"CodigoSalida" INT NOT NULL,

"HorarioSalida" TIME NULL,

"SalidaPeso" DECIMAL(10, 2) NULL,

"Ubicacion" VARCHAR(50) NULL,

"SalidaTipo" VARCHAR(50) NULL

);

ALTER TABLE

"MaterialSalida" ADD CONSTRAINT "materialsalida\_codigosalida\_primary" PRIMARY KEY("CodigoSalida");

CREATE TABLE "Empleado"(

"IDEmpleado" INT NOT NULL,

"Nombre" VARCHAR(50) NULL,

"Apellido" VARCHAR(50) NULL,

"CodigoSucursal" INT NULL,

"Identificacion" VARCHAR(50) NULL,

"FechaEntrada" DATE NULL

);

ALTER TABLE

"Empleado" ADD CONSTRAINT "empleado\_idempleado\_primary" PRIMARY KEY("IDEmpleado");

CREATE TABLE "Gastos"(

"CodigoGasto" INT NOT NULL,

"EnergiaElectrica" DECIMAL(10, 2) NULL,

"Mantenimiento" DECIMAL(10, 2) NULL,

"IDGerente" INT NULL,

"Mes" VARCHAR(20) NOT NULL

);

ALTER TABLE

"Gastos" ADD CONSTRAINT "gastos\_codigogasto\_primary" PRIMARY KEY("CodigoGasto");

CREATE TABLE "ProcesoEntrada"(

"IDEmpleadoN" INT NULL,

"CodigoMaterialM" INT NULL,

"IDProcesoEntrada" BIGINT NOT NULL

);

ALTER TABLE

"ProcesoEntrada" ADD CONSTRAINT "procesoentrada\_idprocesoentrada\_primary" PRIMARY KEY("IDProcesoEntrada");

CREATE TABLE "Cambios"(

"CodigoCambio" BIGINT NOT NULL,

"CodigoMaquina" INT NULL,

"CodigoPieza" INT NULL

);

ALTER TABLE

"Cambios" ADD CONSTRAINT "cambios\_codigocambio\_primary" PRIMARY KEY("CodigoCambio");

CREATE TABLE "Chequeo"(

"CodigoChequeo" INT NOT NULL,

"ResultadoChequeo" VARCHAR(50) NULL,

"NumeroMantenimiento" INT NULL

);

ALTER TABLE

"Chequeo" ADD CONSTRAINT "chequeo\_codigochequeo\_primary" PRIMARY KEY("CodigoChequeo");

CREATE TABLE "MantenimientoMaquina"(

"CodigoMaquina" INT NOT NULL,

"Fecha" DATE NOT NULL,

"SeHizoLimpieza" BIT NOT NULL,

"NumeroMantenimiento" INT NOT NULL

);

ALTER TABLE

"MantenimientoMaquina" ADD CONSTRAINT "mantenimientomaquina\_numeromantenimiento\_primary" PRIMARY KEY("NumeroMantenimiento");

CREATE TABLE "Ganancias"(

"IDGanancia" INT NOT NULL,

"Mes" VARCHAR(20) NULL,

"Cantidad" DECIMAL(10, 2) NULL,

"IDGerente" INT NULL

);

ALTER TABLE

"Ganancias" ADD CONSTRAINT "ganancias\_idganancia\_primary" PRIMARY KEY("IDGanancia");

CREATE TABLE "MaterialEntrada"(

"CodigoMaterial" INT NOT NULL,

"EntradaTipo" VARCHAR(50) NULL,

"HorarioEntrada" TIME NULL,

"EntradaPeso" DECIMAL(10, 2) NULL,

"EntradaOrigen" VARCHAR(50) NULL

);

ALTER TABLE

"MaterialEntrada" ADD CONSTRAINT "materialentrada\_codigomaterial\_primary" PRIMARY KEY("CodigoMaterial");

CREATE TABLE "MantenimientoCambios"(

"id" BIGINT NOT NULL,

"NumeroMantenimientoN" INT NOT NULL,

"CodigoCambioM" BIGINT NOT NULL

);

ALTER TABLE

"MantenimientoCambios" ADD CONSTRAINT "mantenimientocambios\_id\_primary" PRIMARY KEY("id");

CREATE TABLE "Gerente"(

"IDGerente" INT NOT NULL,

"IDSucursal" INT NULL,

"IDEmpleado" INT NULL

);

ALTER TABLE

"Gerente" ADD CONSTRAINT "gerente\_idgerente\_primary" PRIMARY KEY("IDGerente");

CREATE TABLE "Sucursal"(

"IDSucursal" INT NOT NULL,

"Ubicacion" VARCHAR(100) NULL,

"Contacto" VARCHAR(50) NULL,

"Telefono" VARCHAR(20) NULL

);

ALTER TABLE

"Sucursal" ADD CONSTRAINT "sucursal\_idsucursal\_primary" PRIMARY KEY("IDSucursal");

ALTER TABLE

"Piezas" ADD CONSTRAINT "piezas\_idgerente\_foreign" FOREIGN KEY("IDGerente") REFERENCES "Gerente"("IDGerente");

ALTER TABLE

"ProcesoEntrada" ADD CONSTRAINT "procesoentrada\_codigomaterialm\_foreign" FOREIGN KEY("CodigoMaterialM") REFERENCES "MaterialEntrada"("CodigoMaterial");

ALTER TABLE

"Cambios" ADD CONSTRAINT "cambios\_codigomaquina\_foreign" FOREIGN KEY("CodigoMaquina") REFERENCES "Maquinas"("CodigoMaquina");

ALTER TABLE

"Piezas" ADD CONSTRAINT "piezas\_codigomaquina\_foreign" FOREIGN KEY("CodigoMaquina") REFERENCES "Maquinas"("CodigoMaquina");

ALTER TABLE

"MantenimientoCambios" ADD CONSTRAINT "mantenimientocambios\_numeromantenimienton\_foreign" FOREIGN KEY("NumeroMantenimientoN") REFERENCES "MantenimientoMaquina"("NumeroMantenimiento");

ALTER TABLE

"Maquinas" ADD CONSTRAINT "maquinas\_idsucursal\_foreign" FOREIGN KEY("IDSucursal") REFERENCES "Sucursal"("IDSucursal");

ALTER TABLE

"Gerente" ADD CONSTRAINT "gerente\_idsucursal\_foreign" FOREIGN KEY("IDSucursal") REFERENCES "Sucursal"("IDSucursal");

ALTER TABLE

"Cambios" ADD CONSTRAINT "cambios\_codigopieza\_foreign" FOREIGN KEY("CodigoPieza") REFERENCES "Piezas"("CodigoPieza");

ALTER TABLE

"Gerente" ADD CONSTRAINT "gerente\_idempleado\_foreign" FOREIGN KEY("IDEmpleado") REFERENCES "Empleado"("IDEmpleado");

ALTER TABLE

"Chequeo" ADD CONSTRAINT "chequeo\_numeromantenimiento\_foreign" FOREIGN KEY("NumeroMantenimiento") REFERENCES "MantenimientoMaquina"("NumeroMantenimiento");

ALTER TABLE

"Ganancias" ADD CONSTRAINT "ganancias\_idgerente\_foreign" FOREIGN KEY("IDGerente") REFERENCES "Gerente"("IDGerente");

ALTER TABLE

"Maquinas" ADD CONSTRAINT "maquinas\_idgerente\_foreign" FOREIGN KEY("IDGerente") REFERENCES "Gerente"("IDGerente");

ALTER TABLE

"ProcesoSalida" ADD CONSTRAINT "procesosalida\_codigomaterialm\_foreign" FOREIGN KEY("CodigoMaterialM") REFERENCES "MaterialSalida"("CodigoSalida");

ALTER TABLE

"MantenimientoCambios" ADD CONSTRAINT "mantenimientocambios\_codigocambiom\_foreign" FOREIGN KEY("CodigoCambioM") REFERENCES "Cambios"("CodigoCambio");

ALTER TABLE

"Gastos" ADD CONSTRAINT "gastos\_idgerente\_foreign" FOREIGN KEY("IDGerente") REFERENCES "Gerente"("IDGerente");

ALTER TABLE

"Piezas" ADD CONSTRAINT "piezas\_idsucursal\_foreign" FOREIGN KEY("IDSucursal") REFERENCES "Sucursal"("IDSucursal");

ALTER TABLE

"ProcesoEntrada" ADD CONSTRAINT "procesoentrada\_idempleadon\_foreign" FOREIGN KEY("IDEmpleadoN") REFERENCES "Empleado"("IDEmpleado");

ALTER TABLE

"ProcesoSalida" ADD CONSTRAINT "procesosalida\_codigomaterialn\_foreign" FOREIGN KEY("CodigoMaterialN") REFERENCES "MaterialEntrada"("CodigoMaterial");

ALTER TABLE

"Empleado" ADD CONSTRAINT "empleado\_codigosucursal\_foreign" FOREIGN KEY("CodigoSucursal") REFERENCES "Sucursal"("IDSucursal");

ALTER TABLE

"MantenimientoMaquina" ADD CONSTRAINT "mantenimientomaquina\_codigomaquina\_foreign" FOREIGN KEY("CodigoMaquina") REFERENCES "Maquinas"("CodigoMaquina");

**Principales Recomendaciones (Cómo desarrolladores) sobre el uso de DB**

**(Diseño, Vistas, funciones, resolución de problemas, etc)**

Creación y Uso de Vistas

Simplificación de Consultas:

Crear vistas para simplificar consultas complejas, facilitando el acceso a datos agregados o filtrados sin necesidad de escribir consultas SQL complicadas.

Seguridad:

Utilizar vistas para limitar el acceso a columnas sensibles de las tablas, proporcionando un control granular sobre qué datos pueden ser accedidos por diferentes usuarios.

Rendimiento:

Emplear vistas materializadas cuando sea necesario mejorar el rendimiento de consultas repetitivas y costosas.

Funciones y Procedimientos Almacenados

Modularidad y Reutilización:

Crear funciones y procedimientos almacenados para encapsular lógica de negocio reutilizable, reduciendo la duplicación de código y facilitando el mantenimiento.

Optimización del Rendimiento:

Las funciones y procedimientos almacenados se ejecutan en el servidor de base de datos, lo cual puede reducir la cantidad de datos transferidos entre el servidor y la aplicación cliente, mejorando así el rendimiento.

Seguridad:

Limitar el acceso directo a las tablas y utilizar procedimientos almacenados para implementar controles de acceso y validaciones de seguridad.

Resolución de Problemas Comunes

Optimización de Consultas:

Analizar y optimizar consultas SQL utilizando herramientas de análisis de consultas (EXPLAIN, ANALYZE) para identificar cuellos de botella y mejorar el rendimiento.

Manejo de Concurrencia:

Implementar niveles de aislamiento adecuados para evitar problemas de concurrencia como bloqueos y condiciones de carrera.

Utilizar bloqueos explícitos y estrategias de manejo de transacciones para asegurar la consistencia de los datos.

Copia de Seguridad y Recuperación:

Establecer políticas regulares de copias de seguridad y pruebas de recuperación para proteger los datos contra pérdidas y desastres.

Monitoreo y Mantenimiento:

Implementar sistemas de monitoreo para supervisar el rendimiento de la base de datos y detectar problemas de manera proactiva.

Realizar tareas de mantenimiento periódicas como la reindexación y la actualización de estadísticas para mantener el rendimiento óptimo.

**Vistas, funciones, restricciones etc…**

**sobre nuestro proyecto de base de datos**

--vistas

-- Vista para obtener detalles de materiales procesados de una sucursal en específico

use EcoMundo

go

drop view DetallesMaterialesProcesados

go

CREATE VIEW DetallesMaterialesProcesados AS

SELECT

MS.CodigoSalida,

MS.HorarioSalida,

MS.SalidaPeso,

MS.Ubicacion,

MS.SalidaTipo,

MS.Precio,

MS.Fecha,

S.IDSucursal

FROM

Sucursal S

JOIN Empleado E ON S.IDSucursal = E.CodigoSucursal

JOIN ProcesoEntrada PE on PE.IDEmpleadoN = E.IDEmpleado

JOIN MaterialEntrada ME ON PE.CodigoMaterialM = ME.CodigoMaterial

JOIN ProcesoSalida PS ON ME.CodigoMaterial = PS.CodigoMaterialN

JOIN MaterialSalida MS ON PS.CodigoMaterialN = MS.CodigoSalida

go

select \* from DetallesMaterialesProcesados WHERE

IDSucursal = 2 -- Parámetro para filtrar por sucursal específica

GROUP BY CodigoSalida, HorarioSalida, SalidaPeso, Ubicacion, SalidaTipo, Precio, Fecha, IDSucursal

go

-- Vista para obtener las ganancias y gastos por gerente

CREATE VIEW VistaFinanzasGerente AS

SELECT G.IDGerente, G.Cargo, GA.Mes, GA.Cantidad AS Ganancias, GTO.EnergiaElectrica, GTO.Mantenimiento

FROM Gerente G

JOIN Ganancias GA ON G.IDGerente = GA.Gerente\_ID

JOIN Gastos GTO ON G.IDGerente = GTO.Gerente\_ID;

--funciones

-- Función para calcular el total de ganancias por gerente

CREATE FUNCTION TotalGananciasGerente(@GerenteID INT)

RETURNS DECIMAL(10, 2)

AS

BEGIN

DECLARE @TotalGanancias DECIMAL(10, 2);

SELECT @TotalGanancias = SUM(Cantidad)

FROM Ganancias

WHERE Gerente\_ID = @GerenteID;

RETURN @TotalGanancias;

END;

-- Función para calcular el total de gastos por gerente

CREATE FUNCTION TotalGastosGerente(@GerenteID INT)

RETURNS DECIMAL(10, 2)

AS

BEGIN

DECLARE @TotalGastos DECIMAL(10, 2);

SELECT @TotalGastos = SUM(EnergiaElectrica + Mantenimiento)

FROM Gastos

WHERE Gerente\_ID = @GerenteID;

RETURN @TotalGastos;

END;

--procedimientos almacenados

-- Procedimiento para registrar una nueva entrada de material

CREATE PROCEDURE RegistrarMaterialEntrada

@CodigoMaterial INT,

@Tipo VARCHAR(50),

@HorarioEntrada TIME,

@Peso DECIMAL(10, 2),

@Origen VARCHAR(50),

@Proceso VARCHAR(50)

AS

BEGIN

INSERT INTO MaterialEntrada (CodigoMaterial, Tipo, HorarioEntrada, Peso, Origen, Proceso)

VALUES (@CodigoMaterial, @Tipo, @HorarioEntrada, @Peso, @Origen, @Proceso);

END;

-- Procedimiento para registrar una nueva salida de material

CREATE PROCEDURE RegistrarMaterialSalida

@CodigoMaterial INT,

@HorarioSalida TIME,

@SalidaPeso DECIMAL(10, 2),

@Ubicacion VARCHAR(50),

@SalidaTipo VARCHAR(50)

AS

BEGIN

INSERT INTO MaterialSalida (CodigoMaterial, HorarioSalida, SalidaPeso, Ubicacion, SalidaTipo)

VALUES (@CodigoMaterial, @HorarioSalida, @SalidaPeso, @Ubicacion, @SalidaTipo);

END;

--procedimientos almacenados

-- Procedimiento para registrar una nueva entrada de material

CREATE PROCEDURE sp\_RegistrarMaterialEntrada

@CodigoMaterial INT,

@Tipo VARCHAR(50),

@HorarioEntrada TIME,

@Peso DECIMAL(10, 2),

@Origen VARCHAR(50),

@Proceso VARCHAR(50)

AS

BEGIN

INSERT INTO MaterialEntrada (CodigoMaterial, Tipo, HorarioEntrada, Peso, Origen, Proceso)

VALUES (@CodigoMaterial, @Tipo, @HorarioEntrada, @Peso, @Origen, @Proceso);

END;

GO

-- Procedimiento para registrar una nueva salida de material

CREATE PROCEDURE sp\_RegistrarMaterialSalida

@CodigoMaterial INT,

@HorarioSalida TIME,

@SalidaPeso DECIMAL(10, 2),

@Ubicacion VARCHAR(50),

@SalidaTipo VARCHAR(50)

AS

BEGIN

INSERT INTO MaterialSalida (CodigoMaterial, HorarioSalida, SalidaPeso, Ubicacion, SalidaTipo)

VALUES (@CodigoMaterial, @HorarioSalida, @SalidaPeso, @Ubicacion, @SalidaTipo);

END;

GO

--restricciones

-- Restricción para asegurar que el peso del material de entrada sea positivo

ALTER TABLE MaterialEntrada

ADD CONSTRAINT CHK\_PesoPositivo CHECK (Peso > 0);

-- Restricción para asegurar que el peso del material de salida sea positivo

ALTER TABLE MaterialSalida

ADD CONSTRAINT CHK\_SalidaPesoPositivo CHECK (SalidaPeso > 0);

-- Restricción para asegurar que las fechas de mantenimiento sean válidas

ALTER TABLE Mantenimiento

ADD CONSTRAINT CHK\_FechaMantenimiento CHECK (JornadaMantenimiento <= GETDATE());

-- Consultar las vistas

SELECT \* FROM VistaMaterialesSucursal;

SELECT \* FROM VistaFinanzasGerente;

-- Ejecutar funciones

SELECT dbo.TotalGananciasGerente(1) AS TotalGanancias;

SELECT dbo.TotalGastosGerente(1) AS TotalGastos;

-- Ejecutar procedimientos almacenados

EXEC RegistrarMaterialEntrada 2, 'Metal', '09:00:00', 200.00, 'Industrial', 'Procesado';

EXEC RegistrarMaterialSalida 2, '13:00:00', 195.00, 'Centro de Reciclaje', 'Metal';

**Conclusión General**

El desarrollo de un sistema de bases de datos para la gestión de reciclaje de basura y la producción de materiales comerciales presenta múltiples beneficios económicos, ambientales y sociales. La implementación de un diseño de base de datos eficiente, junto con el uso adecuado de vistas, funciones y procedimientos almacenados, es crucial para el éxito del proyecto. La aplicación de las mejores prácticas en el diseño, la optimización del rendimiento, y la resolución de problemas comunes, garantizará la integridad, seguridad y escalabilidad del sistema. Esto no solo facilitará la gestión interna de la compañía, sino que también contribuirá significativamente a la sostenibilidad y la economía circular, apoyando los objetivos de reducción de residuos y aprovechamiento sostenible de los recursos.

**Referencias**

Elmasri, R., & Navathe, S. B. (2015). Fundamentals of Database Systems. Pearson.

Silberschatz, A., Korth, H. F., & Sudarshan, S. (2019). Database System Concepts. McGraw-Hill Education.

Connolly, T., & Begg, C. (2014). Database Systems: A Practical Approach to Design, Implementation, and Management. Pearson.

Date, C. J. (2015). An Introduction to Database Systems. Addison-Wesley.

Coronel, C., Morris, S., & Rob, P. (2016). Database Systems: Design, Implementation, & Management. Cengage Learning.

Hoffer, J. A., Ramesh, V., & Topi, H. (2016). Modern Database Management. Pearson.

Garcia-Molina, H., Ullman, J. D., & Widom, J. (2008). Database Systems: The Complete Book. Prentice Hall.