

PROYECTO FINAL

COMISION 50070 SQL

BRUCE VALERIA ANDREA
| DNI 37348054

Introducción

Un buen sistema de gestión es esencial para cualquier empresa ya que permite mejorar la eficiencia operativa, garantizar el cumplimiento de procedimientos y normas internas, mejorar la experiencia del cliente y proporcionar información valiosa para la toma de decisiones.

Modelo de Negocio

Compañía de Servicios Financieros

Es una entidad que proporciona servicios financieros a individuos, ofreciendo la posibilidad de obtener préstamo personal, prendario o hipotecario.

Atención al Cliente: La oficina de préstamos generalmente cuenta con personal capacitado para atender a los clientes. Estos profesionales pueden proporcionar información sobre los diferentes tipos de préstamos disponibles, los requisitos para la solicitud y los términos y condiciones asociados.

Proceso de Solicitud: Las oficinas de préstamos suelen contar con un proceso de solicitud de préstamo que incluye la presentación de documentos financieros, información personal y otros detalles relevantes. El personal puede guiar a los solicitantes a través de este proceso.

Evaluación de Riesgos: La entidad realiza una evaluación del riesgo crediticio para determinar la viabilidad de otorgar un préstamo. Esto implica revisar el historial crediticio, ingresos, empleo y otras variables relevantes.

Tasas de Interés y Condiciones: La entidad proporciona información clara sobre las tasas de interés asociadas con los préstamos y otras condiciones importantes, como cargos por pagos atrasados o penalizaciones.

Objetivo:

Brindar soporte operativo para la gestión y permitir la evaluación de resultados a través de informes.

Situación problemática

La organización de la información.

El control y supervisión de permisos de aprobaciones

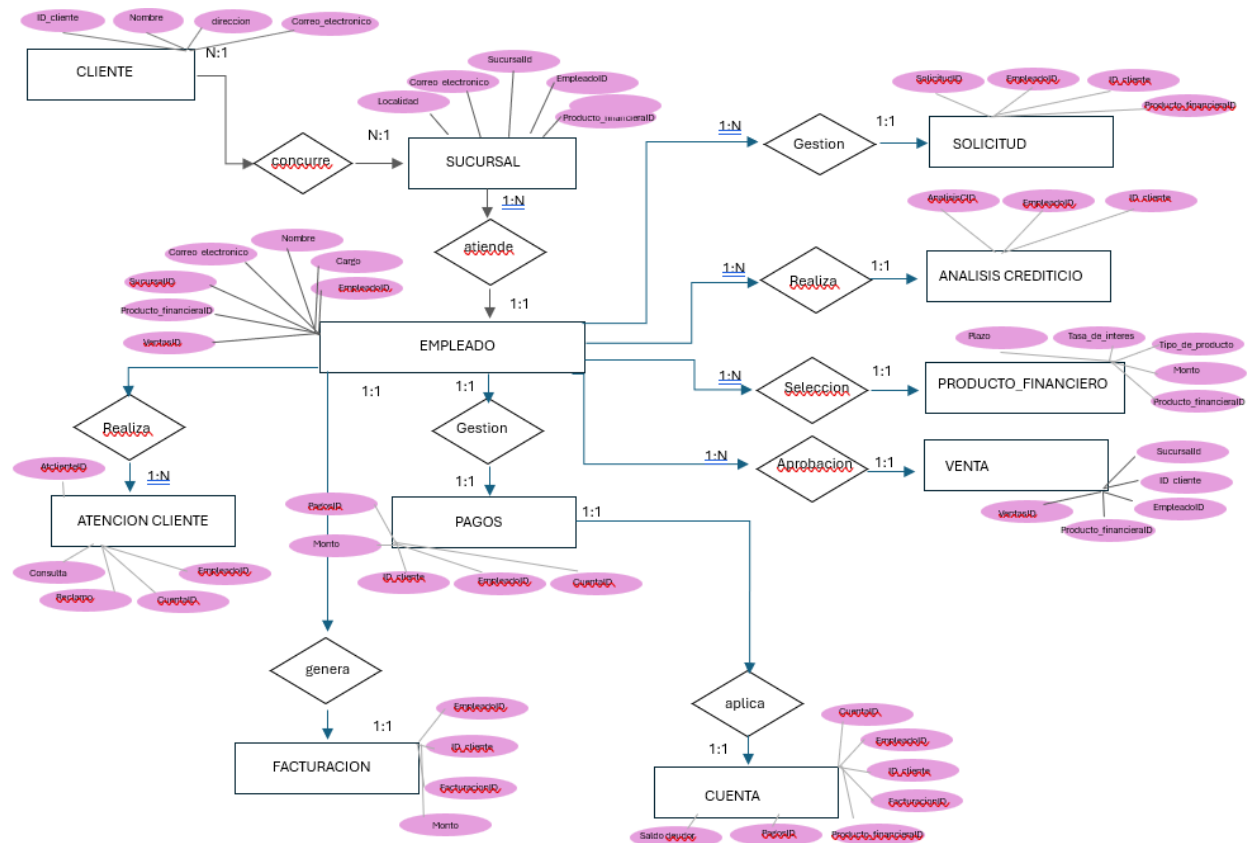
Registro de gestiones operativas

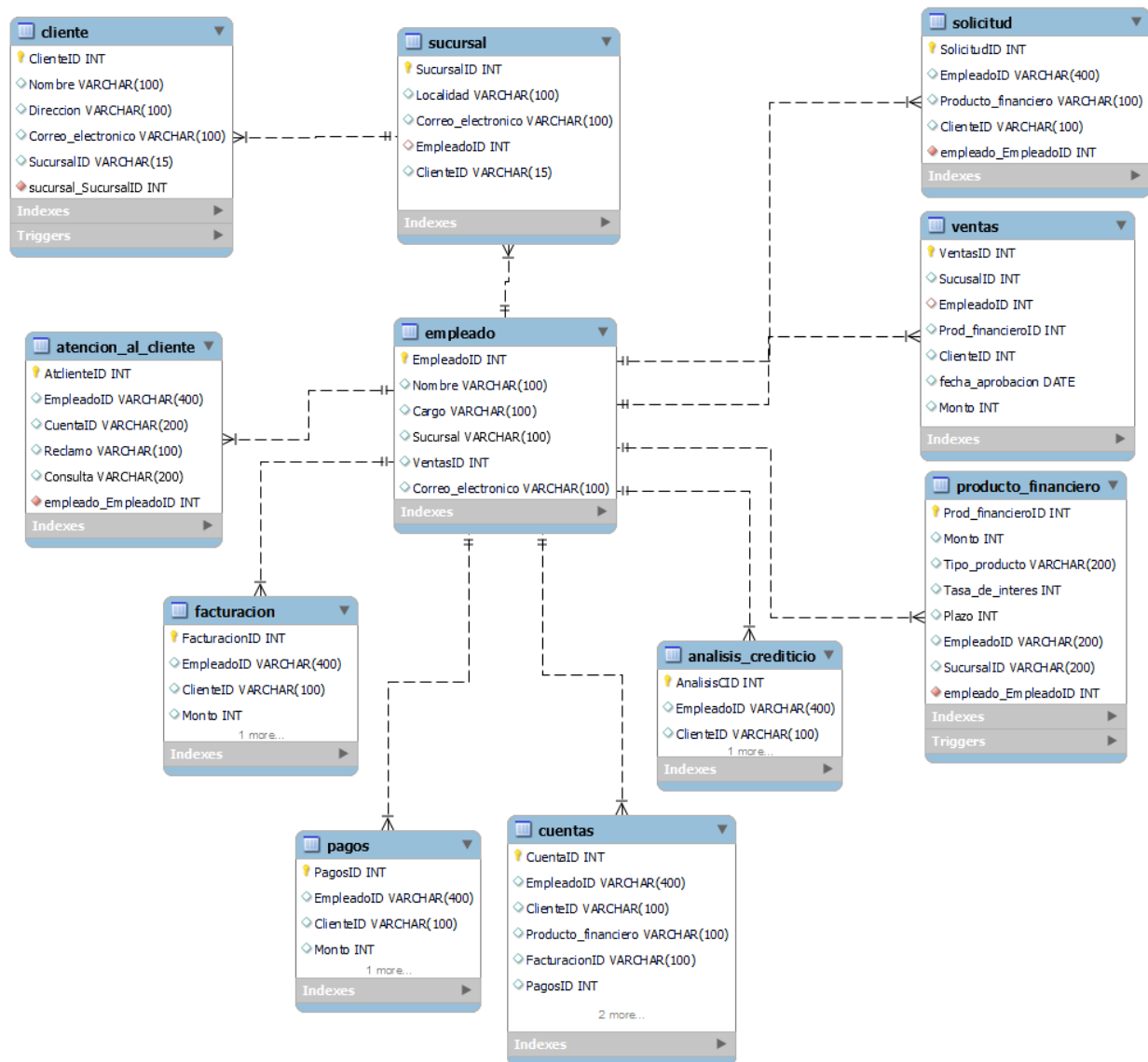
Descripción de tablas:

	Name	Description	field	PRIMARY KEY	FOREIGN KEY	Date Type
1	Clientes	Persona que se acerca a la oficina a solicitar un producto financiero	ClienteID	X		
			Nombre			Integer
			Dirección			Integer
			Correo electrónico			Integer
2	Producto Financiero	Servicio financiero que ofrece la compañía	Prod_financieroID	X		
			Monto			Integer
			Tipo_de_producto			Varchar
			Tasa_de_interés			Integer
			Plazo			Integer
			EmpleadoID		X	INT
			SucursalID		X	INT
			CienteID		X	INT
3	Empleado	Persona contratada para brindar soporte de venta al servicio de la compañía	EmpleadoID	X		Varchar
			Nombre			Varchar
			Cargo			Varchar
			Sucursal			Varchar
			Correo_electronico			Varchar
			Producto_financieroID		X	INT
			VentasID		X	INT
4	Sucursal	Oficina de venta de la compañía ubicada en otra localidad	SucursalID	X		Varchar
			Localidad			Varchar
			Correo_electronico			Varchar
			EmpleadoID		x	INT
			Producto_financieroID		x	INT
5	Ventas	Monto de ventas	VentasID	X		
			SucursalID		X	Varchar
			EmpleadoID		X	INT
			Producto_financieroID		X	INT
			CienteID		X	INT
6	Solicitud	El trámite que se inicia para acceder a un producto	Solicitud ID	X		
			EmpleadoID		x	INT
			Producto_financieroID		x	INT
			CienteID		X	INT
7	Análisis Crediticio	Es el proceso de evaluación de variable crediticias que cumplen con la condición para aprobar la solicitud	AnálisisCID	X	X	INT
			EmpleadoID		X	INT
			CienteID		X	INT
8	Facturación	Proceso de generación de factura al cliente	FacturaciónID	X		INT
			EmpleadoID		X	INT
			CienteID		X	INT
			Monto		X	INT
9	Cuentas	Generación de cuenta corriente del cliente	CuentaID	X		
			EmpleadoID		X	INT
			CienteID		X	INT
			FacturaciónID		X	INT
			PagosID		X	INT

	Name	Description	field	PRIMARY KEY	FOREIGN KEY	Date Type
			Producto_financieroID		X	INT
			Saldo Deudor		X	INT
10	Pagos	Montos abonados por el cliente relativos a la facturación del producto adquirido	PagoID	X		
			EmpleadoID		X	INT
			ClienteID		X	INT
			Monto		X	INT
11	Atencion al cliente	Generación de consultas del cliente	AtclienteID	X		
			EmpleadoID		X	INT
			CuentaID		X	INT
			Reclamo		X	INT
			Consultas		X	INT

DIAGRAMA ENTIDAD RELACION





Script de inserción de Datos

```
INSERT INTO cliente (ClienteID, Nombre, Direccion, Correo_electronico)
VALUES
```

```
(1, 'Juan Pérez', 'Juan B. Justo 12', 'juan@email.com'),
(2, 'María López', 'Corrientes 234', 'maria@email.com'),
(4, 'Carlos Ramírez', 'Boer 45', 'carlos@email.com');
```

```
INSERT INTO cliente (ClienteID, Nombre, Direccion, Correo_electronico)
VALUES
```

```
(5, 'Pedro García', 'Calle 7', 'pedro@email.com'),
(6, 'Ana Martínez', 'Paseo Colón 890', 'ana@email.com'),
(7, 'Carlos Sánchez', 'Apartado 1234', 'carlos@email.com'),
(8, 'Eduardo González & Asociados', 'Av. Libertador 2000', 'eduardo@email.com');
```

```
INSERT INTO producto_financiero (Prod_financieroID, Monto, Tipo_producto, tasa_de_interes, plazo, EmpleadoID,
SucursalID, ClienteID)
```

```
VALUES
```

```
(101, '2000000', 'Credito personal', '95', 24, 301, 201, 1),
(102, '3000000', 'Credito personal', '95', 24, 302, 202, 2),
(103, '60000000', 'Credito hipotecario', '95', 100, 303, 203, 3);
```

```
INSERT INTO producto_financiero (Prod_financieroID, Monto, Tipo_producto, tasa_de_interes, plazo, EmpleadoID,
SucursalID, ClienteID)
```

```
VALUES
```

```
(104, '60000000', 'Credito hipotecario', '95', 150, 303, 203, 5),
(105, '60000000', 'Credito hipotecario', '110', 150, 301, 201, 6),
(106, '60000000', 'Credito personal', '95', 18, 303, 203, 7),
(107, '60000000', 'Credito personal', '110', 12, 302, 202, 8);
```

```
INSERT INTO Sucursal (SucursalID, localidad, Correo_electronico, EmpleadoID, ClienteID)
```

```
VALUES
```

```
(201, 'Cipolletti', 'cipolletti@serviciosfinan.com', 301, 6),
(202, 'Plottier', 'plottier@serviciosfinan.com', 302, 8),
(203, 'Neuquen', 'neuquen@serviciosfinan.com', 303, 7),
(203, 'Neuquen', 'neuquen@serviciosfinan.com', 303, 5);
```

```
INSERT INTO empleado (EmpleadoID, nombre, cargo, sucursal, Prod_financieroID, VentasID, Correo_electronico)
VALUES
```

```
(301, 'J.Julian', 'vendedor', 'Cipolletti', 101, 42, 'cipolletti@serviciosfinan.com'),
(302, 'A.Salinas', 'vendedor', 'Plottier', 102, 43, 'plottier@serviciosfinan.com'),
(303, 'B.Díaz', 'Gerente de ventas', 'Neuquen', 103, 56, 'neuquen@serviciosfinan.com');
```

```
INSERT INTO ventas (VentasID, SucursalID, EmpleadoID, Prod_financieroID, ClienteID)
VALUES
```

```
(42, 201, 301, 101, 1),
(43, 202, 302, 102, 2),
(56, 203, 303, 103, 3);
```

```
INSERT INTO ventas (VentasID, SucursalID, EmpleadoID, Prod_financieroID, ClienteID)
VALUES
```

```
(12, 203, 303, 104, 5),
(7, 201, 301, 105, 6),
(44, 203, 303, 106, 7),
```

```

(05, 202, 303, 107, 8);

INSERT INTO solicitud (SolicitudID, EmpleadoID, Producto_financiero, ClienteID)
VALUES
    (1, 301, 'Credito personal', 102),
    (2, 302, 'Credito personal', 200),
    (3, 303, 'Credito personal', 104);

INSERT INTO solicitud (SolicitudID, EmpleadoID, Producto_financiero, ClienteID)
VALUES
    (4, 303, 'Credito hipotecario', 5),
    (5, 301, 'Credito hipotecario', 6),
    (6, 303, 'Credito personal', 7),
    (7, 302, 'Credito personal', 8);

INSERT INTO analisis_credificio (AnalisisCID, EmpleadoID, ClienteID)
VALUES
    (05, 301, 102),
    (07, 302, 200),
    (04, 303, 104);

INSERT INTO facturacion (FacturacionID, EmpleadoID, ClienteID, Monto)
VALUES
    (1020, 301, 102, 250000),
    (1030, 302, 200, 650000),
    (1040, 303, 104, 124000);

INSERT INTO facturacion (FacturacionID, EmpleadoID, ClienteID, Monto)
VALUES
    (1050, 303, 5, 450000),
    (1060, 301, 6, 350000),
    (1070, 303, 7, 80000),
    (1080, 302, 8, 20000);

INSERT INTO cuentas (CuentaID, EmpleadoID, ClienteID, producto_financiero, facturacionID, PagosID,
Saldo_deudor)
VALUES
    (102650, 301, 102, 'Credito personal', 150, 203, 1020000),
    (35245, 302, 200, 'Credito personal', 120, 402, 1400000),
    (145552, 303, 104, 'Credito personal', 565, 405, 965000);

INSERT INTO cuentas (CuentaID, EmpleadoID, ClienteID, producto_financiero, facturacionID, PagosID,
Saldo_deudor)
VALUES
    (2563, 303, 5, 'Credito hipotecario', 1050, 204, 45000000),
    (10359, 301, 6, 'Credito hipotecario', 1060, 205, 30000000),
    (23589, 303, 7, 'Credito personal', 1070, 206, 4000000),
    (45789, 302, 8, 'Credito personal', 1080, 207, 2500000);

INSERT INTO pagos (PagosID, EmpleadoID, clienteID, Monto)
VALUES
    (203, 301, 102, 145265.45),
    (402, 302, 200, 289565.10),
    (405, 303, 104, 85960.12);

INSERT INTO pagos (PagosID, EmpleadoID, clienteID, Monto)
VALUES
    (407, 301, 5, 450000),
    (520, 302, 6, 350000),
    (605, 303, 7, 80000),
    (704, 302, 8, 20000);

```

```
INSERT INTO atencion_al_cliente (AtclienteID, EmpleadoID, CuentaID, Reclamo, Consulta)
VALUES
(785, 301, 45877, 'Reclamo', NULL),
(415, 302, 105695, NULL, 'Consulta'),
(898, 303, 10025, NULL, 'Consulta');
```

Funcionalidades

VISTAS

Vista 1: **Vista_Cliente_ProductoFinanciero:**

Descripción: Esta vista mostrará la información de los clientes junto con los productos financieros que han adquirido.

Tablas que la componen: Cliente, Producto_financiero.

Vista 2: **Vista_Sucursal_ProductoFinanciero:**

Descripción: Esta vista proporcionará un resumen de las ventas mensuales agrupadas por sucursal.

Tablas que la componen: Sucursal, Producto_financiero

Vista 3: **Vista_Ventas_Empleados:**

Descripción: Esta vista mostrará los empleados junto con el total de ventas realizadas por cada uno de ellos.

Tablas que la componen: Ventas, Sucursal, Empleado, Producto_financiero

Vista 4: **Vista_Ventas_Detalles:**

Descripción: Esta vista mostrara el detalle de las ventas efectuadas por los clientes.

Tablas que la componen: Ventas, Sucursal, Empleado, Producto_financiero, Cliente

Vista 5: **Vista_ProductosMasVendidos:**

Descripción: Esta vista mostrará los productos financieros con mayor cantidad de ventas.

Tablas que la componen: Ventas, , Producto_financiero,

FUNCIONES

Función 1: Obtener Total de Ventas por Empleado: Esta función devuelve el total de ventas realizadas por un empleado específico.

```
DELIMITER //
CREATE FUNCTION TotalVentasEmpleado (empleado_id INT) RETURNS INT
BEGIN
    DECLARE total_ventas INT;
    SELECT SUM(pf.Monto) INTO total_ventas
    FROM Producto_financiero pf
    JOIN Ventas v ON pf.Prod_financieroID = v.Prod_financieroID
    WHERE v.EmpleadoID = empleado_id;
    RETURN total_ventas;
END //
DELIMITER ;
```

Función 2: Plazo promedio de productos financieros: Esta función determinara cual es el plazo promedio elegido por el cliente para realizar su operación.

```
DELIMITER //
CREATE FUNCTION PlazoPromedioPorTipo(tipo_producto VARCHAR(200)) RETURNS DECIMAL(10,2)
BEGIN
    DECLARE plazo_promedio DECIMAL(10,2);
    SELECT AVG(Plazo) INTO plazo_promedio
    FROM Producto_financiero
    WHERE Tipo_producto = tipo_producto;
    RETURN plazo_promedio;
END //
DELIMITER ;
```

Función 3: Monto otorgado en productos financiero más alto: Esta función determinara cual es el monto mas alto vendido entre los tipos de productos financieros

```
DELIMITER //
CREATE FUNCTION ClienteConMontoMasAlto() RETURNS VARCHAR(100)
BEGIN
    DECLARE cliente_nombre VARCHAR(100);
    SELECT c.Nombre INTO cliente_nombre
    FROM Cliente c
    JOIN Producto_financiero pf ON c.ClienteID = pf.ClienteID
    ORDER BY pf.Monto DESC
    LIMIT 1;
    RETURN cliente_nombre;
END //
DELIMITER ;
```

STORED PROCEDURE

1)Procedimiento para Insertar Nuevo Producto Financiero:

Descripción: Este procedimiento permite insertar un nuevo producto financiero en la base de datos.

Objetivo: Facilitar la creación de nuevos productos financieros.

Tablas utilizadas: Producto_financiero

```
DELIMITER //
CREATE PROCEDURE InsertarNuevoProductoFinanciero(
    IN p_Monto INT,
    IN p_TipoProducto VARCHAR(200),
    IN p_TasaInteres INT,
```

```

    IN p_Plazo INT,
    IN p_EmpleadoID VARCHAR(200),
    IN p_SucursalID VARCHAR(200),
    IN p_ClienteID INT
)
BEGIN
    INSERT INTO Producto_financiero (Monto, Tipo_producto, Tasa_de_interes, Plazo, EmpleadoID, SucursalID, ClienteID)
    VALUES (p_Monto, p_TipoProducto, p_TasaInteres, p_Plazo, p_EmpleadoID, p_SucursalID, p_ClienteID);
END //
DELIMITER ;

```

2)Procedimiento para Actualizar Información del Empleado:

Descripción: Este procedimiento actualiza la información de un empleado.

Objetivo: Mantener actualizada la información de los empleados.

Tablas utilizadas: Empleado

```

DELIMITER //
CREATE PROCEDURE ActualizarInformacionEmpleado(
    IN p_EmpleadoID INT,
    IN p_NuevoNombre VARCHAR(100),
    IN p_NuevoCargo VARCHAR(100),
    IN p_NuevaSucursal VARCHAR(100),
    IN p_NuevoCorreo VARCHAR(100)
)
BEGIN
    UPDATE Empleado
    SET Nombre = p_NuevoNombre, Cargo = p_NuevoCargo, Sucursal = p_NuevaSucursal, Correo_electronico = p_NuevoCorreo
    WHERE EmpleadoID = p_EmpleadoID;
END //
DELIMITER ;

```

3)Procedimiento para Calcular Total de Ventas por Sucursal:

Descripción: Este procedimiento calcula el total de ventas para una sucursal específica.

Objetivo: Proporcionar información sobre el rendimiento de cada sucursal.

Tablas utilizadas: Producto_financiero, Ventas.

```

DELIMITER //
CREATE PROCEDURE CalcularTotalVentasPorSucursal(
    IN p_SucursalID INT,
    OUT p_TotalVentas INT
)
BEGIN
    SELECT SUM(pf.Monto) INTO p_TotalVentas
    FROM Producto_financiero pf
    JOIN Ventas v ON pf.Prod_financieroID = v.Prod_financieroID
    WHERE pf.SucursalID = p_SucursalID;
END //
DELIMITER ;

```

4)Procedimiento para Eliminar Producto Financiero y Ventas Asociadas:

Descripción: Este procedimiento elimina un producto financiero y las ventas asociadas a ese producto.

Objetivo: Permitir la eliminación ordenada de productos financieros.

Tablas utilizadas: Producto_financiero, Ventas.

```

DELIMITER //

```

```

CREATE PROCEDURE EliminarProductoFinancieroYVentas(
    IN p_ProdFinancieroID INT
)
BEGIN
    DELETE FROM Ventas WHERE Prod_financieroID = p_ProdFinancieroID;
    DELETE FROM Producto_financiero WHERE Prod_financieroID = p_ProdFinancieroID;
END //
DELIMITER ;

```

TRIGGERS

1) Trigger para Actualizar Total de Ventas al Insertar una Venta:

Descripción: Este trigger actualiza automáticamente el total de ventas en la tabla de Sucursal cada vez que se inserta una nueva venta.

Objetivo: Mantener actualizado el total de ventas por sucursal en tiempo real.

```

DELIMITER //
CREATE TRIGGER ActualizarTotalVentasDespuesDeInsertar
AFTER INSERT ON Ventas
FOR EACH ROW
BEGIN
    UPDATE Sucursal s
    SET TotalVentas = TotalVentas + NEW.Monto
    WHERE s.SucursalID = NEW.SucursalID;
END //
DELIMITER ;

```

2) Trigger para Eliminar Ventas Asociadas al Eliminar un Cliente:

Descripción: Este trigger elimina automáticamente todas las ventas asociadas a un cliente cuando se elimina dicho cliente.

Objetivo: Mantener la integridad referencial y eliminar datos relacionados cuando sea necesario

```

DELIMITER //
CREATE TRIGGER EliminarVentasDespuesDeEliminarCliente
BEFORE DELETE ON Cliente
FOR EACH ROW
BEGIN
    DELETE FROM Ventas WHERE ClienteID = OLD.ClienteID;
END //
DELIMITER ;

```

3) Trigger para Actualizar Ventas Después de Modificar Monto de Producto Financiero:

Descripción: Este trigger se ejecutará después de actualizar el monto de un producto financiero y actualizará automáticamente las ventas asociadas a ese producto.

Objetivo: Reflejar cambios en las ventas cuando se actualizan montos de productos financieros

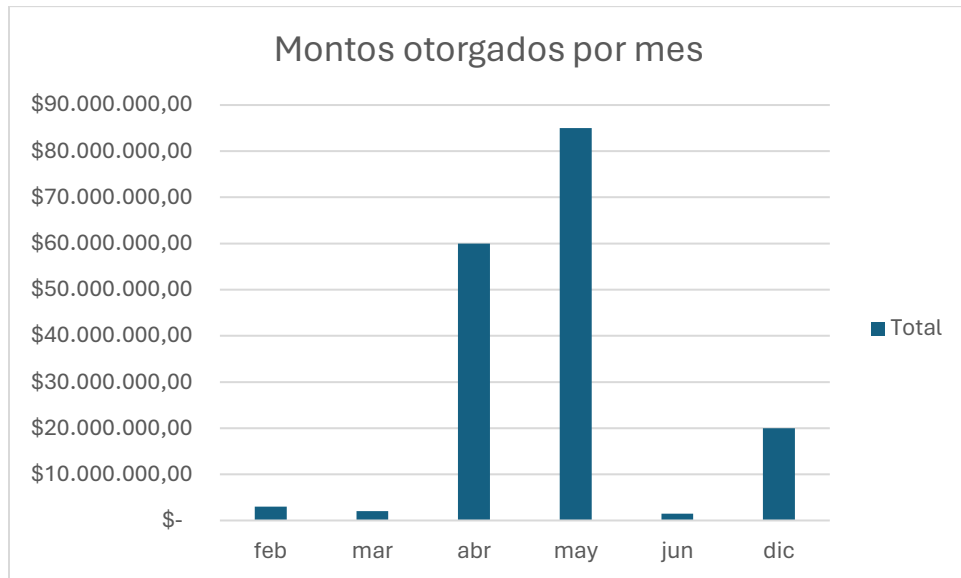
```

DELIMITER //
CREATE TRIGGER ActualizarVentasDespuesDeActualizarMonto
AFTER UPDATE ON Producto_financiero
FOR EACH ROW
BEGIN
    UPDATE Ventas v
    SET v.Monto = NEW.Monto
    WHERE v.Prod_financieroID = NEW.Prod_financieroID;
END //
DELIMITER ;

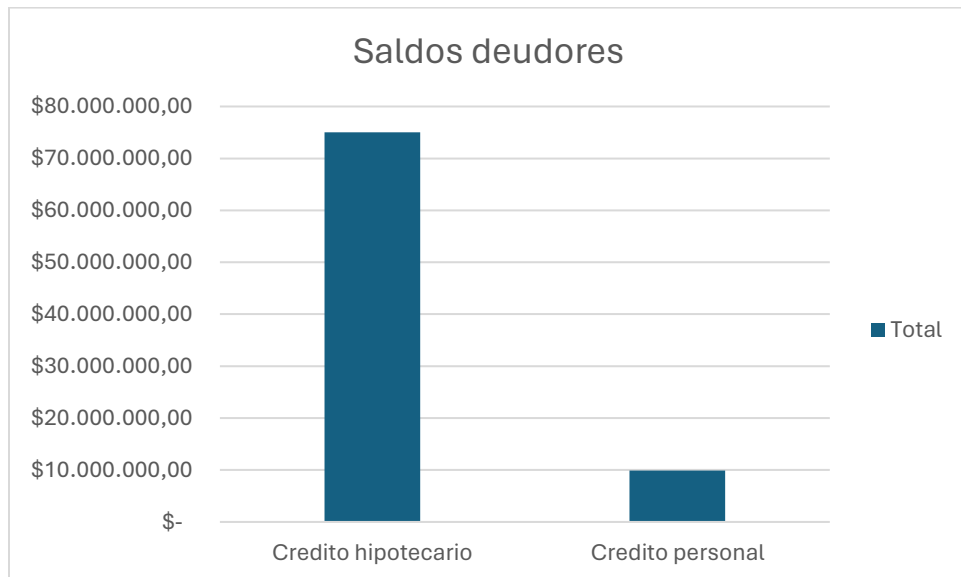
```

Informes generados en base a la información de la BD

1) Montos máximos de crédito entregados por mes



2) Saldo deudor por Tipo de Producto



Herramientas y tecnologías utilizadas

- MySQL Workbench
- Microsoft Excel
- Microsoft Word

Futuras líneas

Para que la base de datos de la compañía de servicios financieros pueda seguir creciendo y adaptándose a las necesidades futuras, es importante considerar varios aspectos clave:

Diseño escalable de la base de datos: A medida que la compañía crezca y añada más servicios o productos financieros, es crucial tener un diseño de base de datos escalable que pueda manejar el crecimiento sin comprometer el rendimiento. Esto implica utilizar buenas prácticas de diseño como la normalización de datos, índices eficientes y estructuras flexibles para adaptarse a nuevos tipos de productos financieros.

Nuevos tipos de productos financieros: A medida que la compañía diversifique su oferta de servicios financieros, la base de datos debe ser capaz de manejar la información asociada con estos nuevos productos. Por ejemplo, si se introducen préstamos prendarios o tarjetas de crédito, se deben añadir nuevas tablas o campos a la base de datos para almacenar la información relevante de estos productos.

Mejora continua de la experiencia del cliente: La base de datos puede ser utilizada para recopilar datos sobre la interacción de los clientes con la compañía, lo que puede ayudar a mejorar continuamente la experiencia del cliente. Por ejemplo, se pueden analizar patrones de comportamiento de los clientes para personalizar ofertas o servicios, o para identificar áreas de mejora en los procesos de solicitud y atención al cliente.

Seguridad de la información: Con el crecimiento de la base de datos y la cantidad de datos confidenciales de los clientes, es fundamental implementar medidas sólidas de seguridad de la información. Esto incluye la encriptación de datos sensibles, el control de acceso basado en roles y auditorías de seguridad regulares.