**Proyecto: Billetera Virtual**

**Entrega N° 2**

**Martiri Pineda Nahuel**

**Introducción**

El proyecto se basa en el diseño de una base de datos para una billetera virtual. Este sistema permitirá gestionar cuentas de clientes y registrar sus transacciones.

Se busca crear una solución que facilite la consulta, modificación y análisis de los datos asociados a las transacciones y saldos de los usuarios.

El sistema será escalable y adaptable a diferentes necesidades que el negocio presente, garantizando su eficiencia y fiabilidad en tiempo real.

**Objetivo**

Este proyecto tiene como objetivo desarrollar una base de datos que permita gestionar la información relacionada con sus clientes, cuentas y transacciones.

áreas funcionales:

* gestión de clientes
* gestión de cuentas
* gestión de transacciones

**Situación Problemática**

A partir de la pandemia los clientes que manejan transacciones electrónicas enfrentan una creciente necesidad de contar con sistemas eficientes y simple que les permita realizar sus operaciones diarias en tiempo real.

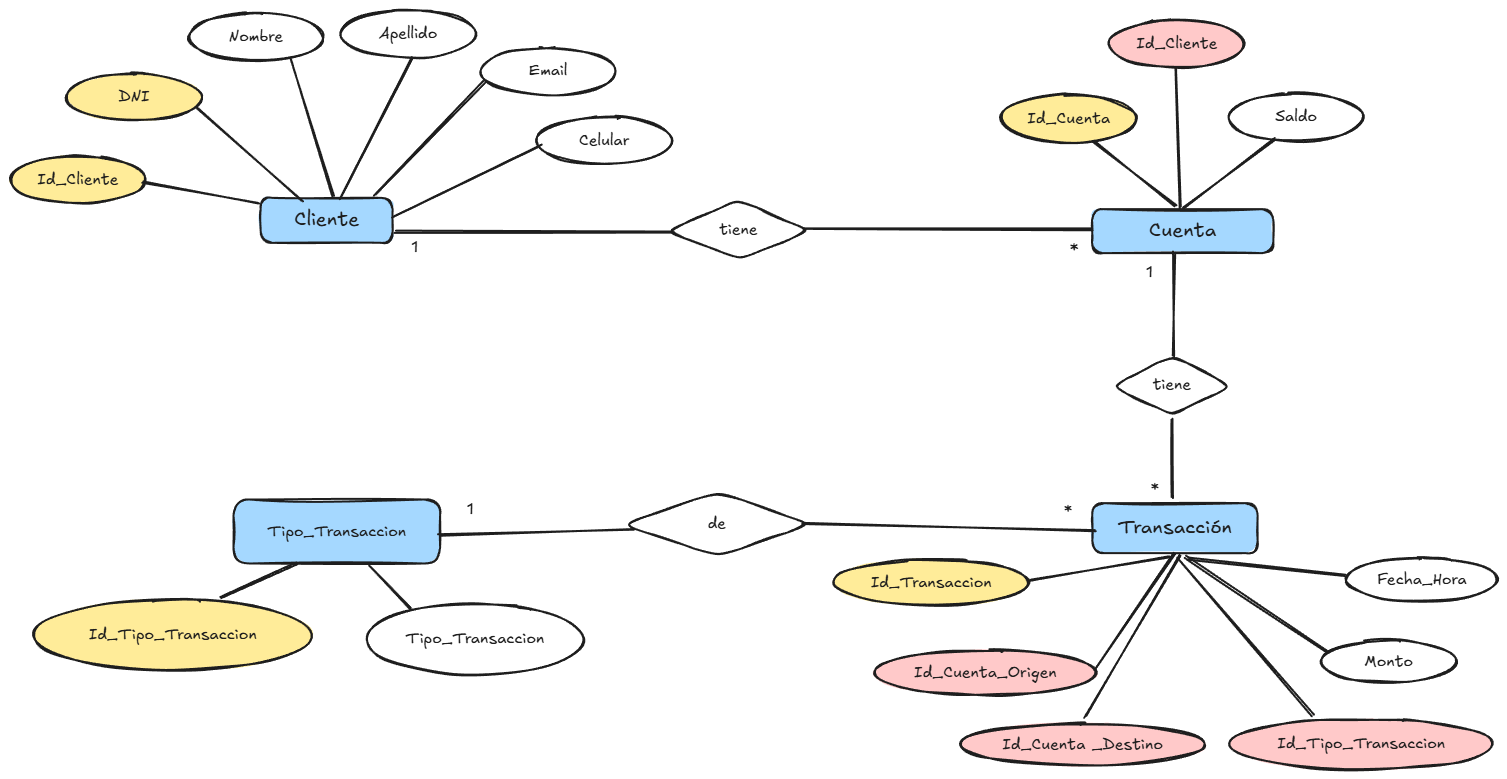
Este proyecto busca cubrir esta necesidad implementado una base de dato robusta que permita centralizar y organizar la información, garantizando la fiabilidad de las transacciones, además de facilitar la visualización de los datos tanto para los clientes como para la empresa.

**Modelo de Negocio**

Esta propuesta esta dirigida a empresas que ofrecen servicios de billeteras virtuales.

Estas organizaciones se dedican al manejo de fondos electrónicos que abarcan múltiples usuarios (clientes) los cuales realizan transacciones diariamente. La solución propuesta permitirá que los usuarios realicen transferencias, depósitos, retiros y consultas de saldos de forma rápida y segura.

**DER**

****

**Tablas**



**Base de datos SQL**

**Datos ingresados**









**Vistas**

1. Vista 1 - para ver saldos y números de cuentas, pero ningún otro dato

CREATE VIEW Saldos\_Actuales\_por\_Cuenta

AS SELECT Id\_cuenta, Saldo

FROM Cuenta;

SELECT \* FROM Saldos\_Actuales\_por\_Cuenta;

1. Vista 2 - para ver qué tipo de transacciones existen

Create VIEW Transacciones\_Existentes

AS SELECT Tipo\_Transaccion

FROM tipo\_de\_transaccion;

Select \* FROM Transacciones\_Existentes;

1. Vista 3 - para ver transacciones realizadas mayores a $2000 con su cuenta origen

CREATE VIEW Transacciones\_Mayores\_de\_2000

AS SELECT Id\_Cuenta\_Origen, Monto

FROM transaccion

WHERE Monto>2000;

Select \* FROM Transacciones\_mayores\_de\_2000;

1. Vista 4 - para ver todo cliente que haya recibido una transferencia

CREATE VIEW Cliente\_Recibio\_Tranferencia

AS

SELECT Cliente.Apellido, Cliente.Nombre, Cliente.Email

FROM Transaccion

JOIN Cuenta ON Transaccion.Id\_Cuenta\_Destino = Cuenta.Id\_Cuenta

JOIN Cliente ON Cuenta.Id\_Cliente = Cliente.Id\_Cliente

WHERE transaccion.Tipo\_transaccion = 1;

Select \* From Cliente\_Recibio\_Tranferencia;

1. Vista 5 -para ver clientes con DNI menores a 30 Millones

Create view Clientes\_DNI\_Menor\_30Millones

AS

Select cliente.Id\_Cliente, cliente.Apellido, cliente.Nombre

From cliente

Where cliente.DNI< 30000000;

Select \* From Clientes\_DNI\_Menor\_30Millones;

**Funciones**

1. Función que sume el total de lo tranferido por cuenta.

DELIMITER //

CREATE FUNCTION CalcularTotalTransferido (idcuenta INT)

RETURNS DECIMAL (10,2)

DETERMINISTIC

BEGIN

DECLARE total DECIMAL (10,2);

Select SUM(Monto) INTO total

FROM transaccion

where Id\_Cuenta\_Origen = idCuenta;

If total IS NULL THEN

SET total = 0;

END IF;

RETURN total;

END //

DELIMITER ;

2. Función que muestre la primer transferencia realizada

CREATE FUNCTION Primeratransferencia (idcuenta INT)

RETURNS DATETIME

DETERMINISTIC

BEGIN

DECLARE fecha\_primer DATETIME;

select MIN(Fecha\_Hora) INTO Fecha\_primer

FROM transaccion

WHERE Id\_Cuenta\_Origen = idcuenta;

return Fecha\_primer;

end //

DELIMITER ;

**Base de datos SQL**

<https://github.com/nmartiri/Coder>

<https://drive.google.com/drive/folders/1exLMhqsqXvFmGxA7HWkQ3k-y0rNgkuAy?usp=sharing>