

Diseño de Base de Datos – Sistema de Cajero Automático

1. Descripción general

El diseño de la base de datos bdcajero tiene como objetivo respaldar la operación de un sistema de cajero automático con funcionalidades para gestión de usuarios, cuentas bancarias, préstamos, transacciones, pagos de servicios y remesas. El modelo está diseñado con un enfoque relacional entre las tablas, aplicando referencias integrales a través de claves primarias y foráneas, y permitiendo la escalabilidad del sistema a través de un diseño modular y normalizado.

2. Tablas principales

usuarios

Contiene la información de todas las personas registradas (clientes o bancos).

- **id_usuario** (PK)
- nombre
- tipo_usuario (cliente / banco)
- usuario
- password_hash
- correo
- cantidad_prestamos
- Activo

cuentas

Almacena los datos de las cuentas bancarias pertenecientes a los usuarios.

- **id_cuenta** (PK)
- numero_cuenta

- tipo (ahorro / corriente)
- banco (nombre en texto)
- id_usuario (FK -> usuarios)
- saldo

transacciones

Registra todos los movimientos realizados en las cuentas.

- **id_transaccion** (PK)
- id_cuenta (FK -> cuentas)
- tipo_transaccion (ej. depósito, retiro)
- monto
- fecha

pagos_servicios

Contiene los pagos realizados por los usuarios a servicios básicos.

- **id_pago** (PK)
- tipo_servicio (agua, luz, teléfono, etc.)
- monto
- fecha_pago
- id_usuario (FK-> usuarios)

prestamos

Registra préstamos otorgados a los usuarios, con montos e historial.

- **id_prestamo** (PK)
- id_usuario (FK -> usuarios)
- monto_total
- saldo_pendiente
- fecha_inicio

remesas

Permite el envío y retiro de remesas entre usuarios.

- **id_remesa** (PK)
- codigo (único)
- monto
- estado (generada / retirada)
- fecha_envio
- id_remitente (FK-> usuarios)
- id_receptor (FK-> usuarios)

3. Relaciones clave

- Un usuario puede tener **una o varias cuentas bancarias**.
- Un usuario puede tener **varios préstamos, pagos de servicios y remesas**.
- Una cuenta puede tener **muchas transacciones**.
- Una remesa siempre relaciona a **dos usuarios diferentes**: remitente y receptor.

4. Consideraciones de diseño

- Se usa texto (VARCHAR) para almacenar nombres de bancos, lo que permite una mayor flexibilidad sin depender de enumeraciones rígidas.
- Se mantiene un campo activo en la tabla usuarios para permitir la activación/desactivación lógica sin eliminar registros.
- Se incorpora password_hash con SHA-256 para asegurar la confidencialidad del PIN del usuario.

5. Ventajas del diseño

- Permite escalabilidad y modularidad (puedes agregar más funcionalidades como historial de préstamos o pagos programados).
- Garantiza integridad de datos mediante claves foráneas.
- Reduce la redundancia de datos mediante relaciones bien definidas.