

# Normalización:

## Dependencias

- Usuario(tipoDocumento, numeroDocumento, Rol[FK(Rol.tipo)], login, contraseña, nombre, nacionalidad, dirección, email, teléfono, ciudad, departamento, codigoPostal, Oficina[FK(Oficina.id)])
- Cliente(IdCliente)  
  {(IdCliente) → ()}  
  Cliente extends Usuario
- Empleado(IdEmpleado, tipoEmpleado[FK(Rol.tipo)])  
  {(IdEmpleado) → (tipoEmpleado)}  
  Empleado extends Usuario
- ClienteEmpleado(IdCliente[FK(Cliente.IdCliente)], IdEmpleado[FK(Empleado.IdEmpleado)])  
  {(IdCliente, IdEmpleado) → ()}
- OperacionCuenta(numOrigen[FK(Cuenta.id)], numDestino[FK(Cuenta.id)], valor)  
  {(id) → (numOrigen, numDestino, valor)}  
  OperacionCuenta extends Operación
- OperacionPrestamo(numOrigen[FK(Prestamo.idPrestamo)], estado)  
  {(id) → (numOrigen, estado)}  
  OperacionPrestamo extends Operacion
- {(tipoDocumento, numeroDocumento, Rol) → (login, contraseña, nombre, nacionalidad, dirección, email, teléfono, ciudad, departamento, codigoPostal, Oficina)}
- Transacciones(OperacionBancaria[FK(OperacionBancaria.id)], Cuenta[FK(Cuenta.id)], PuntoDeAtencion[FK(PuntoDeAtencion.id)])  
  
  {(OperacionBancaria, Cuenta, PuntoDeAtencion) → (OperacionBancaria, Cuenta, PuntoDeAtencion)}
- OperacionBancaria(id, tipo, Cuenta1[FK(Cuenta.id)], Cuenta2[FK(Cuenta.id)], valor, fechaYHora)

$\{(id) \rightarrow (tipo, Cuenta1, Cuenta2, valor, fechaYHora)\}$

- Cuenta(id, saldo, tipo, estado, fechaCreacion, usuario[FK(Usuario.tipoDocumento, numeroDocumento, Rol)])

$\{(id) \rightarrow (saldo, tipo, estado, fechaCreacion, usuario)\}$

- Oficina(id, nombre, direccion, numPuntosPosibles)

$\{(id) \rightarrow (nombre, direccion, numPuntosPosibles)\}$

- Préstamo(id, tipo, monto, estado, Usuario[FK(Usuario.tipoDocumento, numeroDocumento, Rol)])

$\{(id) \rightarrow (tipo, monto, estado, Usuario)\}$

- PuntoDeAtencion(id, tipo, Oficina[FK(Oficina.id)])

$\{(id, tipo) \rightarrow (Oficina)\}$

## Formas normales

### 1. Primera Forma Normal (**1FN**)

Dado que nuestras relaciones no tienen atributos multivalorados, podemos concluir que se encuentran en la primera forma normal.

### 2. Segunda Forma Normal (**2FN**)

Las únicas dos relaciones que presentan una dependencia compuesta son Usuario y PuntoDeAtencion. En el caso de Usuario, su llave candidata está compuesta por los atributos tipoDocumento, numeroDocumento y Rol. Al examinar estos atributos, observamos que ninguno de ellos se encuentra en otra relación que nos permita determinar otros atributos no primos de Usuario, excepto por Rol, que es una llave foránea. Sin embargo, Rol tampoco nos lleva a atributos no primos de Usuario. En cuanto a PuntoDeAtencion, su llave candidata está compuesta por los atributos id y tipo. Al examinar estos atributos, observamos que ninguno de ellos se encuentra en otra relación que nos permita determinar el otro atributo de PuntoDeAtencion. Incluso si el otro atributo es la llave foránea de Oficina, esta no determina nada en PuntoDeAtencion. Por lo tanto, y dado que se encuentra en la primera forma normal, podemos concluir que también se encuentra en la segunda forma normal.

### 3. Tercera Forma Normal (**3FN**)

Para la tercera forma normal, tenemos cuatro relaciones que presentan dependencias que determinan llaves foráneas: Usuario, OperacionBancaria, Cuenta y PuntoDeAtencion. Las llaves candidatas de Usuario determinan la llave foránea Oficina. Sin embargo, dado que Oficina no determina ningún atributo dentro de Usuario, esto no representa un problema. En OperacionBancaria, sus llaves candidatas determinan las Cuentas 1 y 2, donde se realiza una

transacción. A su vez, Cuenta determina otra llave foránea, que es el tipoDocumento y numeroDocumento de Usuario. Esto tampoco representa un problema, ya que ni Cuenta ni Usuario determinan atributos de OperacionBancaria. Además, dentro de Usuario tampoco se determinan atributos de Cuenta. Finalmente, las llaves candidatas de PuntoDeAtencion determinan la llave foránea de Oficina. Esto tampoco es un problema, ya que dentro de Oficina no se determina ningún atributo que esté dentro de PuntoDeAtencion. Por lo tanto, y dado que se encuentra en la primera y segunda forma normal, podemos concluir que también se encuentra en la tercera forma normal.

#### 4. Forma Normal de *Boyce-Codd*

En la forma normal de Boyce-Codd, observamos que las llaves candidatas de las relaciones no se superponen. Usuario determina la llave primaria de Oficina, pero esto no es un problema ya que Oficina no determina la llave primaria de Usuario. OperacionBancaria determina la llave primaria de Cuenta, y Cuenta determina la llave primaria de Usuario. Sin embargo, como ni Cuenta tiene OperacionBancaria ni Usuario tiene Cuenta, esto no representa un problema. En Oficina y Préstamo, no tenemos ninguna relación con alguna llave foránea que pueda hacer que sus llaves primarias se superpongan. Finalmente, PuntoDeAtencion determina la llave primaria de Oficina, pero Oficina no determina PuntoDeAtencion, por lo que tampoco representa un problema. Por lo tanto, y dado que el modelo cumple con las tres formas normales anteriores, podemos concluir que todo el modelo se encuentra en la forma normal de Boyce-Codd.

## Gestión y casos

### Gestión de Usuarios

En el módulo de gestión de usuarios, se deben tener en cuenta varias restricciones. Para la creación de usuarios, un administrador de BancAndes puede registrar empleados del banco, como gerentes generales, gerentes de oficina y cajeros. Los clientes pueden ser registrados por los gerentes de oficina. Aunque, ahora se permite que un usuario tenga más de una dirección electrónica.

#### *Casos Exitosos:*

- Un administrador registra un gerente general.
- Un gerente de oficina registra a un cliente con varias direcciones electrónicas.

#### *Casos de Falla:*

- Intento de registrar un cliente por parte de un administrador.
- Intento de registrar dos usuarios con el mismo login.

### Gestión de Oficinas y Puntos de Atención

En cuanto a la gestión de oficinas y puntos de atención, se requiere un cuidado especial. Un administrador puede crear oficinas, y los puntos de atención (personalizada, cajero automático, digital) deben ser registrados asociándolos a una oficina existente.

### ***Casos Exitosos:***

- Un administrador crea una nueva oficina.
- Se registra un punto de atención personalizada asociado a una oficina.

### ***Casos de Falla:***

- Intento de crear un punto de atención digital sin asociarlo a una oficina.
- Intento de borrar un punto de atención con operaciones registradas.

## **Gestión de Cuentas Bancarias**

Las cuentas bancarias pueden ser de diferentes tipos (ahorros, corriente, AFC) y deben gestionarse con precaución. Los gerentes de oficina pueden crear cuentas, y los clientes pueden tener varias cuentas. Sin embargo, las cuentas pueden ser desactivadas por motivos de seguridad.

### ***Casos Exitosos:***

- Un gerente de oficina abre una nueva cuenta de ahorros para un cliente.
- Un cliente desactiva su cuenta por motivos de seguridad.

### ***Casos de Falla:***

- Intento de desactivar una cuenta con saldo no cero.
- Intento de abrir una cuenta sin especificar el tipo.
- Intento de crear una cuenta para un usuario que no existe en el sistema.

## **Gestión de Préstamos**

Los préstamos deben ser gestionados cuidadosamente, con verificación de estados y fechas. Un gerente de oficina puede crear un préstamo aprobado para un cliente.

### ***Casos Exitosos:***

- Un gerente de oficina aprueba un préstamo para vivienda.
- Un cajero registra el pago de una cuota ordinaria.

### ***Casos de Falla:***

- Intento de cerrar un préstamo que aún tiene saldo pendiente.
- Intento de registrar un pago de cuota en un préstamo rechazado

# Atributos

## 1. Usuario:

- ***tipoDocumento***: Este atributo se refiere al tipo de documento de identificación del usuario, como cédula de ciudadanía, pasaporte, etc.
- ***numeroDocumento***: El número de documento de identificación del usuario.
- ***FK(Rol.tipo)***: Clave externa que establece la relación con el tipo de rol que tiene el usuario en el sistema (cliente, cajero, gerente, etc.).
- ***login***: Nombre de usuario utilizado para iniciar sesión.
- ***contrasena***: Contraseña asociada al nombre de usuario para iniciar sesión.
- ***nombre***: Nombre completo del usuario.
- ***nacionalidad***: Nacionalidad del usuario.
- ***direccion***: Dirección física del usuario.
- ***email***: Dirección de correo electrónico del usuario.
- ***telefono***: Número de teléfono del usuario.
- ***ciudad*, departamento, codigoPostal**: Información relacionada con la ubicación del usuario.
- ***FK(Oficina.id)***: Clave externa que establece la relación con la oficina a la que está asociado el usuario.

## 2. OperacionCuenta:

- ***id***: Identificador único de la operación de cuenta.
- ***tipo***: Tipo de operación realizada en la cuenta (consignación, retiro, transferencia, etc.).
- ***FK(cuenta.id)***: Clave externa que establece la relación con la cuenta involucrada en la operación.
- ***valor***: Monto de la operación.
- ***fechaYHora***: Fecha y hora en que se realizó la operación.
- ***numOrigen*, *numDestino***: Números de cuenta de origen y destino, respectivamente, en caso de transferencias.

## 3. Transacciones:

- ***FK(OperacionCuenta.id)***: Clave externa que establece la relación con la operación de cuenta asociada a la transacción.
- ***FK(Cuenta.id)***: Clave externa que establece la relación con la cuenta involucrada en la transacción.
- ***FK(PuntoDeAtencion.id)***: Clave externa que establece la relación con el punto de atención donde se realizó la transacción.

#### 4. Cuenta:

- ***id***: Identificador único de la cuenta.
- ***saldo***: Saldo actual de la cuenta.
- ***tipo***: Tipo de cuenta (ahorros, corriente, etc.).
- ***estado***: Estado actual de la cuenta (activa, cerrada, desactivada).
- ***FK(IdCliente, Usuario.tipoDocumento, Usuario.numeroDocumento, Usuario.FK(Rol.tipo))***: Llave externa que establece la relación con el cliente titular de la cuenta.

#### 5. Oficina:

- ***id***: Identificador único de la oficina.
- ***nombre***: Nombre de la oficina.
- ***direccion***: Dirección física de la oficina.
- ***numPuntosPosibles***: Número de puntos de atención posibles en la oficina.

#### 6. OperacionPrestamo:

- ***id***: Identificador único de la operación de préstamo.
- ***tipo***: Tipo de operación realizada en el préstamo (solicitud, aprobación, pago de cuota, etc.).
- ***estado***: Estado actual del préstamo.
- ***FK(Id.Prestamo)***: Clave externa que establece la relación con el préstamo asociado a la operación.
- ***numOrigen***: Número de cuenta origen asociada a la operación.
- ***fechaYHora***: Fecha y hora en que se realizó la operación.

#### 7. PuntoDeAtencion:

- ***id***: Identificador único del punto de atención.

- ***FK(Oficina.id)***: Clave externa que establece la relación con la oficina a la que pertenece el punto de atención.

- ***tipo***: Tipo de punto de atención (personalizada, cajero automático, digital).

## 8. Rol:

- ***tipo***: Tipo de rol que puede tener un usuario en el sistema (cliente, cajero, gerente, etc.).

- ***FK(Usuario.numeroIdentificacion)***: Clave externa que establece la relación con el número de identificación del usuario.

## 9. Prestamo:

- ***id***: Identificador único del préstamo.

- ***tipo***: Tipo de préstamo (vivienda, estudio, automóvil, etc.).

- ***valor***: Valor del préstamo.

- ***estado***: Estado actual del préstamo.

- ***interes***: Tasa de interés del préstamo.

- ***numCuotas***: Número de cuotas del préstamo.

- ***numMes***: Número de meses para el pago del préstamo.

- ***FK(Usuario.numeroIdentificacion)***: Clave externa que establece la relación con el cliente titular del préstamo.

- ***valorCuota***: Valor de la cuota mensual del préstamo.

## 10. Empleado:

- ***id***: Identificador único del empleado.

- ***FK(Oficina.id)***: Clave externa que establece la relación con la oficina a la que pertenece el empleado.

- ***tipo***: Tipo de empleado (gerente, cajero, etc.).

## 11. ClienteEmpleado:

- ***id***: Identificador único del cliente-empleado.

- ***cliente***: Clave externa que establece la relación con el cliente asociado al empleado.

- ***empleado***: Clave externa que establece la relación con el empleado asociado al cliente.

## 12. Operación:

- **id:** Identificador único de la operación.
- **tipo:** Tipo de operación realizada en el sistema (crear usuario, crear cuenta, etc.).
- **fechaYHora:** Fecha y hora en que se realizó la operación.

# Justificación de relaciones

## 1. Usuario y Rol:

- **Relación:** Uno a muchos (1:N)
- **Justificación:** Un usuario puede tener uno o varios roles en el sistema bancario (por ejemplo, un cliente puede ser también un empleado). Por lo tanto, la relación es uno a muchos, ya que un mismo rol puede ser asignado a múltiples usuarios, pero un usuario solo puede tener un rol específico.

## 2. OperacionCuenta y Cuenta:

- **Relación:** Muchos a uno (N:1)
- **Justificación:** Cada operación de cuenta está asociada a una cuenta específica en la que se realiza la operación. Sin embargo, una cuenta puede tener múltiples operaciones asociadas a lo largo del tiempo. Por lo tanto, la relación es muchos a uno, ya que múltiples operaciones de cuenta están vinculadas a una única cuenta.

## 3. Transacciones y OperacionCuenta, Cuenta, PuntoDeAtencion:

- **Relación:** Muchos a uno (N:1)
- **Justificación:** Cada transacción realizada en el sistema bancario está asociada a una operación de cuenta específica, una cuenta específica y un punto de atención específico donde se realizó la transacción. Sin embargo, una operación de cuenta, una cuenta y un punto de atención pueden estar asociados a múltiples transacciones a lo largo del tiempo. Por lo tanto, la relación es muchos a uno.

## 4. Oficina y PuntoDeAtencion:

- **Relación:** Uno a muchos (1:N)
- **Justificación:** Una oficina puede tener múltiples puntos de atención asociados, como puestos de atención personalizada, cajeros automáticos y servicios digitales. Sin embargo, cada punto de atención está asociado a una única oficina. Por lo tanto, la relación es uno a muchos.

## 5. OperacionPrestamo y Prestamo:

- **Relación:** Muchos a uno (N:1)



- **Justificación:** Cada operación realizada en un préstamo específico está asociada a ese préstamo en particular. Sin embargo, un préstamo puede tener múltiples operaciones asociadas, como solicitudes, aprobaciones y pagos de cuotas. Por lo tanto, la relación es muchos a uno.

## 6. Empleado y Oficina:

- **Relación:** Muchos a uno (N:1)

- **Justificación:** Varios empleados pueden trabajar en una misma oficina bancaria, pero cada empleado está asignado a una única oficina. Por lo tanto, la relación es muchos a uno.

## 7. ClienteEmpleado y Cliente y Empleado:

- **Relación:** Uno a uno (1:1)

- **Justificación:** Esta relación establece la asociación entre un cliente y un empleado en el sistema bancario. Cada cliente puede tener una relación específica con un empleado, y viceversa. Por lo tanto, la relación es uno a uno. Todos estos son herencia de usuario y se relacionan también ClienteEmpleado como intermediario entre Cliente y Empleado por medio de una agregación uno a uno.

# Casos de prueba

Dado el nuevo esquema y sus restricciones, proporcionaremos propuestas de tuplas para cada caso de falla especificado en la gestión de usuarios, oficinas y puntos de atención, cuentas bancarias y préstamos.

## Gestión de Usuarios

1. Intento de registrar un cliente por parte de un administrador:

- **Usuario:** (cliente, tipoDocumento, numeroDocumento, login, contrasena, nombre, nacionalidad, direccion, email, telefono, ciudad, departamento, codigoPostal, oficina\_id)

- **Tupla de prueba:** (cliente, CC, 1234567890, cliente1, pass123, Juan Perez, Colombia, Calle 123, juan@example.com, 1234567890, Bogotá, Cundinamarca, 110011, null)

- **Justificación:** Se intenta registrar un cliente por parte de un administrador, lo cual no está permitido según las restricciones.

2. Intento de registrar dos usuarios con el mismo login:

- **Usuario:** (gerente, tipoDocumento, numeroDocumento, login, contrasena, nombre, nacionalidad, direccion, email, telefono, ciudad, departamento, codigoPostal, oficina\_id)

- **Tupla de prueba 1:** (gerente, CC, 987654321, gerente1, pass123, Maria Lopez, Colombia, Calle 456, maria@example.com, 0987654321, Medellín, Antioquia, 220022, null)

- **Tupla de prueba 2:** (gerente, CC, 135792468, gerente1, pass456, Carlos Gomez, Colombia, Calle 789, carlos@example.com, 246813579, Barranquilla, Atlántico, 330033, null)

- **Justificación:** Se intenta registrar dos usuarios con el mismo login "gerente1", lo cual no está permitido según las restricciones.

## Gestión de Oficinas y Puntos de Atención

Intento de crear un punto de atención digital sin asociarlo a una oficina:

- **PuntoDeAtencion:** (id, oficina\_id, tipo)

- **Tupla de prueba:** (1, null, digital)

- **Justificación:** Se intenta crear un punto de atención digital sin asociarlo a ninguna oficina, lo cual no está permitido según las restricciones.

Intento de borrar un punto de atención con operaciones registradas:

- **Transacciones:** (operacionCuenta\_id, cuenta\_id, puntoDeAtencion\_id)

- **Tupla de prueba:** (1, 1, 1)

- **Justificación:** Se intenta borrar un punto de atención que tiene operaciones registradas, lo cual no está permitido según las restricciones.

## Gestión de Cuentas Bancarias

Intento de desactivar una cuenta con saldo no cero:

- **Cuenta:** (id, saldo, tipo, estado, usuario\_id)

- **Tupla de prueba:** (1, 50000, ahorros, activa, 1)

- **Justificación:** Se intenta desactivar una cuenta que tiene saldo no cero, lo cual no está permitido según las restricciones.

Intento de abrir una cuenta sin especificar el tipo:

- **Cuenta:** (id, saldo, tipo, estado, usuario\_id)

- **Tupla de prueba:** (null, 0, null, activa, 1)

- **Justificación:** Se intenta abrir una cuenta sin especificar el tipo, lo cual no está permitido según las restricciones.

Intento de crear una cuenta para un usuario que no existe en el sistema:

- **Cuenta:** (id, saldo, tipo, estado, usuario\_id)

- **Tupla de prueba:** (1, 0, ahorros, activa, 999)

- **Justificación:** Se intenta crear una cuenta para un usuario que no existe en el sistema, lo cual no está permitido según las restricciones.

## Gestión de Préstamos

Intento de cerrar un préstamo que aún tiene saldo pendiente:

- **Prestamo:** (id, tipo, valor, estado, interes, numCuotas, numMes, usuario\_id, valorCuota)

- **Tupla de prueba:** (1, vivienda, 10000000, aprobado, 0.05, 120, 240, 1, 120000)

- **Justificación:** Se intenta cerrar un préstamo que aún tiene saldo pendiente, lo cual no está permitido según las restricciones.

Intento de registrar un pago de cuota en un préstamo rechazado:

- **OperacionPrestamo:** (id, tipo, estado, prestamo\_id, numOrigen, fechaYHora)

- **Tupla de prueba:** (1, pago cuota, rechazado, 1, 123, 2024-03-01 10:00:00)

- **Justificación:** Se intenta registrar un pago de cuota en un préstamo que está en estado rechazado, lo cual no está permitido según las restricciones.

Estas propuestas de tuplas de prueba abordan los casos de falla especificados y ayudan a validar las restricciones definidas en cada uno de los módulos del sistema bancario.

## Cambios significativos

El cambio que hicimos del modelo inicial al nuevo refleja un refinamiento de la estructura de la base de datos para adaptarse mejor a las necesidades del sistema.

### Herencias:

- En el modelo inicial, había una sola tabla para capturar toda la información relacionada con el usuario, sin distinciones de roles claramente definidas, aparte de una tabla que funcionaba como enumeración.

- El nuevo modelo introdujo tablas y columnas especializadas para diferentes tipos de usuarios (como cliente y empleado), permitiendo una estructura de datos basada en roles que es mucho más clara. Esto respaldaría el principio de herencia en el diseño de bases de datos, donde los datos comunes se almacenan en una tabla base (usuario), y los datos especializados se almacenan en subtablas, por eso lo hicimos así.

### Operaciones:

- El modelo inicial no era tan específico acerca de los atributos y relaciones de operación, así que, en el nuevo modelo, definimos OperacionCuenta y OperacionPrestamo con atributos añadidos como fechaYHora para registrar cuándo ocurrieron las operaciones.

## **Clientes y Empleados:**

- Inicialmente, las responsabilidades de los clientes y empleados no se distinguían y había una falta de relaciones claras con otras entidades en el sistema.
- En el nuevo modelo las entidades Cliente y Empleado son distintas y están conectadas a Usuario a través de la herencia, mostrando que ambos son tipos de usuarios, pero con atributos y relaciones adicionales. La entidad ClienteEmpleado se introdujo para responder al escenario donde un usuario puede ser tanto un cliente como un empleado, manteniendo una separación de roles mucho más clara que la anterior.

## **Cuenta (Fecha de Creación):**

- Añadir fechaCreacion a la entidad Cuenta en el nuevo modelo permite rastrear la fecha en que se creó una cuenta. Esto es útil para varios procesos bancarios, como el cálculo de intereses o la auditoría.

## **Conclusión**

A lo largo de este proyecto hemos llevado a cabo un proceso detallado de normalización y análisis de la base de datos para la página de BancAndes. Hemos progresado de un diseño inicial que, aunque funcional, presentaba ambigüedades y posibles redundancias, hacia un modelo altamente refinado. Este nuevo esquema refleja una clara distinción entre las entidades y sus relaciones, asegurando que cada elemento de la base de datos tiene un propósito definido y está estructurado de manera lógica.