

# Iteracion 2 BancAndes

Carlos Mario Peña Silva, Ángel Felipe Restrepo Castillo  
Sistemas Transaccionales 2021-20  
Universidad de los Andes, Bogotá, Colombia  
{cm.penas, af.restrepo}@uniandes.edu.co  
Octubre 19 de 2021

## TABLA DE CONTENIDO

<b>INTRDUCCION .....</b>	<b>2</b>
<b>ESQUEMA BANCANDES .....</b>	<b>2</b>
SQL: .....	2
<b>POBLACION TABLAS .....</b>	<b>11</b>
IMAGEN: .....	11
<b>MODELO RELACIONAL.....</b>	<b>13</b>
TABLAS: .....	13
<b>CONCLUSIÓN .....</b>	<b>14</b>

## INTRDUCCION

En este documento se plasma las sentencias sql y metodos implementados en java con el cual se creo el esquema BancAndes con su respectivo funcionamiento y sus reglas de negocio. En resumen se plasmara el avance realizado en la aplicacion y cuales son sus pilares para su funcionamiento.

## Esquema BancAndes

SQL:

```
CREATE SEQUENCE BancAndes_sequence;
CREATE TABLE USUARIO
(
    numeroDocumento INT NOT NULL ,
    tipoDocumento VARCHAR(255) NOT NULL ,
    login VARCHAR (255) NOT NULL UNIQUE,
    palabraClave VARCHAR (255)NOT NULL UNIQUE,
    nombre VARCHAR (255)NOT NULL,
    nacionalidad VARCHAR (255)NOT NULL ,
    direccionFisica VARCHAR (255)NOT NULL ,
    direccionElectronica VARCHAR (255)NOT NULL UNIQUE,
    telefono INT NOT NULL UNIQUE,
    ciudad VARCHAR (255) NOT NULL,
    departamento VARCHAR (255) NOT NULL,
    codigoPostal INT NOT NULL ,
);
ALTER TABLE USUARIO
ADD CONSTRAINT PK_USUARIO
PRIMARY KEY (numeroDocumento, tipoDocumento);

ALTER TABLE USUARIO
ADD CONSTRAINT CK_TIPO_DOCUMENTO
CHECK (tipoDocumento IN ('Registro civil', 'tarjeta de identidad', 'cedula de
ciudadania', 'NIT', 'Pasaporte', 'Cedula de extranjeria' ));

CREATE TABLE EMPLEADO
(
    numeroID INT NOT NULL,
    tipoEmpleado VARCHAR (255)
);
```

```
ALTER TABLE EMPLEADO
ADD CONSTRAINT PK_EMPLEADO
PRIMARY KEY (numeroID);
```

```
ALTER TABLE EMPLEADO
ADD tipoDocumentoUsuario VARCHAR(255);
```

```
ALTER TABLE EMPLEADO
MODIFY tipoDocumentoUsuario VARCHAR (255) NOT NULL ;
```

```
ALTER TABLE EMPLEADO
ADD CONSTRAINT FK_E_USUARIO
FOREIGN KEY (numeroID, tipoDocumentoUsuario) REFERENCES USUARIO (numeroDocumento,
tipoDocumento);
```

```
ALTER TABLE EMPLEADO
MODIFY tipoEmpleado VARCHAR (255) NOT NULL;
```

```
ALTER TABLE EMPLEADO
ADD CONSTRAINT CK_TIPO_EMPLEADO
CHECK (TIPOEMPLEADO IN ('Asesor', 'Cajero', 'Gerente oficina', 'Gerente general'
));
```

```
ALTER TABLE EMPLEADO
ADD CONSTRAINT CK_TIPO_DOCUMENTO_EMPLEADO
CHECK (tipoDocumentoUsuario IN ('Registro civil', 'tarjeta de identidad', 'cedula
de ciudadania', 'NIT', 'Pasaporte', 'Cedula de extranjeria' ));
```

```
CREATE TABLE OFICINA (
nombre VARCHAR (255),
direccion VARCHAR (255) NOT NULL UNIQUE,
numPuntosAtencion INT,
idEmpleado INT
);
```

```
ALTER TABLE OFICINA
ADD CONSTRAINT PK_OFICINA
PRIMARY KEY (nombre);
```

```
ALTER TABLE OFICINA
ADD CONSTRAINT NUMERO_ID_EMPLEADO
FOREIGN KEY (idEmpleado)
REFERENCES EMPLEADO(numeroId);
```

```
ALTER TABLE OFICINA
MODIFY idEmpleado INT NOT NULL;
```

```
CREATE TABLE PUNTODEATENCION
(
id INT,
tipoPuntoAtencion VARCHAR (255) NOT NULL,
localizacion VARCHAR (255) NOT NULL UNIQUE,
oficina VARCHAR (255)
);
```

```
ALTER TABLE PUNTODEATENCION
ADD CONSTRAINT PK_PUNTOATENCION
PRIMARY KEY (id);
```

```
ALTER TABLE PUNTODEATENCION
ADD CONSTRAINT CK_TIPO_PUNTO_ATENCION
CHECK (tipoPuntoAtencion IN ('Oficina', 'Cajero automatico', 'Virtual'));
```

```
ALTER TABLE PUNTODEATENCION
ADD CONSTRAINT NOMBRE_OFICINA_FK
FOREIGN KEY (oficina)
REFERENCES OFICINA(nombre);
```

```
CREATE TABLE CAJEROSPUNTOATENCION (
idPuntoAtencion INT NOT NULL,
idEmpleado INT NOT NULL UNIQUE
);
```

```
ALTER TABLE CAJEROSPUNTOATENCION
ADD CONSTRAINT PK_CAJEROSPUNTOATENCION
PRIMARY KEY (idPuntoAtencion, idEmpleado);
```

```
ALTER TABLE CAJEROSPUNTOATENCION
ADD CONSTRAINT ID_PUNTO_ATENCION_FK
FOREIGN KEY (idPuntoAtencion)
REFERENCES PUNTODEATENCION(id);
```

```
ALTER TABLE CAJEROSPUNTOATENCION
ADD CONSTRAINT ID_EMPLEADO_FK
FOREIGN KEY (idEmpleado)
```

```
REFERENCES EMPLEADO(numeroId);
```

```
CREATE TABLE CLIENTE  
(  
  numeroID INT,  
  tipoPersona VARCHAR (255)  
);
```

```
ALTER TABLE CLIENTE  
ADD tipoDocumentoUsuario VARCHAR(255);
```

```
ALTER TABLE CLIENTE  
MODIFY tipoDocumentoUsuario VARCHAR (255) NOT NULL ;
```

```
ALTER TABLE CLIENTE  
ADD CONSTRAINT PK_CLIENTE  
PRIMARY KEY (numeroID);
```

```
ALTER TABLE CLIENTE  
ADD CONSTRAINT CK_TIPO_CLIENTE  
CHECK (tipoPersona IN ('Natural', 'Juridica'));
```

```
ALTER TABLE CLIENTE  
ADD CONSTRAINT FK_C_CLIENTE  
FOREIGN KEY (numeroID, tipoDocumentoUsuario) REFERENCES USUARIO (numeroDocumento,  
tipoDocumento);
```

```
ALTER TABLE CLIENTE  
ADD CONSTRAINT CK_TIPO_DOCUMENTO_CLIENTE  
CHECK (tipoDocumentoUsuario IN ('Registro civil', 'tarjeta de identidad', 'cedula  
de ciudadanía', 'NIT', 'Pasaporte', 'Cedula de extranjería' ));
```

```
CREATE TABLE ADMINISTRADOR(  
  numeroIDAdministrador INT,  
  tipoDocumentoAdministrador VARCHAR (255)  
);
```

```
ALTER TABLE ADMINISTRADOR  
ADD CONSTRAINT FK_C_ADMINISTRADOR  
FOREIGN KEY (numeroIDAdministrador, tipoDocumentoAdministrador) REFERENCES  
USUARIO (numeroDocumento, tipoDocumento);
```

```
ALTER TABLE ADMINISTRADOR
MODIFY tipoDocumentoAdministrador VARCHAR (255) NOT NULL
;
```

```
ALTER TABLE ADMINISTRADOR
ADD CONSTRAINT PK_ADMINISTRADOR
PRIMARY KEY (numeroIDAdministrador);
```

```
ALTER TABLE ADMINISTRADOR
ADD CONSTRAINT CK_TIPO_DOCUMENTO_ADMINISTRADOR
CHECK (tipoDocumentoAdministrador IN ('Registro civil', 'tarjeta de identidad',
'cedula de ciudadanía', 'NIT', 'Pasaporte', 'Cedula de extranjería' ));
```

```
CREATE TABLE CUENTA (
numeroUnico INT,
tipoCuenta VARCHAR (255) NOT NULL,
saldo INT NOT NULL,
estado VARCHAR (255) NOT NULL,
numeroIDCliente INT,
fechaCreacionCuenta DATE NOT NULL,
fechaUltimoMovimiento DATE
);
```

```
ALTER TABLE CUENTA
ADD CONSTRAINT PK_CUENTA
PRIMARY KEY (numeroUnico);
```

```
ALTER TABLE CUENTA
ADD CONSTRAINT CK_TIPO_CUENTA
CHECK (tipoCuenta IN ('Ahorros', 'Corriente', 'AFC', 'CDT', 'Nomina' ));
```

```
ALTER TABLE CUENTA
ADD CONSTRAINT CK_ESTADO_CUENTA
CHECK (estado IN ('Activa', 'Cerrada'));
```

```
ALTER TABLE CUENTA
ADD CONSTRAINT CUENTAS_Cliente_FK
FOREIGN KEY (numeroIDCliente)
REFERENCES CLIENTE(numeroID);
```

```
CREATE TABLE OPERACIONESBANCARIAS
(
```

```
id INT,  
valor INT NOT NULL,  
fechaHora DATE NOT NULL,  
numeroCuenta INT,  
empleado INT,  
administrador INT  
);
```

```
ALTER TABLE OPERACIONESBANCARIAS  
ADD CONSTRAINT PK_OPERACIONESBANCARIAS  
PRIMARY KEY (id);
```

```
ALTER TABLE OPERACIONESBANCARIAS  
ADD CONSTRAINT NUMERO_CUENTA_FK  
FOREIGN KEY (numeroCuenta)  
REFERENCES CUENTA(numeroUnico);
```

```
ALTER TABLE OPERACIONESBANCARIAS  
ADD CONSTRAINT NUMERO_ADMINISTRADOR_FK  
FOREIGN KEY (administrador)  
REFERENCES ADMINISTRADOR(numeroIDAdministrador);
```

```
ALTER TABLE OPERACIONESBANCARIAS  
ADD CONSTRAINT NUMERO_EMPLEADO_FK  
FOREIGN KEY (empleado)  
REFERENCES ADMINISTRADOR(numeroIDAdministrador);
```

```
CREATE TABLE OPERACIONCUENTA  
(  
idOperacionCuenta INT,  
tipoOpeCuentna VARCHAR(255) NOT NULL  
);
```

```
ALTER TABLE OPERACIONCUENTA  
ADD CONSTRAINT PK_OPERACION_CUENTA  
PRIMARY KEY (idOperacionCuenta);
```

```
ALTER TABLE OPERACIONCUENTA  
ADD CONSTRAINT NUMERO_CUENTA_BANCARIA_FK  
FOREIGN KEY (idOperacionCuenta)  
REFERENCES OPERACIONESBANCARIAS(id);
```

```
ALTER TABLE OPERACIONCUENTA
```

```
ADD CONSTRAINT CK_TIPO_OPERACION_CUENTA
CHECK (tipoOpeCuentna IN ('Abrir', 'Cerrar', 'Transferir', 'Retirar', 'Consignar'
));
```

```
CREATE TABLE OPERACIONCDT(
idOperacionCDT INT,
tipoOpeCDT VARCHAR (255) NOT NULL ,
fechaVencimiento DATE NOT NULL,
tasaRendimientos INT NOT NULL
);
```

```
ALTER TABLE OPERACIONCDT
ADD CONSTRAINT PK_OPERACION_CDT
PRIMARY KEY (idOperacionCDT);
```

```
ALTER TABLE OPERACIONCDT
ADD CONSTRAINT CK_TIPO_OPERACION_CDT
CHECK (tipoOpeCDT IN ('Abrir', 'Cerrar', 'Renovar'));
```

```
ALTER TABLE OPERACIONCDT
ADD CONSTRAINT NUMERO_CUENTA_BANCARIA_CDT_FK
FOREIGN KEY (idOperacionCDT)
REFERENCES OPERACIONESBANCARIAS(id);
```

```
CREATE TABLE OPERACIONACCIONES (
idOperacionAcciones INT,
numAcciones INT,
tipoOpeAcciones VARCHAR(255)
);
```

```
ALTER TABLE OPERACIONACCIONES
ADD CONSTRAINT PK_OPERACION_ACCIONES
PRIMARY KEY (idOperacionAcciones);
```

```
ALTER TABLE OPERACIONACCIONES
ADD CONSTRAINT CK_TIPO_OPERACION_ACCIONES
CHECK (tipoOpeAcciones IN ('Vender', 'Comprar'));
```

```
ALTER TABLE OPERACIONACCIONES
MODIFY tipoOpeAcciones VARCHAR(255) NOT NULL;
```

```
ALTER TABLE OPERACIONACCIONES
MODIFY numAcciones INT NOT NULL;
```

```
ALTER TABLE OPERACIONACCIONES
```



```
ADD CONSTRAINT NUMERO_CUENTA_BANCARIA_ACCCIONES_FK
    FOREIGN KEY (idOperacionAcciones)
    REFERENCES OPERACIONESBANCARIAS(id);
```

```
CREATE TABLE OPERACIONINVERSION (
    idOperacionInversion INT,
    tipoOpeInversion VARCHAR (255) NOT NULL
);
```

```
ALTER TABLE OPERACIONINVERSION
ADD CONSTRAINT PK_OPERACION_INVERSION
PRIMARY KEY (idOperacionInversion);
```

```
ALTER TABLE OPERACIONINVERSION
ADD CONSTRAINT CK_TIPO_OPERACION_INVERSION
CHECK (tipoOpeInversion IN ('Abrir', 'Cerrar', 'Liquidar', 'Renovar'));
```

```
ALTER TABLE OPERACIONINVERSION
ADD CONSTRAINT NUMERO_CUENTA_BANCARIA_INVERSION_FK
    FOREIGN KEY (idOperacionInversion)
    REFERENCES OPERACIONESBANCARIAS(id);
```

```
CREATE TABLE PRESTAMO (
    id INT,
    monto INT NOT NULL,
    interes INT NOT NULL,
    numCuotas INT NOT NULL,
    diaPago DATE NOT NULL,
    estado VARCHAR (255) NOT NULL,
    tipoPrestamo VARCHAR (255) NOT NULL,
    idCliente INT NOT NULL,
);
```

```
ALTER TABLE PRESTAMO
ADD CONSTRAINT PK_PRESTAMO
PRIMARY KEY (id);
```

```
ALTER TABLE PRESTAMO
ADD CONSTRAINT ID_CLIENTE_FK
    FOREIGN KEY (idCliente)
    REFERENCES CLIENTE(numeroId);
```

```
ALTER TABLE PRESTAMO
```

```
ADD CONSTRAINT CK_ESTADO_PRESTAMO
CHECK (estado IN ('Activo', 'Cerrado'));
```

```
ALTER TABLE PRESTAMO
ADD CONSTRAINT CK_TIPO_PRESTAMO
CHECK (tipoPrestamo IN ('Vivienda', 'Estudio', 'Automovil', 'CalamidadDomestica',
'Libre inversion'));
```

```
CREATE TABLE CUOTAMINIMAPRESTAMO
(
idPrestamo INT NOT NULL UNIQUE,
valorCuotaMinima INT NOT NULL
);
```

```
ALTER TABLE CUOTAMINIMAPRESTAMO
ADD CONSTRAINT PK_CUOTAMINIMAPRESTAMO
PRIMARY KEY (idPrestamo, valorCuotaMinima);
```

```
ALTER TABLE CUOTAMINIMAPRESTAMO
ADD CONSTRAINT ID_PRESTAMO_FK
FOREIGN KEY (idPrestamo)
REFERENCES PRESTAMO(id);
```

```
CREATE TABLE OPERACIONPRESTAMO
(
idOperacionPrestamo INT,
tipoOpePrestamo VARCHAR (255) NOT NULL,
idPrestamo INT NOT NULL,
valorPagado INT
);
```

```
ALTER TABLE OPERACIONPRESTAMO
ADD CONSTRAINT PK_OPERACION_PRESTAMO
PRIMARY KEY (idOperacionPrestamo);
```

```
ALTER TABLE OPERACIONPRESTAMO
ADD CONSTRAINT PRESTAMO_ID_FK
FOREIGN KEY (idPrestamo)
REFERENCES PRESTAMO(id);
```

```
ALTER TABLE OPERACIONPRESTAMO
ADD CONSTRAINT CK_TIPO_OPERACIONPRESTAMO
CHECK (tipoOpePrestamo IN ('Abrir', 'Transferir', 'Cerrar', 'Consignar',
'Retirar'));
```

Empleado

## Cliente

## Administrador

Usuario

Cuenta

## Operaciones Bancarias

## Prestamo

	ID	MONTO	INTERES	NUMCUOTAS	DIAPAGO	ESTADO	TIPOPRESTAMO	IDCLIENTE
1	41	0	1	24	5	Activo	Vivienda	8
2	65	235698	2	65	5	Activo	Libre inversion	268754

### Operación préstamo

	IDOPERACIONPRESTAMO	TIPOOPEPRESTAMO	IDPRESTAMO	VALORPAGADO
1	89	Consignar	65	32000

### Oficina

	NOMBRE	DIRECCION	NUMPUNTOSATENCION	IDEMPLEADO
1	asdsdjjjjj	adadadvvvv	5	123456789
2	asdsd	adadad	5	123456789
3	BancAndes de la 38	carrera 38 # 24c-5	1	325654
4	m	m	1	1
5	gggggg	ggggg	3	123456789
6	ghgh	ghgh	6	1

### Punto de atencion

	ID	TIPOPUNTOATENCION	LOCALIZACION	OFICINA
1	18	Oficina	m	m
2	4	Oficina	carrera 38 # 24c-5	BancAndes de la 38

### Cajeros punto de atencion

	IDPUNTOATENCION	IDEMPLEADO
1	4	656585

### Operación cuenta

	IDOPERACIONCUENTA	TIPOOPECUENTNA
1	234567	Transferir

### Operación inversion

	IDOPERACIONINVERSION	TIPOOPEINVERSION
1	125458	Liquidar

# Modelo Relacional

Tablas:

Usuario:

Usuario						
numeroDocumento	tipoDocumento				login	palabraClave
PK, NN, ND	PK, NN				NN, ND	NN, ND
direccionElectronica	telefono	ciudad	departamento	codigoPostal	nombre	
NN, ND	NN, ND	NN	NN	NN	NN	
					nacionalidad	direccionFisica
					NN	NN

Empleado:

Empleado		
numeroID	tipoEmpleado	TipoDocumentoUsuario
PK, FK_usuario.numeroID, NN, ND	{Asesor, Cajero, Gerente oficina, Gerente general }, NN	PK, FK_usuario.tipoDocumento, NN, ND

Cliente:

Cliente		
numeroID	tipoPersona	TipoDocumentoUsuario
PK, FK_usuario.numeroID, NN, ND	{natural, juridica }	PK, FK_usuario.tipoDocumento, NN, ND

Administrador:

Administrador	
NumeroID	TipoDocumentoAdministrador
PK, FK_usuario.numeroID, NN, ND	PK, FK_usuario.tipoDocumento, NN, ND

Cuenta:

Cuenta						
numeroUnico	tipoCuenta	saldo	estado	numeroDCliente	fechaCreacion	fechaUltimoMovimiento
PK, NN, ND	{Ahorros, Corriente, AFC, CDT, Nomina}	NN	{Activo, Cerrado}	FK_cliente.numeroID, NN, ND	NN	NN

Operaciones Bancarias:

OperacionesBancarias					
id	valor	fechaHora	numeroCuenta	empleado	Administrador
PK, NN, ND	NN	NN	FK_cuenta.numeroUnico, ND	FK_empleado.numeroID, ND	FK_administrador.numeroID, ND

Operación Cuenta:

OperacionCuenta	
idOperacion	tipoOpeCuenta
PK, FK_operacionesBancarias.id, NN, ND	{Abrir, Cerrar, Transferir, Retirar, Consignar}, NN

Operación CDT:

OperacionCDT			
idOperacionCDT	tipoOpeCDT	fechaVencimiento	tasaRendimientos
PK, FK_operacionesBancarias.id, NN, ND	{Abrir, Cerrar, Renovar}, NN	NN	NN

OperacionAcciones:

OperacionAcciones		
idOperacionAcciones	numAcciones	tipoOpeAcciones
PK, FK_operacionesBancarias.id, NN, ND	NN	{Vender, Comprar}, NN

OperacionInversion:

OperacionInversion	
idOperacionInversion	tipoOpeInversion
PK, FK_operacionesBancarias.id, NN, ND	{Abrir, Cerrar, Liquidar, Renovar}, NN

### Prestamo:

Prestamo			
id	monto	interes	numCuotas
PK, NN, ND	NN	NN	NN
diaPago	estado	tipoPrestamo	idCliente
NN	{Activo, Cerrado}, NN	{Vivienda, Estudio, Automovil, CalamidadDomestica, LibreInversion}, NN	FK_cliente.id, NN, ND

### CuotaMinimaPrestamo:

CuotaMinimaPrestamo	
idPrestamo	valorCuotaMinima
PK, FK_prestamo.id, NN, ND	NN

### OperacionPrestamo:

OperacionPrestamo			
idPrestamo	tipoOpePrestamo	idPrestamo	valorPagado
PK, FK_prestamo.id, NN, ND	{Abrir, Transferir, Cerrar, Consignar, Retirar}, NN	FK, NN	NN

### Oficina:

Oficina			
nombre	direccion	numPuntosAtencion	idEmpleado
PK, NN, ND	NN, ND	NN	FK_empleado.id, NN, ND

### PuntoAtencion:

puntoAtencion			
id	tipoPuntoAtencion	localizacion	oficina
PK, NN, ND	{Oficina, CajeroAutomatico, Virtual}, NN	NN, ND	NN, FK

### CajerosPuntoAtencion:

CajerosPuntoAtencion	
idPuntoAtencion	idEmpleado
PK, FK_puntoAtencion.id, NN,	PK, FK_empleado.id, NN, ND

## Conclusión

La creacion de la aplicacion de bancaAndes fue un reto el cual tuvimos que realizar durentes un gran periodo de tiempo, esto nos deajo enseñansas las cuales nos van a servir a futuro. Desde la creacion del esquema hasta el view de la aplicacion fueron todas actividades las cuales nos pusieron a prueba, pero que con tiempo y determinación creemos que logramos superar.