

**Universidad Tecnológica de Panamá**  
**Facultad de Ingeniería en Sistemas Computacionales**  
**Licenciatura en Ingeniería en Sistemas de Información Sistemas**

**Cursado:** Bases de Datos II

**L6 – Diseño de Base de Datos para un Modelo de Negocio**

**Facilitador:** Ing. Henry Lezcano

**Integrantes:**

Batista, Johel {8-914-587}

Pinilla, Miguel {8-975-2460}

Riley, Rolando {8-972-1033}

Villarreal, Andrés {8-970-1267}

**Grupo:** 1IF-131

**Fecha de Entrega:** Sábado 29 de octubre de 2022

## Código de la Base de Datos del Modelo de Negocio

```
-- Creación de Tablas
CREATE TABLE Sucursal (
    cod_sucursal NUMBER,
    nombresucursal VARCHAR2(30),
    monto_prestamos NUMBER,
    CONSTRAINT pk_Sucursal PRIMARY KEY (cod_sucursal)
);

CREATE TABLE tipos_correos(
    cod_correo NUMBER NOT NULL,
    descripcion VARCHAR2(50) NOT NULL,
    CONSTRAINT pk_tipos_correos primary key(cod_correo)
);

CREATE TABLE tipos_prestamos(
    cod_prestamo NUMBER NOT NULL,
    descripcion VARCHAR2(100) NOT NULL,
    CONSTRAINT pk_tipos_prestamos primary key (cod_prestamo)
);

CREATE TABLE tipos_telefonos(
    cod_telefono NUMBER NOT NULL,
    descripcion VARCHAR2(50) NOT NULL,
    CONSTRAINT pk_tipos_telefonos primary key (cod_telefono),
    CONSTRAINT u_telefonos unique (descripcion)
);

CREATE TABLE profesiones(
    id_profesion NUMBER NOT NULL,
    descripcion VARCHAR2(100) NOT NULL,
    CONSTRAINT pk_profesion primary key (id_profesion)
);

CREATE TABLE clientes(
    id_cliente NUMBER NOT NULL,
    cedula VARCHAR2(15) NOT NULL UNIQUE,
    nombre VARCHAR2(100) NOT NULL,
    apellido VARCHAR2(100) NOT NULL,
    sexo CHAR(1) NOT NULL CHECK (sexo IN ('M', 'F')),
    fecha_nac date NOT NULL,
    cod_profesion NUMBER NOT NULL,
    CONSTRAINT pk_clientes primary key (id_cliente),
```

```

        CONSTRAINT fk_profesion FOREIGN KEY (cod_profesion) REFERENCES
Profesiones(id_profesion)
);

CREATE TABLE clientes_correos(
    id_cliente NUMBER NOT NULL,
    id_correo NUMBER NOT NULL,
    correo VARCHAR2(100) NOT NULL,
    CONSTRAINT pk_clientes_correos primary key(id_cliente, id_correo),
    CONSTRAINT fk_clientes_correos foreign key(id_cliente) references
clientes(id_cliente),
    CONSTRAINT fk_correo foreign key(id_correo) REFERENCES
tipos_correos(cod_correo)
);

CREATE TABLE clientes_telefonos(
    id_cliente NUMBER NOT NULL,
    id_telefono NUMBER NOT NULL,
    telefono NUMBER NOT NULL,
    CONSTRAINT pk_clientes_telefonos PRIMARY KEY(id_cliente, id_telefono),
    CONSTRAINT fk_cliente_telefono foreign key(id_cliente) REFERENCES
clientes(id_cliente),
    CONSTRAINT fk_telefono foreign key (id_telefono) REFERENCES
tipos_telefonos(cod_telefono)
);

CREATE TABLE prestamos(
    num_prestamo NUMBER UNIQUE NOT NULL,
    id_cliente NUMBER NOT NULL,
    tipoprestamo NUMBER NOT NULL,
    fecha_aprobado date NOT NULL,
    monto_aprobado NUMBER NOT NULL,
    tasa_interes NUMBER NOT NULL,
    letra_mensual NUMBER NOT NULL,
    monto_pagado NUMBER DEFAULT 0 NOT NULL,
    monto_intereses NUMBER DEFAULT 0 NOT NULL,
    fecha_pago date,
    CONSTRAINT pk_prestamos PRIMARY KEY (id_cliente, tipoprestamo),
    CONSTRAINT fk_tipoprestamo foreign key(tipoprestamo) REFERENCES
tipos_prestamos(cod_prestamo),
    CONSTRAINT fk_cliente foreign key(id_cliente) REFERENCES clientes(id_cliente)
);

CREATE TABLE SucursalPrestamos (
    id_sucursal NUMBER,

```

```

        cod_prestamo NUMBER,
        CONSTRAINT pk_SucursalPrestamos PRIMARY KEY (id_sucursal, cod_prestamo),
        CONSTRAINT fk_SucursalPrestamos_Sucursal FOREIGN KEY (id_sucursal) REFERENCES
Sucursal (id_sucursal),
        CONSTRAINT fk_SucursalPrestamos_Prestamo FOREIGN KEY (cod_prestamo)
REFERENCES prestamos (cod_prestamo)
)

```

```

CREATE SEQUENCE seq_num_prestamo
    start with 1
    increment by 1
    minvalue 1
    cache 20;

```

```

CREATE TABLE Transacpagos (
    id_transaccion NUMBER,
    cod_sucursal NUMBER,
    id_cliente NUMBER,
    tipoprestamo NUMBER,
    fechatransaccion DATE,
    monto_de_pago NUMBER,
    estado CHAR,
    fechainsercion DATE,
    usuario VARCHAR2(25),
    CONSTRAINT pk_Transacpagos PRIMARY KEY (id_transaccion),
    CONSTRAINT fk_Sucursal_pagos FOREIGN KEY (cod_sucursal) REFERENCES Sucursal
(cod_sucursal),
    CONSTRAINT fk_Cliente_pagos FOREIGN KEY (id_cliente) REFERENCES clientes
(id_cliente),
    CONSTRAINT fk_tipos_prestamos FOREIGN KEY (tipoprestamo) REFERENCES
tipos_prestamos (cod_prestamo)
);

```

```

CREATE SEQUENCE seq_id_transac
    minvalue 1
    start with 1
    increment by 1
    Cache 20;

```

```
-- alter a la tabla
```

```

ALTER TABLE Clientes
ADD Edad NUMBER;
ALTER TABLE Prestamos
ADD saldo_actual NUMBER NOT NULL;

```

```

ALTER TABLE Prestamos
ADD interes_pagado NUMBER NOT NULL;
ALTER TABLE Prestamos
ADD fecha_modificacion date;
ALTER TABLE Prestamos
ADD usuario VARCHAR2(25)NOT NULL;
ALTER TABLE Prestamos
ADD id_sucursal NUMBER NOT NULL;
ALTER TABLE Clientes
ADD id_sucursal NUMBER;

-- *Creación de procedimientos e inserción de datos*
-- Creando secuencia de profesiones
CREATE SEQUENCE seq_id_profesion
    minvalue 1
    start with 1
    increment by 1
    cache 20;

-- Profesiones

CREATE OR REPLACE procedure agregar_prof(
    p_descripcion profesiones.descripcion%TYPE
) AS
BEGIN
    INSERT INTO profesiones
    VALUES (seq_id_profesion.nextval, p_descripcion);

    EXCEPTION
        WHEN DUP_VAL_ON_INDEX THEN
            DBMS_OUTPUT.PUT_LINE('Datos repetidos');
        WHEN OTHERS THEN
            DBMS_OUTPUT.PUT_LINE('Error en la insercion de datos');
    END agregar_prof;

-- Insertando las profesiones
BEGIN
    agregar_prof('Pintor');
    agregar_prof('CEO');
    agregar_prof('Capataz');
END;

--Creando secuencia para clientes
CREATE SEQUENCE seq_id_cliente

```

```

    minvalue 1
    start with 1
    increment by 1
    Cache 20;

-- Clientes
CREATE OR REPLACE procedure agregar_cli(
    p_cedula clientes.cedula%TYPE,
    p_nombre clientes.nombre%TYPE,
    p_apellido clientes.apellido%TYPE,
    p_sexo clientes.sexo%TYPE,
    p_fecha_nac clientes.fecha_nac%TYPE,
    p_cod_profesion clientes.cod_profesion%TYPE,
    p_cod_sucursal clientes.id_sucursal%TYPE
) AS
    v_edad clientes.Edad%TYPE;
BEGIN
    v_edad := ROUND(TRUNC(MONTHS_BETWEEN(SYSDATE,p_fecha_nac))/12, 0);
    INSERT INTO clientes
    VALUES(seq_id_cliente.nextval, p_cedula, p_nombre,
        p_apellido, p_sexo, p_fecha_nac, p_cod_profesion, v_edad,
p_cod_sucursal);

    EXCEPTION
        WHEN DUP_VAL_ON_INDEX THEN
            DBMS_OUTPUT.PUT_LINE('Datos repetidos');
        WHEN value_error then
            DBMS_OUTPUT.PUT_LINE('error por tamaño de dato ingresado');
        WHEN OTHERS THEN
            DBMS_OUTPUT.PUT_LINE('error en la insercion de datos');
    END agregar_cli;

-- Insertando datos a la tabla clientes con su secuencia
BEGIN
    agregar_cli('8-958-1869','Jhair','pinilla','M','29-JUN-2000', 1, 1);
    agregar_cli('8-745-7448','Ermenegilda','Solis','F','29-JUL-1989', 2, 1);
    agregar_cli('8-970-1267','Andres','Villarreal','M','24-JAN-1999', 2, 1);
END;

-- tipos_correos
CREATE OR REPLACE procedure agregar_tipos_correos(
    p_cod_correo tipos_correos.cod_correo%TYPE,
    p_descripcion tipos_correos.descripcion%TYPE
) AS

```

```

BEGIN
    INSERT INTO tipos_correos
    VALUES (p_cod_correo, p_descripcion);

EXCEPTION
    WHEN DUP_VAL_ON_INDEX THEN
        DBMS_OUTPUT.PUT_LINE('Datos repetidos');
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE('Error en la insercion de datos');
END agregar_tipos_correos;

-- insertando datos al procedimiento
BEGIN
    agregar_tipos_correos(1, 'personal');
    agregar_tipos_correos(2, 'Institucional');
    agregar_tipos_correos(3, 'trabajo');
END;

-- clientes_correos
CREATE OR REPLACE procedure agregar_clientes_correos(
    p_id_cliente clientes_correos.id_cliente%TYPE,
    p_id_correo clientes_correos.id_correo%TYPE,
    p_correo clientes_correos.correo%TYPE
) AS
BEGIN
    INSERT INTO clientes_correos(id_cliente,id_correo,correo)
    VALUES(p_id_cliente,p_id_correo,p_correo);
EXCEPTION
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE('Error en la insercion de datos');
END agregar_clientes_correos;

-- Inserción en la tabla clientes_correos
BEGIN
    agregar_clientes_correos(1, 2, 'jhair@gmail.com');
    agregar_clientes_correos(2, 3, 'ermenegilda@gmail.com');
    agregar_clientes_correos(3, 1, 'andres@gmail.com');
END;

-- tipos_telefonos
-- Creación de procedimiento tipos_telefonos
CREATE OR REPLACE procedure agregar_tipos_telefonos(
    p_cod_telefono tipos_telefonos.cod_telefono%TYPE,
    p_descripcion tipos_telefonos.descripcion%TYPE

```

```

) AS
BEGIN
    INSERT INTO tipos_telefonos(cod_telefono,descripcion)
    VALUES(p_cod_telefono,p_descripcion);
EXCEPTION
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE('error en la insercion de datos');
END agregar_tipos_telefonos;

-- Insertando datos a tipos_telefonos
BEGIN
    agregar_tipos_telefonos(1,'personal');
    agregar_tipos_telefonos(2,'familiar');
    agregar_tipos_telefonos(3,'residencial');
    agregar_tipos_telefonos(4,'conyuge');
END;

-- clientes_telefonos
CREATE OR REPLACE PROCEDURE agregar_clien_tel(
    p_id_cliente clientes_telefonos.id_cliente%TYPE,
    p_id_telefono clientes_telefonos.id_telefono%TYPE,
    p_telefono clientes_telefonos.telefono%TYPE
) AS
BEGIN
    INSERT INTO clientes_telefonos
    VALUES(p_id_cliente,p_id_telefono,p_telefono);
EXCEPTION
    WHEN DUP_VAL_ON_INDEX THEN
        DBMS_OUTPUT.PUT_LINE('Datos repetidos');
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE('error en la insercion de datos');
END agregar_clien_tel;

-- Insercion de datos en clientes_telefonos
BEGIN
    agregar_clien_tel(1, 2, 29865418);
    agregar_clien_tel(2, 3, 37865477);
    agregar_clien_tel(3, 1, 44845455);
END;

-- PROCEDIMIENTO Sucursal
CREATE OR REPLACE procedure agregar_Sucursal(

```



```

p_cod_sucursal sucursal.cod_sucursal%TYPE,
p_nombresucursal sucursal.nombresucursal%TYPE,
p_montoprestamo sucursal.monto_prestamos%TYPE DEFAULT 0
) AS
BEGIN
    INSERT INTO Sucursal
    VALUES(p_cod_sucursal, p_nombresucursal, p_montoprestamo);
EXCEPTION
    WHEN DUP_VAL_ON_INDEX THEN
        DBMS_OUTPUT.PUT_LINE('Datos repetidos');
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE('error en la insercion de datos');
END agregar_Sucursal;

-- Insercion de datos en clientes_telefonos
BEGIN
    agregar_sucursal(1, 'San Miguelito');
    agregar_sucursal(2, 'Mananitas');
    agregar_sucursal(3, 'Atlapa');
END;

-- PROCEDIMIENTO DE AGREGAR PRÉSTAMOS A LA SUCURSAL
CREATE OR REPLACE PROCEDURE agregar_SucursalPrestamos(
    p_id_sucursal SucursalPrestamos.id_sucursal%TYPE,
    p_cod_prestamo SucursalPrestamos.cod_prestamo%TYPE
) AS
BEGIN
    INSERT INTO SucursalPrestamos
    VALUES(p_id_sucursal, p_cod_prestamo);
    COMMIT;
EXCEPTION
    WHEN DUP_VAL_ON_INDEX THEN
        DBMS_OUTPUT.PUT_LINE('Datos repetidos');
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE('error en la insercion de datos');
END agregar_SucursalPrestamos;

-- PROCEDIMIENTO TIPO PRESTAMOS
CREATE OR REPLACE PROCEDURE agregar_tipos_prestamos(
    p_cod_prestamo tipos_prestamos.cod_prestamo%TYPE,
    p_descripcion tipos_prestamos.descripcion%TYPE
) AS
BEGIN
    INSERT INTO tipos_prestamos
    VALUES(p_cod_prestamo, p_descripcion);

```

```

EXCEPTION
    WHEN DUP_VAL_ON_INDEX THEN
        DBMS_OUTPUT.PUT_LINE('Datos repetidos');
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE('error en la insercion de datos');
END agregar_tipos_prestamos;

-- Insercion de datos en clientes_telefonos
BEGIN
    agregar_tipos_prestamos(1, 'personal');
    agregar_tipos_prestamos(2, 'hipotecario');
    agregar_tipos_prestamos(3, 'auto');
END;

--PROCEDIMIENTO PRESTAMOS APROBADOS
CREATE OR REPLACE PROCEDURE PrestamoAprobado(
    p_id_cliente Clientes.id_cliente%TYPE,
    p_tipoprestamo Prestamos.tipoprestamo%TYPE,
    p_monto_aprobado Prestamos.monto_aprobado%TYPE,
    p_tasa_interes Prestamos.tasa_interes%TYPE,
    p_letra_mensual Prestamos.letra_mensual%TYPE,
    p_cod_sucursal Sucursal.cod_sucursal%TYPE
) AS
BEGIN
    INSERT INTO Prestamos
    VALUES (seq_num_prestamo.nextval, p_id_cliente, p_tipoprestamo,
        TO_DATE(SYSDATE, 'DD/MM/YYYY HH:MI:SS'), p_monto_aprobado,
p_tasa_interes,
        p_letra_mensual, 0, 0, NULL, p_monto_aprobado, 0,
        TO_DATE(SYSDATE, 'DD/MM/YYYY HH:MI:SS'), USER, p_cod_sucursal);

    UPDATE Sucursal
    SET monto_prestamos = monto_prestamos + p_monto_aprobado
    WHERE cod_sucursal = p_cod_sucursal;

EXCEPTION
    WHEN DUP_VAL_ON_INDEX THEN
        DBMS_OUTPUT.PUT_LINE('Datos repetidos');
    WHEN NO_DATA_FOUND THEN
        DBMS_OUTPUT.PUT_LINE('Registro no Localizado');
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE('Ocurrió un error en la inserción de los
datos');
END PrestamoAprobado;

```

```

--LLAMADA A PROCEDIMIENTO
BEGIN
    PrestamoAprobado (1, 1, 5200, 0.08, 200, 1);
END;

--PROCED PARA LOS PAGOS RECIBIDOS
CREATE OR REPLACE PROCEDURE PagosRecibidos(
    p_cod_sucursal transacpagos.cod_sucursal%TYPE,
    p_id_cliente transacpagos.id_cliente%TYPE,
    p_tipoprestamo transacpagos.tipoprestamo%TYPE,
    p_montopagado transacpagos.monto_de_pago%TYPE,
    p_estado Transacpagos.estado%TYPE,
    p_fechainsercion transacpagos.fechainsercion%TYPE
) AS
BEGIN
    INSERT INTO Transacpagos
    VALUES(seq_id_transac.nextval, p_cod_sucursal, p_id_cliente,
    p_tipoprestamo, TO_DATE(SYSDATE, 'DD/MM/YYYY HH:MI:SS'), p_montopagado,
p_estado,
    p_fechainsercion, USER);
    EXCEPTION
        WHEN DUP_VAL_ON_INDEX THEN
            DBMS_OUTPUT.PUT_LINE('Id Duplicado');
        WHEN OTHERS THEN
            DBMS_OUTPUT.PUT_LINE('Ocurrió un error en la inserción de los
datos.');
```

```

    END PagosRecibidos;

--LLAMADA A PROCEDIMIENTO
BEGIN
    PagosRecibidos(1, 1, 1, 300, 'C', SYSDATE);
END;

--VISTA
create view datos_clienteprofesion AS
SELECT c.id_cliente,concat(concat(c.nombre, ' '),c.apellido)
AS cliente,
tr.tipoprestamo,
tr.monto_de_pago,
ct.telefono
FROM profesiones p
INNER JOIN clientes c
```

```
on c.id_cliente = p.id_profesion
INNER JOIN clientes_telefonos ct
on c.id_cliente = ct.id_cliente
INNER JOIN transacpagos tr
on c.id_cliente = tr.id_cliente;
```